

Volume 45 Number 4 December 2021

ISSN 0350-5596

# *Informatica*

**An International Journal of Computing  
and Informatics**



1977

## Editorial Boards

Informatika is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatika is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatika is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

### Executive Editor – Editor in Chief

Matjaž Gams  
Jamova 39, 1000 Ljubljana, Slovenia  
Phone: +386 1 4773 900, Fax: +386 1 251 93 85  
matjaz.gams@ijs.si  
<http://dis.ijs.si/mezi/matjaz.html>

### Editor Emeritus

Anton P. Železnikar  
Volaričeva 8, Ljubljana, Slovenia  
s51em@lea.hamradio.si  
<http://lea.hamradio.si/~s51em/>

### Executive Associate Editor - Deputy Managing Editor

Mitja Luštrek, Jožef Stefan Institute  
mitja.lustrek@ijs.si

### Executive Associate Editor - Technical Editor

Drago Torkar, Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana, Slovenia  
Phone: +386 1 4773 900, Fax: +386 1 251 93 85  
drago.torkar@ijs.si

### Executive Associate Editor - Deputy Technical Editor

Tine Kolenik, Jožef Stefan Institute  
tine.kolenik@ijs.si

### Editorial Board

Juan Carlos Augusto (Argentina)  
Vladimir Batagelj (Slovenia)  
Francesco Bergadano (Italy)  
Marco Botta (Italy)  
Pavel Brazdil (Portugal)  
Andrej Brodnik (Slovenia)  
Ivan Bruha (Canada)  
Wray Buntine (Finland)  
Zhihua Cui (China)  
Aleksander Denisiuk (Poland)  
Hubert L. Dreyfus (USA)  
Jozo Dujmović (USA)  
Johann Eder (Austria)  
George Eleftherakis (Greece)  
Ling Feng (China)  
Vladimir A. Fomichov (Russia)  
Maria Ganzha (Poland)  
Sumit Goyal (India)  
Marjan Gušev (Macedonia)  
N. Jaisankar (India)  
Dariusz Jacek Jakóbczak (Poland)  
Dimitris Kanellopoulos (Greece)  
Samee Ullah Khan (USA)  
Hiroaki Kitano (Japan)  
Igor Kononenko (Slovenia)  
Miroslav Kubat (USA)  
Ante Lauc (Croatia)  
Jadran Lenarčič (Slovenia)  
Shiguo Lian (China)  
Suzana Loskovska (Macedonia)  
Ramon L. de Mantaras (Spain)  
Natividad Martínez Madrid (Germany)  
Sando Martinčić-Ipišić (Croatia)  
Angelo Montanari (Italy)  
Pavol Návrat (Slovakia)  
Jerzy R. Nawrocki (Poland)  
Nadia Nedjah (Brasil)  
Franc Novak (Slovenia)  
Marcin Paprzycki (USA/Poland)  
Wiesław Pawłowski (Poland)  
Ivana Podnar Žarko (Croatia)  
Karl H. Pribram (USA)  
Luc De Raedt (Belgium)  
Shahram Rahimi (USA)  
Dejan Raković (Serbia)  
Jean Ramaekers (Belgium)  
Wilhelm Rossak (Germany)  
Ivan Rozman (Slovenia)  
Sugata Sanyal (India)  
Walter Schempp (Germany)  
Johannes Schwinn (Germany)  
Zhongzhi Shi (China)  
Oliviero Stock (Italy)  
Robert Trappl (Austria)  
Terry Winograd (USA)  
Stefan Wrobel (Germany)  
Konrad Wrona (France)  
Xindong Wu (USA)  
Yudong Zhang (China)  
Rushan Ziatdinov (Russia & Turkey)

# Index Dependent Nested Loops Parallelization with an Even Distributed Number of Steps

Ádám Pintér and Sándor Szénási

John von Neumann Faculty of Informatics, Óbuda University, Hungary

Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University, Hungary

E-mail: pinter.adam@nik.uni-obuda.hu

John von Neumann Faculty of Informatics, Óbuda University, Hungary

Faculty of Economics, J. Selye University, Slovakia

E-mail: szenasi.sandor@nik.uni-obuda.hu, szenasis@ujss.sk

**Keywords:** parallel algorithm, optimization, nested loop parallelization, index distribution

**Received:** April 27, 2020

*Parallel processing of algorithms is an effective way to achieve higher performance on multiprocessor systems rather. During parallelization, it is critical to minimize the difference between the processing time for threads. It is necessary to choose a method that can efficiently distribute the workload evenly across the threads. This paper deals with a special kind of nested loops where the internal loop iterator depends on the outer loop iterator. In such cases, the process can be represented as an upper (or lower) triangular matrix. This paper introduces a method for partitioning the outer loop according to the indices in an almost optimal manner, so that the partial loops in each thread will take nearly the same number of steps. In addition, we examine the potential of a perfect partition and try to determine the maximum (but still meaningful) partition size.*

*Povzetek: Predstavljena je metoda paralelizma za posebno vrsto vgnezenih zank.*

## 1 Introduction

There are multiple ways to find duplicated elements of a dataset, but in the cases where the size of the set drastically increases, simple sequential solutions have serious runtime limitations. In such cases, it is advisable to parallelize this process for faster execution.

The most common way to find duplicates is the brute force method, which compares each element to every other element in the dataset. This solution has the time complexity of  $O(n^2)$  and is rarely used in the real world. [17] An advanced version of this method considers that an element will always be self-consistent (reflexivity) and assumes that if an element is identical to the other, the inverse of the condition is also satisfied (symmetricity), or necessarily (for a more general solution) transitivity is not allowed. Given these rules, we got a non empty  $S \neq \emptyset$  set and a reflexive symmetric non-transitive  $R$  relation:

$$\begin{aligned} \forall a \in S : (aRa) \\ \forall a, b \in S : (aRb \Leftrightarrow bRa) \\ \exists a, b, c \in S : (aRb \wedge bRc) \not\Leftrightarrow aRc \end{aligned}$$

Suppose that we want to construct equivalence classes between the elements of the dataset so that the matching elements (duplicates) are considered as one class. If the dataset element pairs are represented as a matrix where the rows and columns represent the  $i$ -th element of the set, and

each cell indicates whether a comparison is needed, then the matrices are shown in Figure 1. If the  $R$  relation is ignored, that is, the brute force algorithm is used in the processing, then for a set of 20 elements 400 comparisons (Figure 1.b.) are needed. If we consider the reflexivity attribute of the relation, we can reduce the number of comparisons by 380 (Figure 1.c.), since the elements of the main diagonal can be omitted. If the symmetric property is included along with reflexivity, the number of comparisons will be reduced to 190 (Figure 1.d.), since the property guarantees that the order of two elements is interchangeable. Because of non-transitivity, it will be necessary to traverse this upper triangle to find all duplication for each element.

In this case at best options  $\frac{(N^2-N)}{2}$  comparisons are required (where  $N$  is the number of elements in the dataset), which cannot be further reduced (or more precisely, cannot be further reduced without additional information about the dataset). For example, if the dataset contains 10,000 elements, then 4,999,500 comparisons are required.

Although transitivity exists as an attribute during the testing for equality of elements in a dataset, non-transitivity may be unavoidable for other tasks satisfying the  $R$  relation. Examples of such cases include:

- In the first example, the elements of the dataset represent the countries, where the  $R$  relation represents the neighborhoods. It is said that country  $A$  borders  $B$

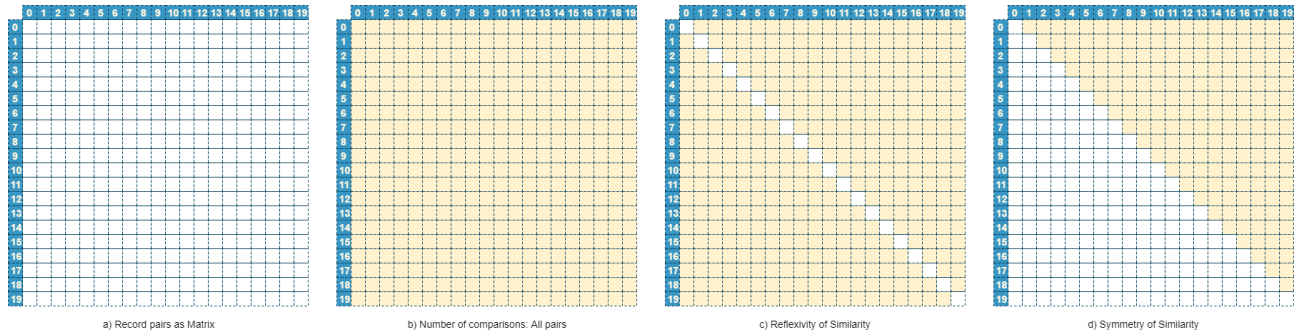


Figure 1: Dataset element pairs as a matrix marked with the different type of comparison requirement.

country, as well as  $B$  and  $C$  are neighbours, but that does not mean that countries  $A$  and  $C$  are also neighbours.

- In the second example, the  $S$  dataset contains a set of one-dimensional points where the relation  $R$  describes the distance between elements. It is also true that if there is  $\gamma$  distance between the  $A$  and the  $B$  and  $\gamma$  distance between  $B$  and  $C$ , it does not follow that there is also  $\gamma$  distance between  $A$  and  $C$ . In this case, non-transitivity will be met for all elements.
- Our last example concerns the topic of graphs, where  $S$  dataset contains the nodes of the graph and the relation  $R$  describes the edges between them. Here, to give an example, if a vertex  $C$  can be reached from  $A$  through  $B$ , it does not follow that there is an edge between  $A$  and  $C$ :  $A \rightarrow B \rightarrow C$ .

The third relation is the non-transitivity which cannot simplify the algorithm further. This case must check every potential pairs in the  $S$  set to determine the equivalence classes.

### 1.1 Sequential approach

To process the  $S$  dataset it is necessary to implement nested loops where the loops iterate through twice the whole set of data in case of brute force algorithm. (Algorithm 1) shows the pseudo code of the algorithm, called Naive Sequential Approach. In this case the main problem is the number of steps of the algorithm. For example, if the set size is  $N = 20$  the number of iterations (incrementation of one of the loop variables) is  $F_N = N^2 = 400$  and the number of comparisons is  $C_N = N^2 = 400$ .

Number of iterations and comparisons can be highly reduced if the elements meet the reflexivity and symmetricity requirements. In this case, it is enough to start the inner loop from the  $i + 1$ -th element due to symmetry. (Algorithm 2) shows the modification, where the main advantage is to link both objects in the same iteration due to reflexivity; this algorithm is called Improved Sequential Approach. Due to the change, the number of iterations is reduced to  $F_N = \frac{N*(N+1)}{2} = 210$  and the number of comparisons is

---

#### Algorithm 1 Naive Sequential Approach

---

**Require:**  $S$ : series of data

**Require:**  $N$ : number of elements in the series

**Ensure:**  $S$ : series of processed data

```

1: function NAIVESEQUENTIALAPPROACH( $S, N$ )
2:   for  $i \leftarrow 1$  to  $N$  do
3:     for  $j \leftarrow 1$  to  $N$  do
4:       if ( $S[i].P_1 R A[j].P_1$ ) then      ▷  $R$  is the
                                         relation between elements
5:          $S[i].P_2 = S[i].P_2 \cup \{A[j]\}$ 
6:       end if
7:     end for
8:   end for
9:   return  $S$ 
10: end function
    
```

*In the pseudo code the dataset contains objects having  $P_1$  and  $P_2$  properties. Where the  $P_1$  properties match, it links the target object to the  $P_2$  property of the source object.*

---

reduced to  $C_N = \frac{N^2 - N}{2} = 190$  for the same number of elements.

The algorithm presented in this article is generic and can handle all kinds of datasets (including ones without transitivity), such as string arrays or even object lists (as long as the  $R$  relation can be interpreted). The rest of this paper is organized as follows. In the next Section, an example of a triangular loop nest parallelization - which is distributed efficiently - using our technique is used to show the motivation of the paper and the already existing related results. Section 4 explains the mathematical aspects of the proposed technique and discusses its limitations. Experiments are presented in Section 5, highlighting the significant time improvements provided by the index-based distribution. And finally, we present our conclusion in the last section.

## 2 Related work

The parallelization of index dependent nested loops has been considered and developed by several authors to minimize execution time and optimize processor core utiliza-

**Algorithm 2** Improved Sequential Approach

---

**Require:**  $S$ : series of data  
**Require:**  $N$ : number of element in the series  
**Ensure:**  $S$ : series of processed data

```

1: function IMPROVEDSEQUENTIALAPPROACH( $S, N$ )
2:   for  $i = 1$  to  $N$  do
3:     for  $j = i + 1$  to  $N$  do  $\triangleright$  Due to the reflexivity.
4:       if ( $S[i].P_1 R A[j].P_1$ ) then  $\triangleright R$  is the
         relation between elements
5:          $S[i].P_2 = S[i].P_2 \cup \{A[j]\}$ 
6:          $S[j].P_2 = S[j].P_2 \cup \{A[i]\}$ 
7:       end if
8:     end for
9:   end for
10:  return  $S$ 
11: end function

```

---

tion. [11, 3, 15, 10, 6, 4, 9, 1, 12]

The collapsing of perfectly nested loops with constant loop bounds was originally introduced by Polychronopoulos [14] as loop coalescing, and Philippe Clauss et. al. [13] was further developed to work with non-rectangular loops. The problem of the original implementation - which only operates with loops that define constant loop boundaries - has been solved by defining non-rectangular iteration spaces whose bounds are linear functions of the loop iterators.

In [8], Nedal Kafri and Jawad Abu Sbeih found a solution to how to partition the triangular iteration space nearly optimal. In their case, the iteration space is defined as a lower triangle matrix and presents a method for calculating the lower and upper bound index of the outer loop of each partition. They have been able to achieve near-optimal load balancing and minimize load imbalance in parallel processing of a perfect triangular loop nest.

In [16], Rizos Sakellariou introduced a compile-time scheme that can efficiently partition non-rectangular loops whose indexes of internal loops depend on those of outermost loop. The technology presented is based on symbolic cost estimates, which minimize the imbalance of the load while avoiding other additional sources.

Adrian Jackson and Orestis Agathokleous [7] presented a developed system that allows a code to dynamically select which parallelization method to use at runtime. During the operation of their system, the programmer only has to specify the loop to be parallelized, after which the applied parallel technique will be selected by their dynamic library during the execution of the code.

### 3 Naive parallel approach

Considering the nested loops, the iterators  $i$  and  $j$  doesn't carrying any dependency, so they can be parallelized. [18] However, it is worth noting that the number of  $j$  iterations is not constant, it depends on the current value of  $i$ . If paral-

lization is performed according to the external cycle, the result is an imbalanced runtime for the threads. Therefore, it is necessary to choose another parallelization strategy (if we keep the original approach) to improve the runtime of the algorithm.

For example, if we keep the outer loop parallelization and create threads based on the number of outer loop iterations, the problem will be the difference in the runtime of the threads. The last thread will be completed much sooner than the first, whose internal loop will iterate through the entire dataset. In this case, the total runtime of the parallel algorithm will be determined by *thread 1*, as illustrated by Figure 2 and pseudo code called Naive Parallel Algorithm shown in Algorithm 3.

Therefore, if we simply halve the outer loop executions, the total iterations of the loops are 155 in the first part and 55 in the second part. So, the second part is only one third of the first one and finishes much earlier. It would better to use equal 105-105 iterations in both threads.

In our solution, we present a method that implements outer loop parallelization for the purpose of obtaining a nearly equal iteration of each threads, thereby eliminating differences in thread runtime.

#### 3.1 Nested loop parallelization

Numerous fields (numerical calculations, big data, etc.) use nested loops to either compute mathematical formulas or process large amounts of data, which typically stored in (often multidimensional) arrays. In the case of arrays, data is accessed through nested loops, their dependency relies on the data type and the processing method. When parallelizing such cases, it is first necessary to determine at which level of the cycles we want to parallelize, the strategies of which are illustrated in Table 1. [4, 7, 5, 19]

Table 1: Most common nested loops parallelization strategies

Type	Description
Outermost	Parallelization of the outermost loop.
Inner	Parallelization of one of the inner loop.
Nested	Parallelize multiple nested loops.
Collapsing	Nested loops collapse into a single loop.

##### 3.1.1 Outermost

This is the most common approach for parallelizing nested loops. In this strategy, the outermost cycle will be parallelized, with iterations distributed between the threads, thereby running the threads in parallel and performing the tasks assigned to them. In this case, the internal cycles are executed sequentially.

This strategy is generally a good choice (especially for high iteration counts) as it minimizes the cost of par-

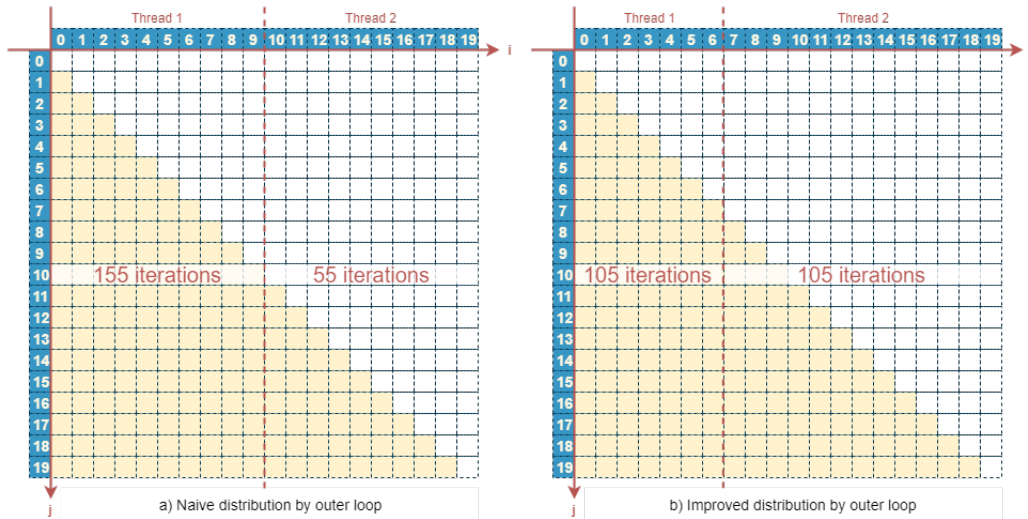


Figure 2: Unbalance of the runtime of the threads based on a uniform distribution of iterations.

allelization (for example, initializing threads, scheduling loop iterations on threads, thread synchronization).

Despite the advantages, the disadvantage is the limit on the maximum level of parallelization, which must not exceed the number of iterations of the loop to be parallelized. For example, a cycle that goes from  $i : 1 \rightarrow N$  can be divided into a maximum of  $N$  parts. This can limit the number of threads, which may prevent taking advantage of using all processors and cores of the system.

### 3.1.2 Inner

A variant of outermost parallelization, with the difference that one of the internal cycles is parallelized (as opposed to the external) while the outermost loop is executed sequentially. This strategy is necessary or useful if the external loop does not have a sufficient number of iterations for efficient parallelization, but it has the disadvantage that initializing, managing, and synchronizing threads can lead to performance problems.

### 3.1.3 Nested

This strategy takes advantage of the opportunity to execute multiple loops in parallel. Unlike the previous ones, which can use up to maximum as many threads as the number of iteration loops, this strategy also provides additional parallelization, which can give good results on systems with a large number of processors.

### 3.1.4 Collapsing

The basis for loop collapsing is that it transforms nested loops into a single loop, and then the newly created loop will be parallelized. The advantage of the transformation is that the ratio of parallelization is increased by the fact that the newly created loop will have a greater number of iterations. The method produces better runtime results than in-

ner and nested strategies, since reduces the amount of loop overhead (although this is not always possible as not all compilers provide this functionality).

## 4 Methodology

### 4.1 Index based equal distribution

To balance the number of iterations in every threads, it is necessary to determine the optimal number ( $O$ ) of iteration in one partition based on the size of dataset ( $N$ ) and the target number of partitions ( $P$ ) and calculate lower (inclusive) and upper (exclusive) indices for all partitions. It is important to note that partitioning happens according to the outer loop, so perfect resolution occurs only in exceptional cases. In all other cases, it is sufficient to achieve approximately the same number of iterations.

The optimal number of iterations for a partition is obtained by determining the number of all iterations (for a dataset of  $N$  elements):  $F_N = \frac{N(N+1)}{2}$  which is divided by  $P$  to get the optimal number of iterations of a partition:  $O = \frac{\frac{N(N+1)}{2}}{P} = \frac{N(N+1)}{2P}$ .

Using the previous example, the optimal distribution of iterations can be determined as follows:

- $N = 20 \rightarrow F_N = \frac{N(N+1)}{2} = \frac{20*(20+1)}{2} = 210$   
- total number of iterations,
- $P = 2 \rightarrow O = \frac{N(N+1)}{2P} = \frac{F_N}{P} = \frac{210}{2} = 105$   
- iteration within 1 partition.

Our goal is to achieve this  $O$  iteration count for each thread. We know that the first thread will start processing the elements from  $I_0 = 0$  (inclusive) index and the last thread index is  $I_2 = 20$  (exclusive). The question is how to determine internal indices. In the example where we want to split the set into two threads, we only need to define an

**Algorithm 3** Naive Parallel Approach**Require:**  $S$ : series of data**Require:**  $N$ : number of element in the series**Ensure:**  $S$ : series of processed data

```

1: function NAIVEPARALLELAPPROACH( $S, N$ )
2:    $P = \text{NumberOfProcessors}()$                                 ▷ Determine the optimal level of parallelism.
3:    $T = \text{CreateThreads}(P)$                                     ▷ Create  $P$  number of working thread.
4:   for  $p = 0$  to  $P - 1$  do
5:      $lower = \lceil \frac{p*N}{P} \rceil$                                 ▷ Calculate the  $lower$  (inclusive) index in thread.
6:      $upper = \lceil \frac{(p+1)*N}{P} \rceil$                             ▷ Calculate the  $upper$  (exclusive) index in thread.
7:      $T[p] = \text{ThreadProcess}(lower, upper, S)$ 
8:   end for
9:    $\text{WaitAll}(T)$ 
10:  return  $S$ 
11: end function

12: function THREADPROCESS( $lower, upper, S, N$ )
13:  for  $i = lower$  to  $upper$  do
14:    for  $j = lower + 1$  to  $N$  do
15:      if ( $S[i].P_1 \ R \ A[j].P_1$ ) then                            ▷  $R$  is the relation between elements
16:         $S[i].P_2 = S[i].P_2 \cup \{A[j]\}$ 
17:         $S[j].P_2 = S[j].P_2 \cup \{A[i]\}$ 
18:      end if
19:    end for
20:  end for
21:  return  $S$ 
22: end function

```

internal index (will be:  $I_1 = 6$ ), which will be an exclusive on the first, and an inclusive one on the second thread.

**4.2 Steps of individual indices calculation**

1. Determine the total number of iterations ( $F_N$ ) based on dataset size ( $N$ ):  $F_N = \frac{N(N+1)}{2}$
2. Define the target number of partitions ( $P$ ), then calculate the optimal number of one partition's iterations:  $O = \frac{F_N}{P}$
3. Knowing the first index ( $I_0 = K_0 = 0$ ), determine the following index approximation ( $K_1$ ) for the partition using the following equations:

- Number of steps from  $K_0$  to the end of the set ( $K_0$  is known):

$$\frac{(N - K_0)(N - K_0 + 1)}{2} \quad (1)$$

- Number of steps from  $K_1$  to end of the set ( $K_1$  is unknown):

$$\frac{(N - K_1)(N - K_1 + 1)}{2} \quad (2)$$

- Equation (3) is obtained by combining and explaining (1) and (2).

- Roots of (3) are:

$$a = -1$$

$$b = 2N + 1$$

$$c = \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P} - N^2 - N \quad (4)$$

- where  $b^2 - 4ac > 0$ ,  $-\frac{b}{a}, \frac{c}{a} > 0$  and  $a < 0$  so  $X_1 \leq X_2$  therefore: [2]

$$X_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (5)$$

- from above we can calculate the (next) approximation to the index,  $K_1$ :

$$K_1 = X_1 = \frac{-(2N + 1) + \sqrt{(2N + 1)^2 - 4 \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P}}}{-2} \quad (6)$$

- To get the real index (exclusive) need to round the  $K_1$  (but to calculate the next  $K_x$  index, we use the unrounded  $K$  value):

$$I_1 = \lceil K_1 \rceil \quad (7)$$

4. The index is calculated  $P$  times, and then  $K_p = N$  is obtained.

This methods is the Improved Parallel Approach (IPA), and its algorithm is illustrated by Algorithm 4.

$$\begin{aligned}
 & \frac{(N - K_0)(N - K_0 + 1)}{2} - \frac{(N - K_1)(N - K_1 + 1)}{2} = \frac{N(N + 1)}{2P} \\
 & P(N - K_0)(N - K_0 + 1) - P(N - K_1)(N - K_1 + 1) = N(N + 1) \\
 & P(N - K_0)(N - K_0 + 1) - N(N + 1) = P(N - K_1)(N - K_1 + 1) \\
 & \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P} = (N - K_1)(N - K_1 + 1) \\
 & \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P} = N^2 - NK_1 + N - NK_1 + K_1^2 - K_1 \\
 & \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P} = N^2 - 2NK_1 + N + K_1^2 - K_1 \\
 & \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P} = N^2 + N + K_1^2 - K_1(2N + 1) \\
 & \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P} - N^2 - N = K_1^2 - K_1(2N + 1) \\
 & -K_1^2 + K_1(2N + 1) + \frac{P(N - K_0)(N - K_0 + 1) - N(N + 1)}{P} - N^2 - N = 0
 \end{aligned} \tag{3}$$

### 4.3 Example of intermediate indices calculation

In this example the length of the set is  $N = 8$ , which we would like to distribute into  $P = 4$  parts. The total number of iterations are  $F_N = \frac{N*(N+1)}{2} = \frac{8*(8+1)}{2} = 36$ . One part optimally iterations are  $O = \frac{F_N}{P} = \frac{36}{4} = 9$ . From Equation 4 the  $a = -1$  is constant,  $b = 2 * N + 1 = 2 * 8 + 1 = 17$  is depends by the  $N$ , and  $c$  need to calculate in every iteration. The outer loop first index is also known:  $K_0 = 0$ . The intermediate indices' calculations are the following:

- Step 1 - calculate  $K_1$  and  $I_1$ :

$$\begin{aligned}
 c_1 &= \frac{4 * 8 * 9 - 8 * 9}{4} - 8^2 - 8 = -18 \\
 K_1 &= \frac{-17 + \sqrt{17^2 - 4 * -1 * -18}}{-2} = 1,135 \tag{8} \\
 I_1 &= 1
 \end{aligned}$$

For the first thread, the outer loop will iterate from 0 to 1, resulting in a total of 8 iterations.

- Step 2 - calculate  $K_2$  and  $I_2$ :

$$c_2 = -36 \quad K_2 = 2,479 \quad I_2 = 2 \tag{9}$$

For the second thread, the outer loop will iterate from 1 to 2, resulting in a total of 7 iterations.

- Step 3 - calculate  $K_3$  and  $I_3$ :

$$c_2 = -54 \quad K_2 = 4,227 \quad I_2 = 4 \tag{10}$$

For the third thread, the outer loop will iterate from 2 to 4, resulting in a total of 11 iterations.

- Step 4 - calculate  $K_4$  and  $I_4$ :

$$c_2 = -72 \quad K_2 = 8,000 \quad I_2 = 8 \tag{11}$$

For the fourth thread, the outer loop will iterate from 4 to 8, resulting in a total of 10 iterations.

As visible, we cannot reach the equal distribution, but we get close to it. The optimal iteration count for one thread is  $O = 9$ . The differences to this are as follows:  $-1$  ( $-11\%$ ),  $-2$  ( $-22\%$ ),  $+2$  ( $22\%$ ),  $+1$  ( $11\%$ ), with a cumulative error of 0.

To compare these results to the Naive Distribution, it generates the following indices:  $0 - 2, 2 - 4, 4 - 6, 6 - 8$ , with iteration counts as follows: 15 (44%), 11 (22%), 7 ( $-22\%$ ), 3 ( $-44\%$ ). It is clearly visible that the first thread will run approximately five times longer than the last one. In contrast, the threads of the improved version will finish at about the same time as is visible in Figure 3.

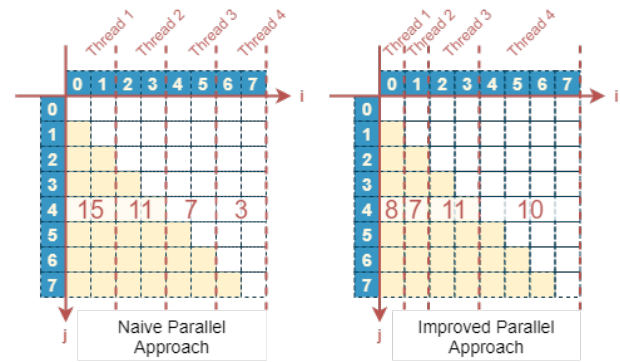


Figure 3: Difference between the two type of outer loop distribution.

### 4.4 Error in indices calculation

There are some cases where the last index will never be reached due to computational inaccuracy. Since rounding is an important step in defining indices, this problem will appear when the error "goes beyond" the decimal point, after that, rounding operation will not be able to correct the deviation. This error will only occur when calculating the last index, because in the case of intermediate indices, this difference is negligible as it causes minimal differences in iterations.



**Algorithm 4** Improved Parallel Approach**Require:**  $S$ : series of data**Require:**  $N$ : number of element in the series**Ensure:**  $S$ : series of processed data

```

1: function IMPROVEDPARALLELAPPROACH( $S, N$ )
2:    $P = \text{NumberOfProcessors}()$                                 ▷ Determine the optimal level of parallelizm.
3:    $T = \text{CreateThreads}(P)$                                     ▷ Create  $P$  number of working threads.
4:    $a = -1$ 
5:    $b = 2N + 1$ 
6:    $K_{lower} = 0$ 
7:    $I_{lower} = 0$ 
8:   for  $p = 1$  to  $P$  do
9:      $c = \frac{P(N - K_{previous})(N - K_{previous} + 1) - N(N + 1)}{P} - N^2 - N$ 
10:     $K_{upper} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ 
11:     $I_{upper} = \lceil K_{next} \rceil$                                 ▷ The upper (exclusive) index in thread.
12:     $T[p] = \text{ThreadProcess}(I_{lower}, I_{upper}, S, N)$ 
13:     $K_{lower} = K_{upper}$ 
14:     $I_{lower} = I_{upper}$ 
15:  end for
16:   $\text{WaitAll}(T)$ 
17:  return  $S$ 
18: end function

19: function THREADPROCESS( $lower, upper, S, N$ )
20:  for  $i = lower$  to  $upper$  do
21:    for  $j = lower + 1$  to  $N$  do
22:      if ( $S[i].P_1 \ R \ A[j].P_1$ ) then                                ▷  $R$  is the relation between elements
23:         $S[i].P_2 = S[i].P_2 \cup \{A[j]\}$ 
24:         $S[j].P_2 = S[j].P_2 \cup \{A[i]\}$ 
25:      end if
26:    end for
27:  end for
28:  return  $S$ 
29: end function

```

As an example, suppose that the size of the set is  $N = 350,000,000$  and the partition is  $P = 8$ , where the optimal iteration within a thread is  $O = 7,656,250,021,875,000$ . In this case, after the calculation, the indices are shown in Table 2.

At the last calculation, the index is below the expected value; therefore, it is necessary to correct it. To overcome this, it may be a solution to calculate the indices for  $P - 1$  partitions instead, and then take the  $N$  as the last index of an additional partition.

#### 4.5 Perfect partition

A perfect division is when every thread takes the same number of iterations, which matches the optimal number of calculated iterations. We have done practical experiences with all set sizes between 1 and 2147483647. It can be said that certain sets can be divided into two equal parts by iterations (Table 3). It is not possible to do the same with larger number of partitions (where the number of partitions is a power of two).

#### 4.6 Maximum meaningful size of the partition

Maximum meaningful partition stands for the  $P$  value, where at least one iteration of the external loop is executed for each partition (the  $I_{lower} < I_{upper}$  condition is met). This condition prevents us from creating threads that will not contain iterations. This condition is usually violated when either the set we want to process is too small, or when the level of parallelization is too high.

For example, take the previous example where  $N = 8$  and the value of  $P$  should be 6. In this case, the computed indices will be: 0–1, 1–2, 2–2, 2–4, 4–5, 5–8, where the third thread indices violate the condition: because it has a  $I_{lower}$  and  $I_{upper}$  indices value of 2, this thread will do nothing.

The maximum value of  $P$  has not yet been determined but can be restricted between a lower limit and an upper limit. The lower limit is obtained by taking  $\frac{N+1}{2}$ , because we can create partitions at least half of the element number. This equation can be derived from the following relation-

Table 2: Index distribution and deviation when  $N = 350,000,000$  and  $P = 8$ 

Index lower value	Index upper value	Number of iterations	Deviation
0	22 604 979	7 656 250 123 507 270	0,0000013274%
22 604 979	46 891 109	7 656 249 998 313 340	-0,0000003077%
46 891 109	73 300 705	7 656 249 988 081 220	-0,0000004414%
73 300 705	102 512 627	7 656 250 044 133 910	0,0000002907%
102 512 627	135 669 648	7 656 250 019 577 120	-0,0000000300%
135 669 648	175 000 000	7 656 249 913 887 130	-0,0000014105%
175 000 000	226 256 314	7 656 250 113 194 860	0,0000011927%
226 256 314	349 999 995	7 656 249 974 305 130	-0,0000006213%

Table 3: Size of datasets where the perfect partitioning with partition size of  $N = 2$  is possible

3	20	119	696	4 059	23 660
137 903	803 760	4 684 659	27 304 196	159 140 519	213 748 912
236 368 449	290 976 842	345 585 235	477 421 558	532 029 951	554 649 488
609 257 881	663 866 274	741 094 204	863 561 208	882 299 846	904 919 383
927 538 920	950 158 457	959 527 776	982 147 313	1 004 766 850	1 059 375 243
1 081 994 780	1 091 364 099	1 100 733 418	1 127 233 854	1 136 603 173	1 145 972 492
1 155 341 811	1 191 211 566	1 200 580 885	1 245 819 959	1 255 189 278	1 268 439 496
1 300 428 352	1 332 417 208	1 345 667 426	1 400 275 819	1 422 895 356	1 432 264 675
1 454 884 212	1 477 503 749	1 518 861 924	1 522 742 823	1 528 231 243	1 541 481 461
1 564 100 998	1 596 089 854	1 618 709 391	1 631 959 609	1 637 448 029	1 650 698 247
1 673 317 784	1 682 687 103	1 718 556 858	1 741 176 395	1 769 284 352	1 795 784 788
1 818 404 325	1 850 393 181	1 859 762 500	1 869 131 819	1 882 382 037	1 886 262 936
1 891 751 356	1 927 621 111	1 946 359 749	1 959 609 967	1 968 979 286	1 982 229 504
2 000 968 142	2 014 218 360	2 036 837 897	2 059 457 434	2 068 826 753	2 078 196 072
2 091 446 290	2 104 696 508	2 132 804 465	2 146 054 683		

ship:

$$\begin{aligned}
 N &\leq \frac{N(N+1)}{2P} \\
 1 &\leq \frac{N+1}{2P} \\
 P &\leq \frac{N+1}{2}
 \end{aligned} \tag{12}$$

The upper limit has not been proved yet, but we get the following theorem about it after expanding the lower limit as is shown in Equation 13.

$$P < \frac{N+1}{2 - \frac{2}{\sqrt{N}}} - 1 \tag{13}$$

Based on the *lower* and *upper* limit equations, the first 50 partition values are shown in Table 4 and Figure 4. As visible, the upper limit follows the real maximum  $P$  value (except some special cases), until then the lower limit is slowly getting further away.

In summary, if the  $P$  value is chosen less than or equal to the *lower* limit, the partitioning will be certainly correct, and each thread will contain processable iterations. In the case of higher values, the difference between the *lower* limit and the optimal value increasing; therefore, it is advisable to use the *upper* limit  $-1$  to gain high level of parallelism.

## 5 Evaluation

This section presents the testing methods for the runtime of the different versions of nested loops algorithms. The inputs were generated by the following parameters (resulting in a total of 96 datasets).

1. Size of the dataset ( $N$ ) can be 10, 100, 1 000, 10 000, 20 000, 50 000, 100 000, 200 000.
2. Minimum and maximum size of the elements in the dataset ( $E_l$ ) can be 10-100, 1 000-5 000, 10 000-15 000.
3. Probability of duplicates in the dataset ( $\beta$ ) can be 0, 0.33, 0.5, 0.8.

A C# application was developed for testing which runs each measurement 10 times using each algorithm. During the measurements, the best and worst results are discarded and the final result was the average of the remaining values.

The following configuration was used for testing:

- CPU: Intel(R) Core(TM) i5-7300 (2 physical cores, 4 logical cores)
- RAM: 16 GB

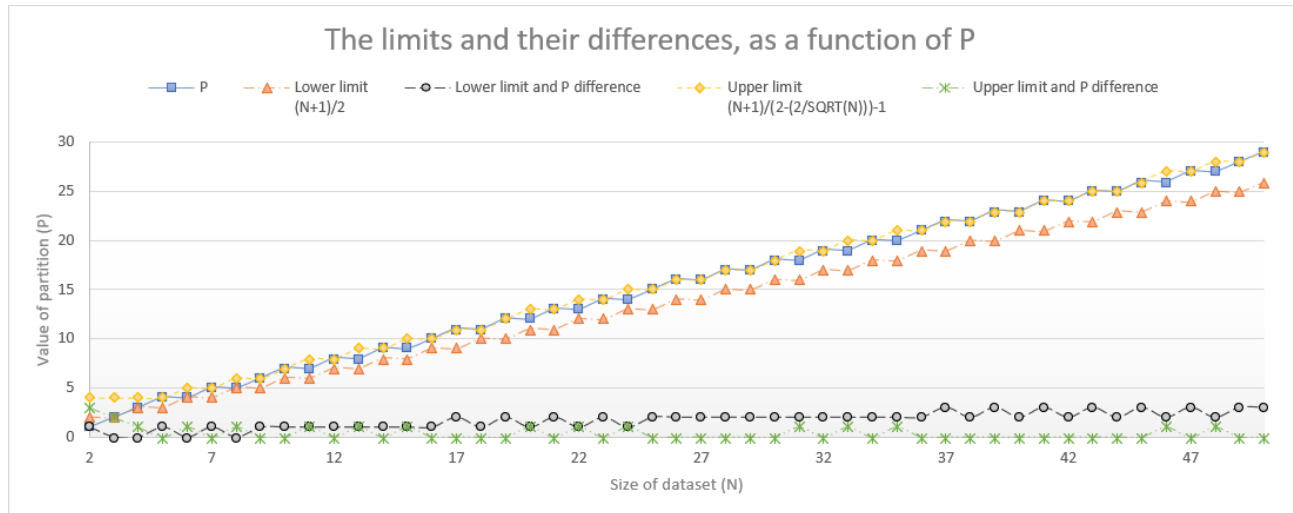


Figure 4: The evolution of limits and their differences, as a function of P

- HDD: Samsung MZVLB512HAJQ-000L7, 475,48 GB
- FileSystem: NTFS v3.1, cluster size: 4096 bytes
- Used level of parallelizm (P): 4

The results of the measurements show that the amount of duplication of elements within the set ( $\beta$ ) and the “length” of the elements ( $E_l$ ) do not effect the runtime significantly. The runtime difference between the four algorithms presented is determined by the differences in the dataset sizes. This observation is illustrated by the results of the measurements in Figure 5.

The diagrams can be seen that the slopes of the functions are nearly the same. For example, comparing the execution time of the Naive Serial algorithm in the second ( $E_l : 1000 - 5000$ ) and the third ( $E_l : 10000 - 15000$ ) diagram, can be seen, that even though in the second case the elements in the set are twice as long, runtime only increased depending on the number of elements. For example, for 50 000 elements, when the length of the elements was increased, the difference was only 100 927 317 ticks ( $\approx 0.010092732$  seconds), while for doubling the size of the set, the execution time increased by an average of 3 083 536 094 ticks ( $\approx 0.308353609$  seconds). Of course, this relationship also applies to other measurements.

It is also determined by the measurements (as visible in Figure 6) that it is worth using the parallelization in the case of more than 10,000 elements.

In cases where the size of the dataset does not reach that number, serial processing will result in faster execution time, regardless of the length of the elements.

In Figure 7, the two versions of parallel algorithms are compared and the differences between runtimes are highlighted.

As can be seen in cases where the dataset size exceeds the minimum number of elements required for parallelization ( $N \geq 10\,000$ ), the improved version of the parallel

algorithm always results in faster execution time (between 15 – 40 percentage) than the naive version.

## 6 Conclusion

The objective of this paper was to speed-up the calculation of a given  $R$  operation using all possible pairs of a set as operands. Several sequential and parallel procedures for this purpose have been presented and analysed.

The main contribution of this paper is a novel nested loop parallelization method based on the number of items. This method is able to give an efficient partitioning of the problem based on the number of items and the number of available processors.

Some further analysis was also presented about the numerical error of the method, perfect partitioning and the maximum meaningful size of the partition.

The evaluation section shows that the presented method is very effective and is able to give almost optimal results in all cases.

## Acknowledgement

This work is supported by the Doctoral School of Applied Informatics and Applied Mathematics, Óbuda University.

## References

- [1] Volodymyr Beletskyy and Maciej Poliwoda Parallelizing. Perfectly nested loops with non-uniform dependences. 2002.
- [2] H. Blinn. How to solve a quadratic equation? *IEEE Computer Graphics and Applications (Volume: 25 , Issue: 6)*, 2005.

- [3] Kiran Bondalapati. Parallelizing dsp nested loops on reconfigurable architectures using data context switching. *Proceedings of the 38th Design Automation Conference (IEEE Cat. No.01CH37232)*, 2001. doi:<https://doi.org/10.1109/dac.2001.935519>.
- [4] Beata Bylina and Jarosław Bylina. Strategies of parallelizing nested loops on the multicore architectures on the example of the wz factorization for the dense matrices. *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 629–639, 2015. doi:<https://doi.org/10.15439/2015f354>.
- [5] Beata Bylina and Jarosław Bylina. Parallelizing nested loops on the intel xeon phi on the example of the dense wz factorization. *Proceedings of the Federated Conference on Computer Science and Information Systems*, 8:655–664, 2016. doi:<https://doi.org/10.15439/2016f436>.
- [6] Benjamin James Gaska, Neha Jothi, Mahdi Soltan Mohammadi, and Kat Volk. Handling nested parallelism and extreme load imbalance in an orbital analysis code. *arXiv:1707.09668*, 2017.
- [7] Adrian Jackson and Orestis Agathokleous. Dynamic loop parallelisation. *arXiv:1205.2367*, 2012.
- [8] Nedal Kafri and Jawad Abu Sbeih. Simple optimal partitioning approach to perfect triangular iteration space. *Proceedings of the 2008 High Performance, Computing & Simulation Conference ©ECMS, Waleed W. Smari (Ed.), ISBN: 978-0-9553018-7-2 / ISBN: 978-0-9553018-6-5 (CD)*, pages 124–131, 2008.
- [9] Arun Kejariwal, Paolo D’Alberto, Alexandru Nicolau, and Constantine D. Polychronopoulos. A geometric approach for partitioning n-dimensional non-rectangular iteration spaces. In *Languages and Compilers for High Performance Computing*, pages 102–116, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [10] Sriram Krishnamoorthy, Muthu Baskaran, Uday Bondhugula, J. Ramanujam, Atanas Rountev, and P Sadayappan. Effective automatic parallelization of stencil computations. *Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 235–244, 2007. doi:<https://doi.org/10.1145/1250734.1250761>.
- [11] Christian Lengauer. Loop parallelization in the polytope model. *CONCUR’93*, pages 398–416, 1993.
- [12] Junyi Liu, John Wickerson, and George A. Constantinides. Loop splitting for efficient pipelining in high-level synthesis. *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2016. doi:<https://doi.org/10.1109/FCCM.2016.27>.
- [13] Matthieu Kuhn Philippe Clauss, Ervin Altintas. Automatic collapsing of non-rectangular loops. *Parallel and Distributed Processing Symposium (IPDPS), Orlando, United States*, pages 778–787, 2017.
- [14] S. D. Polychronopoulos. Loop coalescing: a compiler transformation for parallel machines. *University of Illinois at Urbana-Champaign, Center for Supercomputing Research and Development, Technical Report CSR-635*, 1987.
- [15] J. Ramanujam. Optimal software pipelining of nested loops. In *Proceedings of 8th International Parallel Processing Symposium*, pages 335–342, 1994. doi:<https://doi.org/10.1109/IPPS.1994.288280>.
- [16] Rizos Sakellariou. A compile-time partitioning strategy for non-rectangular loop nests. *Proceedings of the 11th International Parallel Processing Symposium, IPPS’97 (Geneva), IEEE Computer Society Press*, pages 633–637, 1997.
- [17] Thomas Schmidt and Jens Stoye. Quadratic time algorithms for finding common intervals in two and more sequences. *Proc of CPM 2004 LNCS*, 3109:347–358, 2004. doi:[https://doi.org/10.1007/978-3-540-27801-6\\_26](https://doi.org/10.1007/978-3-540-27801-6_26).
- [18] T. H. Tzen and L. M. Ni. Dependence uniformization: a loop parallelization technique. *IEEE Transactions on Parallel and Distributed Systems*, 4(5):547–558, 1993. doi:<https://doi.org/10.1109/71.224217>.
- [19] M.E. Wolf and M.S. Lam. A loop transformation theory and an algorithm to maximize parallelism. *IEEE Transactions on Parallel and Distributed Systems*, 2(4):452–471, 1991. doi:<https://doi.org/10.1109/71.97902>.

Table 4: The evolution of limits and their differences, as a function of P

N	P	Lower limit		Upper limit		N	P	Lower limit		Upper limit	
		Value	Diff.	Value	Diff.			Value	Diff.	Value	Diff.
2	1	2	1	4	3	27	16	14	2	16	0
3	2	2	0	4	2	28	17	15	2	17	0
4	3	3	0	4	1	29	17	15	2	17	0
5	4	3	1	4	0	30	18	16	2	18	0
6	4	4	0	5	1	31	18	16	2	19	1
7	5	4	1	5	0	32	19	17	2	19	0
8	5	5	0	6	1	33	19	17	2	20	1
9	6	5	1	6	0	34	20	18	2	20	0
10	7	6	1	7	0	35	20	18	2	21	1
11	7	6	1	8	1	36	21	19	2	21	0
12	8	7	1	8	0	37	22	19	3	22	0
13	8	7	1	9	1	38	22	20	2	22	0
14	9	8	1	9	0	39	23	20	3	23	0
15	9	8	1	10	1	40	23	21	2	23	0
16	10	9	1	10	0	41	24	21	3	24	0
17	11	9	2	11	0	42	24	22	2	24	0
18	11	10	1	11	0	43	25	22	3	25	0
19	12	10	2	12	0	44	25	23	2	25	0
20	12	11	1	13	1	45	26	23	3	26	0
21	13	11	2	13	0	46	26	24	2	27	1
22	13	12	1	14	1	47	27	24	3	27	0
23	14	12	2	14	0	48	27	25	2	28	1
24	14	13	1	15	1	49	28	25	3	28	0
25	15	13	2	15	0	50	29	26	3	29	0
26	16	14	2	16	0	51	29	26	3	29	0

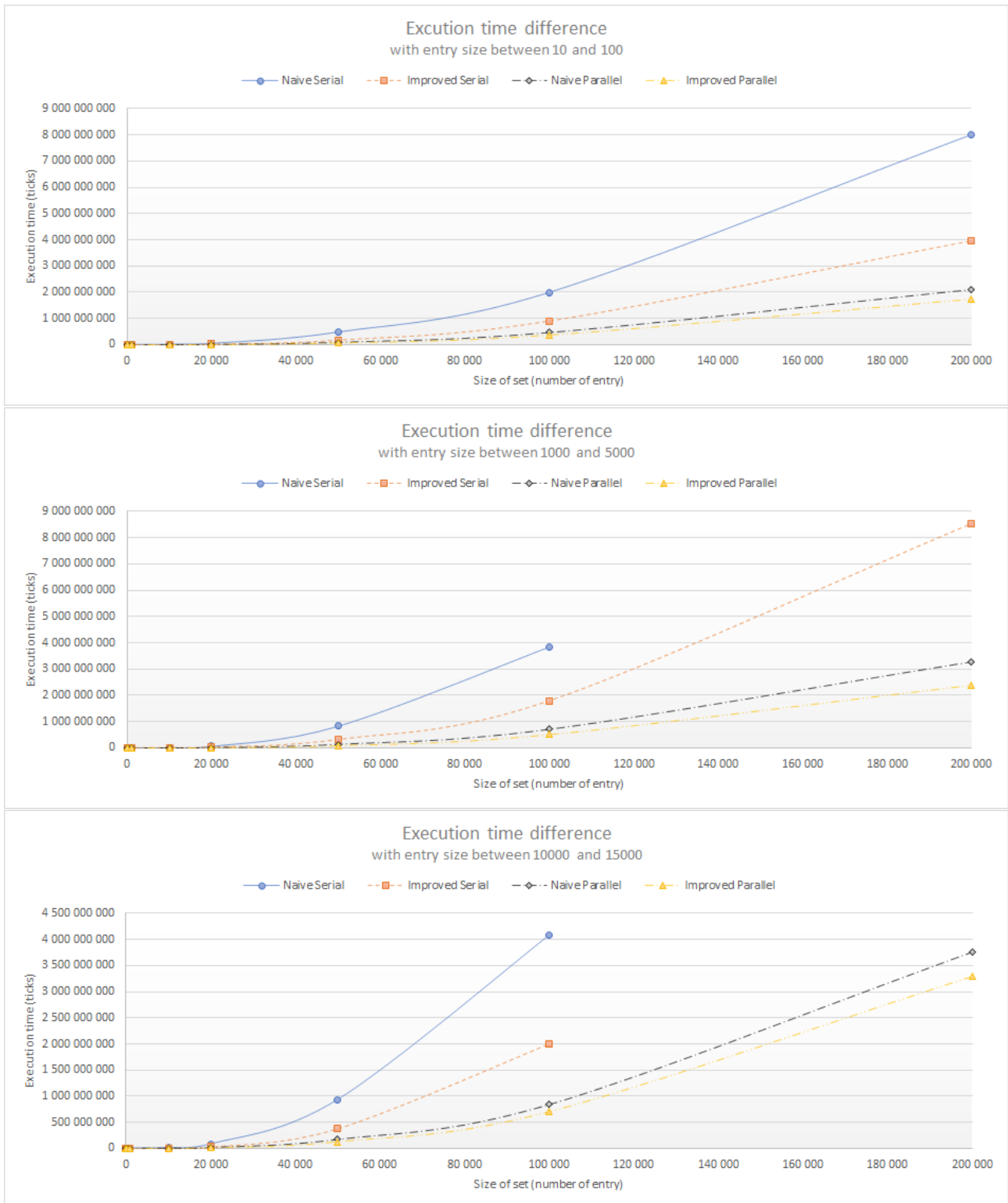


Figure 5: Execution time difference

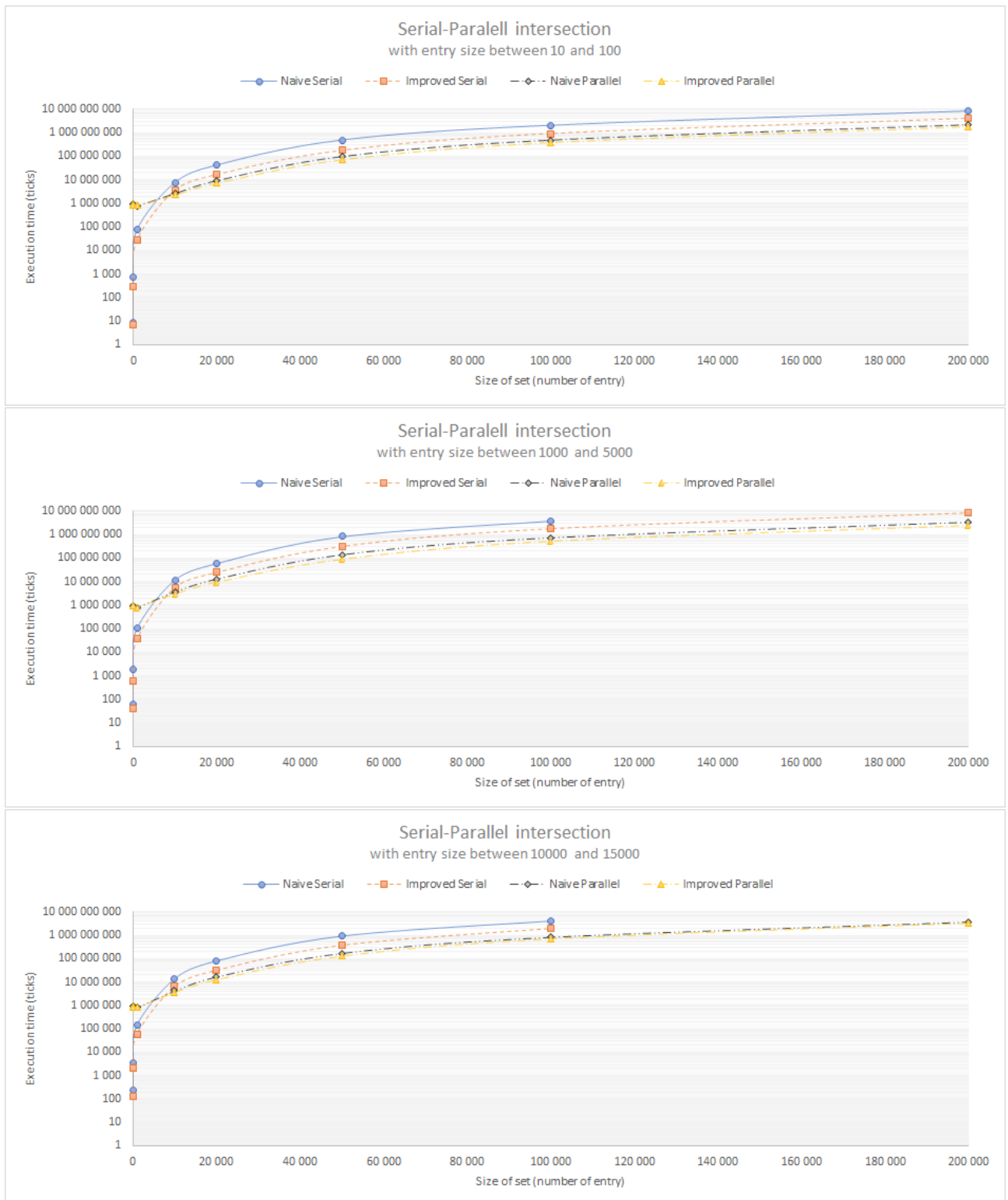


Figure 6: Serial-Parallel intersection

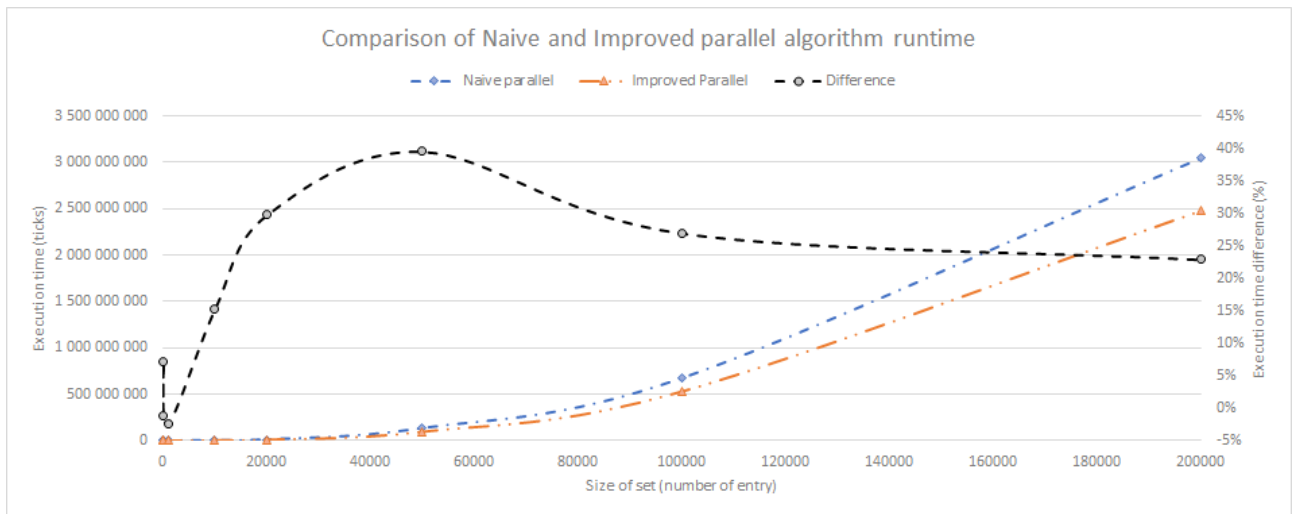


Figure 7: Comparison of Naive and Improved parallel algorithm execution runtime



# Clustering of Variables for Enhanced Interpretability of Predictive Models

Evelyne Vigneau

StatSC, Oniris, Inrae, 44 322 Nantes Cedex 03, France

E-mail: evelyne.vigneau@oniris-nantes.fr

**Keywords:** clustering of variables, linear model, boosting, high-dimensional data, dimensionality reduction, authentication

**Received:** August 18, 2020

*A new strategy is proposed for building easy to interpret predictive models in the context of a high-dimensional dataset, with a large number of highly correlated explanatory variables. The strategy is based on a first step of variables clustering using the CLustering of Variables around Latent Variables (CLV) method. The exploration of the hierarchical clustering dendrogram is undertaken in order to sequentially select the explanatory variables in a group-wise fashion. For model fitting implementation, the dendrogram is used as the base-learner in an L2-boosting procedure. The proposed approach, named *lmCLV*, is illustrated on the basis of a toy-simulated example when the clusters and predictive equation are already known, and on a real case study dealing with the authentication of orange juices based on <sup>1</sup>H-NMR spectroscopic analysis. In both illustrative examples, this procedure was shown to have similar predictive efficiency to other methods, with additional interpretability capacity. It is available in the R package *ClustVarLV*.*

*Povzetek: Prispevek opisuje nov pristop za izboljšano razlago prediktivnih modelov.*

## 1 Introduction

In the context of high-dimensional data with a large number of variables,  $p$ , and small number of observations,  $n$ , such as microarray data, metabolomic and volatolomic data (among a large variety of -omic data collected using high-throughput fingerprinting technologies), most recent statistical modelling strategies aim to achieve both efficient prediction and enhanced interpretability outcomes. For 20 years, the question of variable selection has driven a great deal of work, from discrete processes in which variables are either retained or removed, to lasso regularization processes in which several model coefficients may be shrunk towards zero [1, 2]. Regularization strategies discussions are still an active topic due to the high-dimensional of current data, either with model-based approach as in [3] or in combination with filtering step as in [4].

However, as Lasso tends to arbitrarily select one predictive variable among a group of highly correlated relevant variables, the sparsity principle of Lasso have been embedded into strategies for the selection of grouped predictors. The Elastic Net penalty [5] has been proven to be an efficient compromise between the Lasso ( $L_1$ ) penalty and the Ridge ( $L_2$ ) penalty. Fused Lasso [6] and Group Lasso [7] are designed for explanatory variables naturally ordered (as in vibrational spectroscopy) or arranged into groups (as for design factors). Sparse Group Lasso [8, 9] yields sparsity at both the group and individual feature levels, in order to select groups and predictors within a group. The Octagonal Shrinkage and Clustering Algorithm for Regression (OSCAR) [10] is another approach which combines an  $L_1$ -type penalty and a pairwise  $L_\infty$ -type penalty on the model's co-

efficients. Thus OSCAR allows to simultaneously select variables and perform supervised clustering in the context of linear regression. The CLERE methodology [11] is also based on the clustering of the regression coefficients using a Gaussian latent mixture model. Several of the penalized least squares approaches have Bayesian analogues also developed for variable selection and grouping [12, 13].

The model we propose here stems from another family of approaches. Emphasis is put on reducing the dimensionality of the data when a large number of highly correlated explanatory variables ( $p \gg n$ ) has been collected. We consider reduction dimension approaches that define latent components as being linear combinations of the explanatory variables and taking their correlation-structure into account. It is also desirable that these latent components are easy to interpret by making several component loadings to be exactly zeros for instance. This goal has also been addressed with approaches designed to constrain Principal Component [14, 15, 16] or PLS regression [17] loadings to be shrunk to zeros. Another strategy was to first perform a cluster analysis in order to construct interpretable components [18, 19].

Our proposal is also to first perform a clustering of the explanatory variables and then to fit a linear model between a response variable  $Y$  and  $p$  explanatory variables  $X_j$  ( $j = 1, \dots, p$ ) in a groupwise fashion. The algorithm adopted herein is easy to implement and is oriented towards the interpretability of the latent component introduced sequentially into the model. In the case study considered herein, each selected latent component can be assumed to be associated with a compound from its chemical spectrum in NMR (Nuclear Magnetic Resonance). With

metabolomic data, it is possible to imagine that the latent components associated to subsets of metabolites may be related to specific biological pathways. Within the context of DNA microarray data, similar studies have been undertaken [20, 21] in which hierarchical cluster analysis allows to identify supergenes, obtained by averaging genes within the clusters. These supergenes are thereafter used to fit regression models, thereby attaining concise interpretation and accuracy. One of the main differences in this study, compared with previous research works [20, 21], is that representatives of the clusters of variables, or latent variables, are not necessarily an average of the observed variables, but can be components that best reflect the variability within each cluster of variables. The second difference is in the procedure adopted for the progressive construction of the regression linear model.

The clustering of the explanatory variables is based herein on the CLV (CLustering of Variables around Latent Variables) approach [22, 23], implemented in the `ClustVarLV` R package [24]. In summary, the CLV approach consists of clustering together highly correlated variables into clusters (or groups of variables), while exhibiting within each cluster a latent variable (or latent component) representative to this cluster. It turns out that each latent component is defined as a linear combination of only the variables belonging to the corresponding cluster. From this point of view, CLV components are sparse components in the space of observations, and aim to best reflect the variance-covariance structure between the explanatory variables. Using the same approach, but defining a slightly different criterion, [18] also proposed a clustering of variables method for producing interpretable principal components. [25] investigated an approach for clustering of variables, based on the identification of a mixture with bipolar Watson components defined on the hypersphere. Herein, the central feature of the clustering procedure is a bottom-up aggregation approach of the explanatory variables involving the CLV criterion.

The dendrogram obtained is then explored in order to identify and select the predictive group's latent components regarding the response variable  $Y$ . In contrast with a previous study [26], hierarchical clustering, which is the more time consuming step, is performed only once and the fitting model stage has been modified accordingly. More precisely, an L2-boosting procedure for which the base-learner model is the CLV hierarchical algorithm has been considered. It consists to, iteratively, select a group of explanatory variables. The residuals of the response variable is then regressed on the latent component associated with the selected group and the predicted response is updated with the shrunken version of this local predictor.

The proposed methodology combining variables clustering and iterative linear model fitting, designated as *lmCLV*, is described in Section 2. Section 3 includes a simple simulated dataset in order to illustrate the behavior of the procedure in a known context, as well as one real case study dealing with the authentication of orange juice based on

$^1\text{H-NMR}$  spectroscopy.

## 2 Methodology

### 2.1 Notation

We consider the high-dimensional linear model :

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \quad (1)$$

where  $\mathbf{y} = [y_i]$  is a  $n \times 1$ -dimensional random response vector;  $\mathbf{X} = [\mathbf{x}_1 | \dots | \mathbf{x}_p] = [x_{ij}]$ , a  $n \times p$ -dimensional quantitative explanatory variables matrix (with  $i = 1, \dots, n$  and  $j = 1, \dots, p$ );  $\beta$ , a  $p \times 1$  regression coefficients vector and  $\varepsilon$  the  $n \times 1$ -dimensional random vector of the residuals, with zero mean and constant variance.

We consider contexts where  $p > n$  or  $p \gg n$  and where the explanatory variables may be arranged into groups of highly correlated variables.

Both the explanatory variables matrix and the response vector are assumed to be column-centered. In addition, the user may choose to standardize, or not, the variables to a unit variance.

### 2.2 *lmCLV*'s outlines

*lmCLV* combines two main methods:

1. The CLV method which is performed first. The similarities between the variables, herein evaluated on the basis of their covariance or correlation coefficients, are depicted by a tree diagram which is used as the learner for the model fitting method. The clustering of variables, using CLV method, is detailed in Sect.2.3.
2. The L2-boosting procedure which provides an efficient iterative model fitting method. The outline of the L2-boosting algorithm is depicted in Sect.2.4, and the way the CLV base-learner is used in the course of this procedure is explained in Sect.2.5.

### 2.3 Clustering of the variables

The clustering of the explanatory variables using the CLV method for directional groups [22, 24] aims to both identify clusters of variables and define a latent component associated with each cluster.

For a given number of clusters,  $K$ , the aim of the CLV method is to seek a partition  $\mathcal{G}_K = \{G_1, \dots, G_K\}$  of the variables into  $K$  disjoint clusters and a matrix  $\mathbf{C}_K = [\mathbf{c}_1 | \dots | \mathbf{c}_K]$  of  $K$  latent variables, each being associated with one cluster, so as to maximize the internal coherence within the clusters. When the agreement between the variables is assessed using their covariance or correlation coefficients, regardless of the sign of these coefficients, the clusters we are looking for are named directional groups. In this case, the aim is to define the partition,  $\mathcal{G}_K$ , and group's

latent variables matrix,  $\mathbf{C}_K$ , so that to get the optimal value  $T^*$  of the CLV criterion, so that:

$$T^* := \max_{(\mathcal{G}_K, \mathbf{C}_K)} \sum_{k=1}^K \sum_{j=1}^p \delta_{kj} \text{cov}(\mathbf{x}_j, \mathbf{c}_k)^2, \quad (2)$$

under the constraints  $\|\mathbf{c}_k\| = 1$ .

In eq.(2),  $\delta_{kj} = 1$  if the  $j^{\text{th}}$  variable belongs to the group  $G_k$ , and  $\delta_{kj} = 0$  otherwise. In other terms,  $(\delta_{kj})$  is the generic term of a binary matrix  $\Delta$  of the group's membership of the  $p$  variables.  $\Delta$  has only one nonzero element per row.

It is easy to show [22] that when the partition  $\mathcal{G}_K$  is fixed, the latent variable in a cluster  $G_k$  ( $k = 1, \dots, K$ ) is defined as the first standardized principal component of a data matrix formed by the variables belonging to this cluster. The latent variable  $\mathbf{c}_k$  associated with the cluster  $G_k$  ( $k = 1, \dots, K$ ) can therefore be expressed as a linear combination of the variables of this group:

$$\mathbf{c}_k = \sum_{j/\delta_{kj}=1} v_j \mathbf{x}_j. \quad (3)$$

For various numbers of clusters, from  $K = p$  clusters, in which each variable is considered to form a cluster by itself, to  $K = 1$ , where there is a single cluster containing all the variables, an ascendant hierarchical clustering algorithm is also proposed with respect to the CLV criterion and aggregating rules detailed in [22]. The strategy proposed herein consists of firstly constructing the whole hierarchy of the  $p$  explanatory variables and to explore repeatedly the dendrogram obtained.

### 2.4 L2-boosting procedure for model fitting

The second feature of the strategy is a boosting procedure, which can also be represented as a functional gradient descent (FGD) algorithm [27, 28], performed on the residuals of an iteratively updated shrunken linear model for the prediction of the response variable  $\mathbf{y}$ . The outline of the algorithm can be depicted as follows:

1. Set  $m = 0$ , and initialize  $\hat{\mathbf{y}}^{(0)}$ , for instance by choosing  $\hat{\mathbf{y}}^{(0)} = \mathbf{1}_n \bar{y}$ , with  $\mathbf{1}_n$  a vector of ones of length  $n$ , and  $\bar{y} = 1/n \sum_{i=1}^n y_i$ ,
2. Increase  $m$  by 1, and compute the residuals  $e_i^{(m)} = y_i - \hat{y}_i^{(m-1)}$  for  $i = 1, \dots, n$ ;
3. Apply the base-learner procedure to the actual residuals. The aim at this step is to identify the "best" CLV latent component denoted  $\mathbf{c}^{*(m)}$ . This base-learner procedure will be described in Sect.2.5;
4. Update the predictive function, i.e.  $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \nu \alpha^{(m)} \mathbf{c}^{*(m)}$  where  $0 < \nu \leq 1$  is a shrinkage parameter and  $\alpha^{(m)}$  the Ordinary Least Squares (OLS)

coefficient estimate of the linear regression of  $\mathbf{c}^{*(m)}$  on  $\mathbf{e}^{(m)} = [e_1^{(m)}, \dots, e_n^{(m)}]$ .

5. Return to step 2., until  $m = M$  ( $M$  being a large pre-terminated integer).

This procedure depends on two parameters: the stopping iteration parameter,  $M$ , and the shrinkage parameter,  $\nu$ , which can be determined via cross-validation or other information criterion as previously suggested in [28].

In practice, because our base-learner procedure returns one-dimensional components associated with sequential group-wise selections of the explanatory variables, we have often observed (as it will be shown in Sect.3) that a relatively high value of  $\nu$  (greater than 0.5) generally performs better. Moreover, as the predictive ability of the model appeared to be relatively stable, large values of  $M$  (say 50 or more) are not necessarily useful.

### 2.5 Base-learner procedure

The third step of the algorithm presented in Sect.2.4 is the core of the proposed strategy. At each iteration,  $m$ , of the algorithm, the aim is to select a cluster of explanatory variables and their associated representative, i.e. the group's latent component  $\mathbf{c}^{*(m)}$ . This choice is guided by CLV hierarchical clustering of the  $p$  explanatory predictors of  $\mathbf{X} = [\mathbf{x}_1 | \dots | \mathbf{x}_p]$  (Sect.2.3).

- For each size of partition (from  $p$  clusters to one cluster), we first aim to identify the CLV latent component which has the largest correlation (in absolute value) with the actual residuals  $\mathbf{e}^{(m)}$ . For  $q = 1, \dots, p$ , that is for each partition  $\mathcal{G}_{p-q+1}$ , we define :

$$\mathbf{c}_q^* := \max_{k \in \{1, \dots, (p-q+1)\}} |\text{cor}(\mathbf{c}_k, \mathbf{e}^{(m)})|. \quad (4)$$

- The next step consists of choosing a specific level  $q$  between 1 and  $p$ , that is a latent component  $\mathbf{c}_q^*$  and its associated group of predictors  $G_q^*$ , in such a way that  $G_q^*$  is as large as possible while accommodating an unidimensionality criterion. The unidimensionality condition is assessed herein using the modified Kaiser-Guttman (KG) rule [29].

If we denote  $m$  this specific level:

$$m := \arg \max_{q \in \{1, \dots, p\}} (|G_q^*| / \lambda_1 > \mathcal{L} \text{ and } \lambda_2 \leq \mathcal{L}), \quad (5)$$

where  $|G_q^*|$  denotes the cardinal of  $G_q^*$ ,  $\lambda_1$  and  $\lambda_2$  are respectively the first and the second largest eigenvalues of the correlation matrix of the explanatory variables belonging to  $G_q^*$  and the threshold  $\mathcal{L}$  is defined according to [29] by:

$$\mathcal{L} = 1 + 2\sqrt{\frac{|G_q^*| - 1}{p - 1}}.$$

The latent component  $\mathbf{c}^{*(m)}$  and the groups of explanatory variables in  $G_m^*$  are returned to the main algorithm (Sect.2.4) which continues at step 4.

Finally, it can be noticed that at each step the base learner returns a latent component which is itself a linear combination of a subset of the explanatory variables (eq.3). It is therefore easy to reformulate the prediction function in terms of a linear combinations of the  $p$  predictors, and to obtain an estimate of the coefficients' vector  $\beta$  (eq.1).

### 3 Applications

#### 3.1 A toy-simulated example

The *lmCLV* procedure is illustrated in this section using a simple example based on a simulated model as in [26]. We considered herein  $p = 70$  explanatory variables supposed to be measured on  $n = 100$  observations. Moreover, these variables were assumed to be structured into five groups ( $\mathcal{G}_1$  to  $\mathcal{G}_5$ ) of various sizes:  $\mathcal{G}_1$  was the largest group consisting of 35 variables,  $\mathcal{G}_2$  was the smallest group with 5 variables, whereas  $\mathcal{G}_3, \mathcal{G}_4$  and  $\mathcal{G}_5$  were 10 variables each.

Each group of variables was generated around a prototypical variable. The five prototypes were centered and standardized random variables with a known structure of covariance. In practice,  $n$  realizations of a vector  $(Z_1, \dots, Z_5)^t$  were generated from a centered multivariate normal distribution with a covariance matrix  $\Sigma$ :

$$\Sigma = \begin{pmatrix} 1 & 0.5 & 0.5 & 0.5 & 0.1 \\ 0.5 & 1 & 0.5 & 0.1 & 0.1 \\ 0.5 & 0.5 & 1 & 0.1 & 0.1 \\ 0.5 & 0.1 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{pmatrix}. \tag{6}$$

Let us denote  $\mathbf{Z}$  the  $n \times 5$  generated prototypical matrix. Then, the variables of each group were randomly simulated according to a multivariate normal distribution  $\mathcal{N}(\mathbf{0}_n, \mathbf{I}_n)$ , where  $\mathbf{0}_n$  represents the  $n$ -dimensional null vector and  $\mathbf{I}_n$  the  $n \times n$  identity matrix, as follows:

$$\mathbf{x}_j = \omega_j \mathbf{Z} \boldsymbol{\Lambda}^t + \varepsilon_j \text{ for } j \in \{1, \dots, p\}, \tag{7}$$

where  $\boldsymbol{\Lambda}$  is a  $p \times 5$  binary matrix defining the allocation of explanatory variables into the 5 groups. The column-wise marginal sum of  $\boldsymbol{\Lambda}$  was  $[35, 5, 10, 10, 10]$ . In eq.(7),  $\omega_j \in \{+1, -1\}$  was used to randomly create positive or negative correlations between each simulated variable  $\mathbf{x}_j$  in a group and the associated prototypical variable, in order to generate directional groups of variables.

Finally the response variable  $\mathbf{y}$  ( $\in \mathbf{R}^n$ ) was generated with the following model:

$$\mathbf{y} = \mathbf{Z} \mathbf{b} + \varepsilon \text{ with } \mathbf{b} = [1, 5, 3, 0, 0]^t, \tag{8}$$

where  $\varepsilon$  resulting from a multivariate normal distribution  $\mathcal{N}(\mathbf{0}_n, \mathbf{I}_n)$ .

The response  $\mathbf{y}$  was supposed to be mainly related to the variables of the smallest group,  $\mathcal{G}_2$ , moderately to the variables in  $\mathcal{G}_3$  and the smallest for the most numerous variables in  $\mathcal{G}_1$ . According to the parameters of the simulation, the expected correlation coefficients between  $\mathbf{y}$  and five prototypical variables are 0.69, 0.96, 0.82, 0.18 and 0.12, respectively. This toy-simulated example was designed to represent a simplified, but realistic, case study.

The choice of the shrinkage parameter  $\nu$  is discussed on the basis of the Root Mean Squared Error criterion evaluated using a five-fold cross-validation procedure, denoted  $RMSE_{CV}$ . This criterion was assessed at each iteration  $m$  from 0 (null model) up to  $M = 20$ . Figure 1 shows that as the parameter  $\nu$  becomes smaller, the decrease in prediction errors becomes slower. In this example, a value of  $\nu = 0.7$  or  $0.8$  led to a low level of errors ( $RMSEP_{CV} = 2.29$ ) after only three iterations of the algorithm. In the following,  $\nu$  was fixed to 0.7.

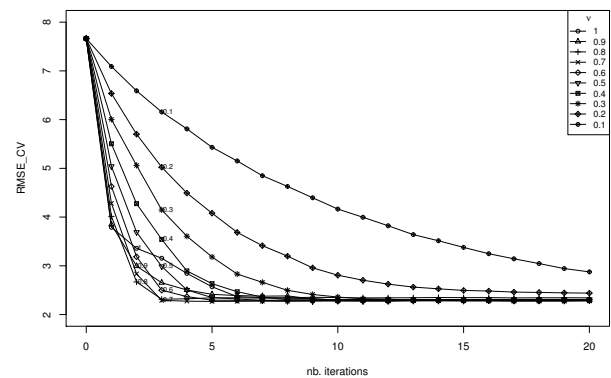


Figure 1: Evolution of the Root Mean Squared Error in Cross-validation with *lmCLV*, according to the number of iterations, and for various values of the shrinkage parameter,  $\nu$  (from 1 to 0.1).

At the first iteration, and for each fold of the cross-validation procedure, the variable numbers 36 to 40 ( $\mathcal{G}_2$ ) were in the cluster associated with the selected CLV latent component (for one of the five folds the variable 33 was also added, and for another fold the variable 46 was added). At the second iteration, the  $\mathcal{G}_3$  variables (41 to 50) were always retained (except for one fold, for which the variable 48 was lacking). Finally, at the third iteration, as expected, all the variables associated with  $\mathcal{G}_1$ , number 1 to 35, were in the selected cluster of variables.

For comparison of *lmCLV* with two usual regression methods based on latent components, namely Principal Components Regression (PCR) and Partial Least Squares Regression (PLSR), the same dataset was considered, using the same five consecutive segments for Cross-Validation. The evolution of the  $RMSE_{CV}$  according to the number of latent components included are shown in Figure 2(b). The left hand side panel (Figure 2(a)) shows the evolution of

the errors in prediction, more precisely the  $RMSE$ , evaluated on the calibration set formed by the whole dataset. As can be observed in Figure 2(a), PLSR and, to a weaker extent, PCR are prone to over-fitting, with the  $RMSE$  value reaching 0.94 when  $p = 70$  components are included (i.e. the OLS solution). On the contrary the  $RMSE$  with  $lmCLV$  remained relatively stable with increasing numbers of iterations. As a matter of fact, the higher the number of iterations, the more often the same groups of variables, and thus the same latent components, are likely to be included in the model. However, the loadings of these repeated latent components are becoming increasingly smaller, which has no impact on the overall quality of the model. With three latent components, extracted during three iterations of the algorithm,  $lmCLV$  made it possible to obtain a value of  $RMSE = 2.23$  and of  $RMSE_{CV} = 2.29$ . Regarding the  $RMSE_{CV}$  criterion, in this example,  $lmCLV$  performed a little better than PLSR and PCR. As with  $lmCLV$ , the PLSR solution with three components could be retained. However, except for the first PLS component which was quite well correlated ( $r = -0.94$ ) with the third latent component retained with  $lmCLV$  (also well correlated with the third prototypical variable), the other components were not pairwise well correlated. In fact, the PLS components are two by two uncorrelated, which is not the case for components from CLV.

The component-wise gradient boosting, or L2-boosting approach [28, 30] has also been considered for comparison purposes using the `glmboost` function included in the R package `mboost`. We used the same cross-validation structure as previously and tested step length values  $\nu = 0.01$  and  $0.1$  to  $1$  by step of  $0.1$ , with a maximal number of iteration of  $1000$ . The best combination of the shrinkage parameter  $\nu$  and the number of iterations, was chosen according to the  $RMSE_{CV}$  criterion. For this example, the best combination was  $\nu = 0.3$  and a stopping iteration number  $m_{stop} = 30$ , leading to a  $RMSE_{CV}$  value of  $2.45$ , which is similar to the optimal values observed with previous methods. With both of these parameters,  $23$  variables were selected: seven from  $\mathcal{G}_1$ , the five variables from  $\mathcal{G}_2$ , six variables from  $\mathcal{G}_3$ , and two variables from each of the last two group.

### 3.2 Orange juice authentication case study

In this case study, a  $^1\text{H}$  NMR spectroscopic profiling approach was investigated to discriminate between authentic and adulterated juices [31]. In this study, we considered the adulteration of orange juice (*Citrus sinensis*) with clementine juice (*Citrus reticulata*). Supplementation of substitution with cheaper or easier to find similar fruit is one of the type of fraud conducted within the fruit juice industry. For the experiment, twenty pure orange juices and ten pure clementine juices were selected from the Eurofins database. They are deemed to be representative of the variability of the fruit juices available on the market. From these juices,  $120$  blends were prepared by mixing one of the twenty or-

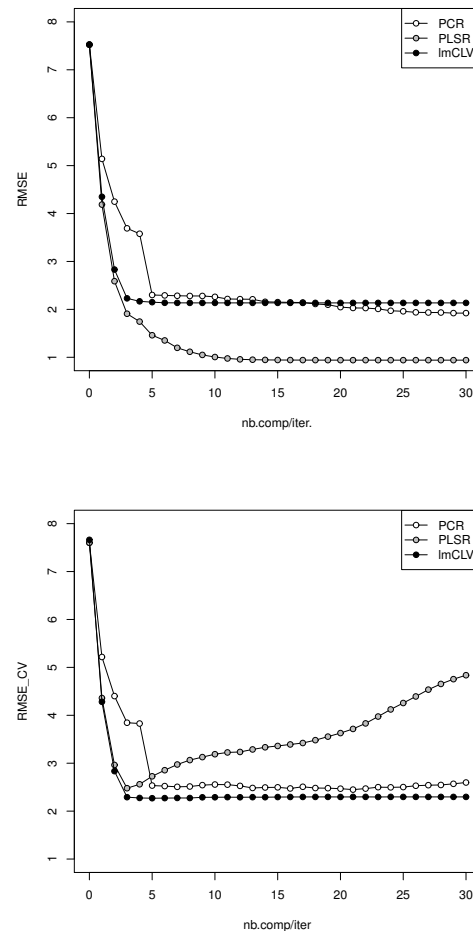


Figure 2: Evolution of the Root Mean Squared Errors.

ange juices and one of the ten clementine juices in known proportions. The proportion of the clementine juice in a mix were  $10\%$ ,  $20\%$ ,  $\dots$ , or  $60\%$ . The experimental design is described in more detail in [31]. The database was completed with the twenty authentic orange juices, for a total of  $140$  juice samples, and these were analyzed on an  $^1\text{H}$  Nuclear Magnetic Resonance (NMR) spectroscopy platform.

The NMR variables are associated with chemical shifts, given in ppm. In the following, two spectral regions were simultaneously considered: The region from  $6$  to  $9$  ppm and the region from  $0.5$  to  $2.3$  ppm. The first spectral region mostly includes chemical shifts associated with aromatic components and the second one due to amino-acid-specific spectral shifts (among others). Between these regions lies the typical  $^1\text{H}$  NMR spectra for orange juice sugars [32], which cannot discriminate between *Citrus sinensis* and *Citrus reticulata* juice. After a preprocessing binning step,  $300$  chemical shift variables were collected between  $6$  to  $9$  ppm, and  $180$  variables between  $0.5$  to  $2.3$  ppm.

The data matrix is therefore composed of  $n=140$  observations and  $p=480$  variables. The log-transformed data

are available in the R package *ClustVarLV* (dataset *authen\_NMR*). The mean of the log-transformed signals for both spectral regions are shown in Figure 3.

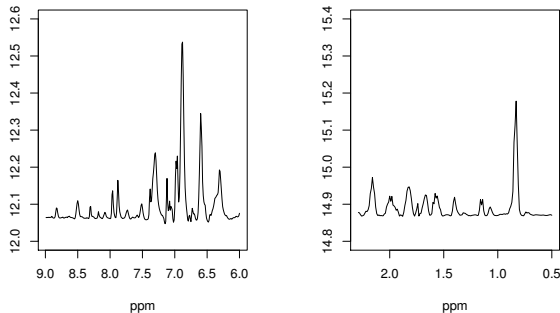


Figure 3: Mean spectrum of the log-transformed NMR signals according to the spectral region. Two subplots are shown due to the differences with order of magnitude in both regions.

In the following, a pareto-scaling was applied to each variable. This scaling, which consists of dividing each variable by the square root of its standard deviation, was shown in this case study [31] to be preferable to the usual pre-scaling by the standard deviation. Moreover, the splitting of the observations into ten segments was defined for Cross-Validation purposes. A proportional stratified allocation rule has been adopted in such a way that each segment contained two observations of each of the seven experimental levels, ranging from 0 to 60% of co-fruit added to pure orange juices.

Figure 4 shows the evolution of the errors in prediction, evaluated by means of the  $RMSE_{CV}$  criterion, when the shrinkage parameter  $\nu$  was set to 0.2, 0.5 or 0.8. On this basis, we choose to consider a value of the shrinkage parameter  $\nu = 0.5$  for more detailed analysis of the model.

Besides the predictive purpose, one of the key features of *lmCLV* is to identify groups of explanatory variables and their associated latent variables that are, step by step, selected and involved in the model. The introduction sequence of the CLV latent variables provides an interesting insight. However, it should be noted that the smaller the shrinkage parameter, the more often the same group of explanatory variables, by the means of its latent variable, will appear. Therefore, some of the *lmCLV* algorithm's outputs are provided for each exhibited group, in order of its first occurrence, rather than by iteration. Thus, the OLS coefficients,  $\alpha^{(m)}$ , associated with a selected latent component  $\mathbf{c}^{*(m)}$  (see step 4 in Sect.2.4) are aggregated on all iterations,  $m$ , to which this specific latent component is selected, up to the predefined maximum number of iterations,  $M$ . The same applies to the  $\beta$  coefficients of each of the (pre-processed) explanatory variables belonging to a selected group. Finally, a group importance criterion has been introduced. This importance is assessed as the sum,

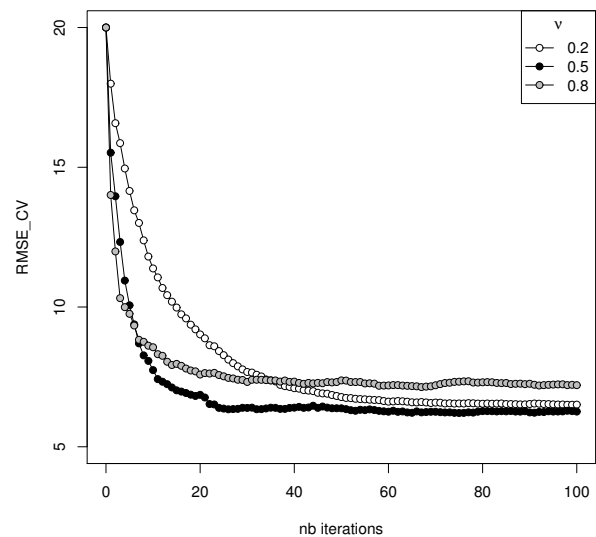


Figure 4: Evolution of the Root Mean Squared Error in Cross-validation with *lmCLV*, according to the number of iterations, and the values of the shrinkage parameter,  $\nu$ , in the orange juice authentication case study.

over all its occurrences, of the decrease in residual variance allowed by introducing the shrunken OLS estimate of the associated latent component.

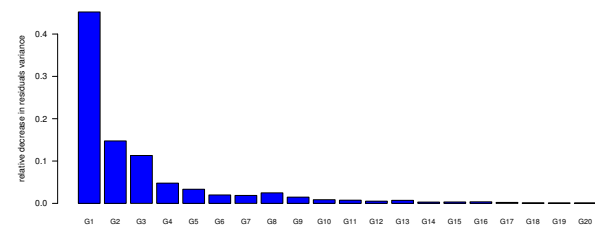


Figure 5: Relative Group Importance for the orange juice authentication case study. The Group Importance values are expressed relatively to the total variance of the response variable  $\mathbf{y}$ .

In our case study, using the whole data set, with *lmCLV* parameters  $\nu = 0.5$  and  $M = 25$ , the group importance estimates are depicted in Figure 5. This reveals the presence of a group of spectral variables (G1) of high importance, as well as two other important groups (G2 and G3). Each group is numbered according to its first occurrence, which corresponds rather well with its importance in the model.

The first group involved nine spectral variables at 7.52, 7.51, 7.50, 6.77, 2.10, 2.08, 2.07, 2.06 and 2.04 ppm. The spectral range between 7.50 and 7.52 ppm, combined

with the signal at 6.77 ppm and around 2 ppm, is particularly interesting and could be associated with 4 amino-3-methylbenzoic acid (see for instance the Spectral Database for Organic Compounds [33]). This information is quite stable: the spectral variables at 7.52, 7.51, 6.77 and 2.08 ppm had been selected in the first group at each of the 10 iterations of the CV procedure. The same variables have also been noted in [31], but their presumed association with the same compound could not be so clearly highlighted. The second group of spectral variables consisted of ten variables (7.15, 7.10-7.09, 1.10-1.05 and 0.96 ppm). The CV procedure showed that the range between 1.09 and 1.05 ppm and the peak at 7.10-7.09 ppm was simultaneously retrieved 7 times out of 10 as the second or third group. Lastly, the third group consisted of nine spectral variables, and specifically a range between 1.65 and 1.60 ppm. The relationship between these first three group components and the proportion of clementine juice in a mix is shown in Figure 6.

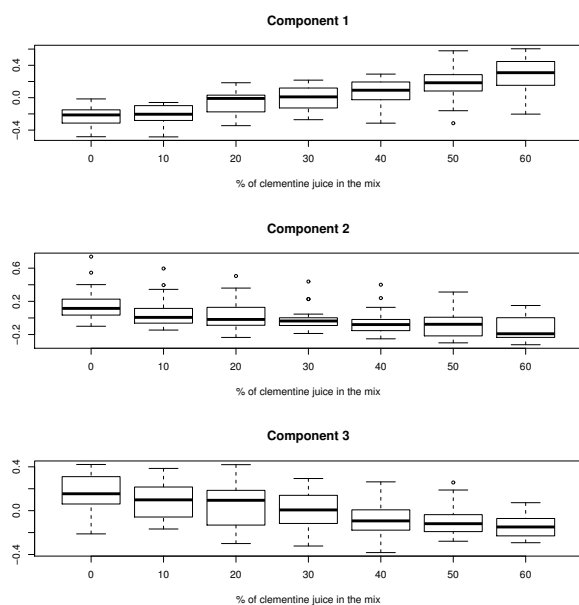


Figure 6: relationship between the first three group components and the proportion of clementine juice added to a mix.

In Table 3.2, the predictive ability of *lmCLV* with  $\nu = 0.5$  is compared with those of various alternatives among the most popular statistical learning approaches [1, 27]: (i) Sparse PLS Regression [17] which is very widely used in chemometrics; (ii) Elastic-Net [5], a penalized regression method allowing sparsity and grouping effect in variable selection; (iii) L2-boosting [28, 30], a simple boosting approach for squared error loss function which is based on a regression model, unlike the base learner involved in *lmCLV*; and (iv) Random Forest [34], one of the most popular and intuitive machine learning method for high dimensional predictive problem. The  $RMSE_{CV}$  were evaluated using the same 10-fold Cross-Validation division as above.

For Sparse PLS Regression (sPLSR) the R package *spls* was used, with parameter  $\eta \in \{0, 0.1, \dots, 0.9\}$  and  $K$ , the number of components, varying from 1 to 20. Elastic-Net (E-Net) was adjusted using the R package *glmnet* and by considering the Gaussian family model with mixing parameter  $\alpha$  fixed to 0.5. The parameter  $\lambda$  was chosen as the mean of optimal values determined for each CV fold. For the L2-boosting approach (L2-boost), in the R package *mboost*, values of parameter  $\nu$  from 0 (0.01 precisely) to 1, by 0.1, with a maximal number of 1000 iterations, was explored. Random Forest was considered because this machine learning approach often proves to be effective in high-dimensional classification or regression problems and provides an interesting variables importance ranking. The R package *randomForest* was used with parameter  $n_{tree} = 5000$ , and default values for the two other parameters,  $m_{try}$  (i.e. 160) and  $nodesize$  (i.e. 5).

method <sup>a</sup>	parameters setting	$RMSE_{CV}$	# var.selected
<i>lmCLV</i>	$\nu = 0.5, M = 25$	6.366	145
sPLSR	$\eta = 0.3, K = 5$	6.427	150
E-Net	$\alpha = 0.5, \lambda = 0.08$	6.335	90
L2-boost	$\nu = 0.2, M = 209$	6.742	75
RdF	(5000, 160, 5) <sup>b</sup>	9.992	146 <sup>c</sup>

<sup>a</sup> All the methods was applied using available R packages. Their parameters were determined on the basis of Cross-Validation with the same samples division.

<sup>b</sup>  $n_{tree} = 5000, m_{try} = 160, nodesize = 5$ .

<sup>c</sup> Variables with a standardized variable importance value greater than 3 were selected.

Table 1: Root Mean Squared Error in cross-validated prediction of the percentage of clementine juice added to the orange juices, according to various methods.

As shown in Table 3.2, *lmCLV* has an expected prediction performance similar to Sparse PLS Regression, Elastic-Net and the L2-boosting approach. Quite surprisingly, in this case study, Random Forest did not perform very well. In fact, one can observe that by Random Forest the prediction of the percentage of co-fruit added was overestimated or underestimated at the extremes of the experimental scale, i.e. for 0-10% or 50-60%.

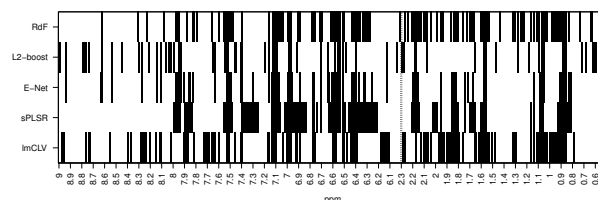


Figure 7: Location of the spectral variables involved in the models according to the method considered, in the orange juice authentication case study.

The number of variables involved in the model fitted on the whole dataset, using the predetermined model's parameters, is indicated in the last column of Table 3.2. As shown in Figure 7, while Sparse PLS Regression has the tendency to identify fewer but larger spectral ranges, the L2-boosting

approach retained a relatively small number of variables associated with narrow ranges. For *lmCLV*, 20 groups and their latent variable were identified. Since several small spectral ranges were merged within the same group, the total number of spectral variables involved was rather high. However, the order of extraction and the grouping effect, which are a specificity of *lmCLV*, cannot be revealed in Figure 7. Globally, 19 spectral variables were retained with the five methods considered herein. We systematically found the variables at 7.51–7.52 ppm, but not the other variables that belonged to the first group extracted with *lmCLV*. The variables between 7.09 to 7.12 ppm as well as the variable at 1.07 ppm are also present, as in the second group extracted with *lmCLV*. We finally noted that the areas at 6.96–6.98 ppm, 1.56–1.57 ppm, 0.86–0.88 ppm were selected with the five methods.

## 4 Conclusion

In this study, we introduced a strategy for linear model fitting based on the hypothesis that high-dimensional datasets often include highly correlated variables having similar effect on the response variable. This is specifically the case for modern scanning instruments such as those used in spectroscopy (infrared, near-infrared, Raman, NMR,...) which are able to collect a large number of sequential spectral variables, several of them being representative of one feature/signal. Omic data, and specifically metabolomic data, contain a large quantity of measured elements that are components of the same metabolic pathways and that constitute the biological information that is sought. The basis of the approach is to then cluster the explanatory variables first, as other authors have already proposed [19, 18]. In *lmCLV*, the clustering stage is based on the CLV method and consists of identifying unidimensional latent components which represent clusters of variables. These latent components play the role of the predictors in the second stage which results in introducing the explanatory variables in a group-wise fashion.

In contrast with [19] for the CRL approach (Cluster Representative Lasso), and [20] or [21] in which each cluster representative is defined as the mean variable of a group of variables, the CLV latent components are defined at the same time as the clusters, and are derived from the eigen decomposition of the within cluster covariance matrix. In addition, for the construction of the regression model itself, we have adopted an L2-boosting approach, rather than a Lasso approach. This makes it possible to build a simple and efficient algorithm in which the CLV dendrogram constitutes the base-learner.

Compared to our previous study [26], the algorithm proposed here requires much less computer resources. In the previous study, the clustering stage was performed on the data matrix combining the residuals of the response variable and the explanatory variables, and was repeated for each iteration. However, the clustering of several hun-

dred variables requires the largest part of the computation time. Using the new version of *lmCLV*, for the case study on orange juice authentication (Sect. 3.2), which involved 480 explanatory variables, the whole procedure ( $\nu = 0.5$ ,  $M = 100$ ) took 1 min 48 sec on one 3.4 GHz processor, including 1 min 25 sec for the clustering stage alone. For the implementation of this algorithm, a function *lmCLV* will be available in version 2.0.2 of the R package *ClustVarLV*.

In both examples presented in this study, *lmCLV* was shown to have similar predictive efficiency to other methods, and importantly, provided additional interpretability capacity. The use of latent components that are easy to interpret, because each of them is associated with a group of collinear variables, is consistent with the development of modern simplifying approaches for modelization. However, compared to Lasso-based methodologies, dimensionality reduction based on variables clustering makes it easier to identify directions of interest for prediction and makes it possible to highlight functional links between explanatory variables where they exist.

## Acknowledgements

The author is grateful to Freddy Thomas (Authenticity unit, Eurofins Analytics France, Nantes, France) for the opportunity to present the case study on orange juice authentication.

## References

- [1] T. Hastie, R. Tibshirani and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, (Second ed.). Springer Series in Statistics. New-York: Springer
- [2] R. Tibshirani (1996). Regression Shrinkage and Selection Via the Lasso, *J. Roy. Stat. Soc. B*, **58**, 267–288.  
<https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>,
- [3] G. Celeux, C. Maugis-Rabusseau and M.Sedki (2019). Variable selection in model-based clustering and discriminant analysis with a regularization approach. *Data Anal. Classif.*, **13**, 259–278.  
<https://doi.org/10.1007/s11634-018-0322-5>
- [4] Z.Y. Algamal and M.H. Lee (2019). A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification, *Adv. Data Anal. Classif.*, **13**, 753–771.  
<https://doi.org/10.1007/s11634-018-0334-1>
- [5] H. Zou and T. Hastie (2005). Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc.*



- B*, **67**, 301–320.  
<https://doi.org/10.1111/j.1467-9868.2005.00503.x>
- [6] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. B*, **67**, 91–108.  
<https://doi.org/10.1111/j.1467-9868.2005.00490.x>
- [7] M. Yuan and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Soc. Ser. B*, **68**, 49–67.  
<https://doi.org/10.1111/j.1467-9868.2005.00532.x>
- [8] J. Friedman, T. Hastie, and R. Tibshirani (2010). *A note on the group lasso and a sparse group lasso*. Technical report, Statistics Department, Stanford University.
- [9] B. Zeng, X. M. Wen and L. Zhu (2017). A linkfree sparse group variable selection method for single-index model. *J. Appl. Stat.*, **44**, 2388–2400.  
<https://doi.org/10.1080/02664763.2016.1254731>
- [10] Bondell, H. D. and B. J. Reich (2008). Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR. *Biometrics*, **64**, 115–123.  
<https://doi.org/10.1111/j.1541-0420.2007.00843.x>
- [11] L. Yengo, J. Jacques, C. Biernack, and M. Canoui (2016). Variable clustering in highdimensional linear regression: the R package CLERE. *R journal*, **8**, 92–106.  
<https://doi.org/10.32614/RJ-2016-006>
- [12] S. M. Curtis and S. K. Ghosh (2011). A bayesian approach to multicollinearity and the simultaneous selection and clustering of predictors in linear regression. *J. Stat. Theory Pract.*, **5**, 715–735.  
<https://doi.org/10.1080/15598608.2011.10483741>
- [13] S. Chakraborty and A. C. Lozano (2019). A graph laplacian prior for bayesian variable selection and grouping. *Comput. Stat. Data An.*, **136**, 72–91.  
<https://doi.org/10.1016/j.csda.2019.01.003>
- [14] I. T. Jolliffe, N. T. Trendafilov and M. Uddin (2003). A modified Principal Component technique based on the lasso. *J. Comput. Graph. Stat.*, **12**, 531–547.  
<https://doi.org/10.1198/1061860032148>
- [15] H. A. Chipman and H. Gu (2005). Interpretable dimension reduction. *J. Appl. Stat.*, **32**, 969–987.  
<https://doi.org/10.1080/02664760500168648>
- [16] T. F. Cox and D. S. Arnold (2018). Simple components. *J. Appl. Stat.*, **45**, 83–99.  
<https://doi.org/10.1080/02664763.2016.1268104>
- [17] H. Chun and S. Keles (2010). Sparse partial least squares for simultaneous dimension reduction and variable selection. *J. Roy. Stat. Soc. B*, **72**, 3–25.  
<https://doi.org/10.1111/j.1467-9868.2009.00723.x>
- [18] D. G. Enki, N. T. Trendafilov and I. T. Jolliffe (2013). A clustering approach to interpretable principal components. *J. Appl. Stat.*, **40**, 583–599.  
<https://doi.org/10.1080/02664763.2012.749846>
- [19] P. Bühlmann, P. Rütimann, S. van de Geer and C.-H. Zhan (2013). Correlated variables in regression: clustering and sparse estimation. *J. Stat. Plan. Infer.*, **143**, 1835–1858.  
<https://doi.org/10.1016/j.jspi.2013.05.019>
- [20] T. Hastie, R. Tibshirani, D. Botstein and P. Brown (2001). Supervised harvesting of expression trees. *Genom. Biol.*, **2**, 1–12.  
<https://doi.org/10.1186/gb-2001-2-1-research0003>
- [21] M. Y. Park, T. Hastie and R. Tibshirani (2007). Averaged gene expressions for regression. *Biostat.*, **8**, 212–227.  
<https://doi.org/10.1093/biostatistics/kxl002>
- [22] E. Vigneau and E. Qannari (2003). Clustering of variables around latent components. *Comm. Stat.-Simul Comput.*, **32**, 1131–1150.  
<https://doi.org/10.1081/SAC-120023882>
- [23] E. Vigneau and M. Chen (2016). Dimensionality reduction by clustering of variables while setting aside atypical variables. *Electron. J. Appl. Stat. An.*, **9**, 134–153.  
<https://doi.org/10.1285/i20705948v9n1p134>
- [24] E. Vigneau, M. Chen and E. M. Qannari (2015). ClustVarLV: An R package for the clustering of variables around latent variables. *R journal*, **7**, 134–148.  
<https://doi.org/10.32614/RJ-2015-026>

- [25] A. Figueiredo and P. Gomes (2015). Clustering of variables based on Watson distribution on hypersphere: A comparison of algorithms. *Comm. Stat. - Simul Comput.*, **44**, 2622–2635.  
<https://doi.org/10.1080/03610918.2014.901353>
- [26] M. Chen and E. Vigneau (2016). Supervised clustering of variables. *Adv. Data Anal. Classif.*, **10**, 85–101.  
<https://doi.org/10.1007/s11634-014-0191-5>
- [27] B. Efron. and T. Hastie (2016). *Computer age statistical inference: algorithms, evidence and data science*. New York: Cambridge University Press. <https://doi.org/10.1017/cbo9781316576533>
- [28] P. Bühlmann and T. Hothorn (2007). Boosting algorithms: Regularization, prediction and model fitting. *Stat. Sc.*, **22**, 477–505.  
<http://dx.doi.org/10.1214/07-STS242>
- [29] D. Karlis, G. Saporta and A. Spinakis (2003). A simple rule for the selection of principal components. *Comm. Stat. - Theor. M.*, **32**, 643–666.  
<https://doi.org/10.1081/STA-120018556>
- [30] B. Hofner, A. Mayr, N. Robinzonov and M. Schmid (2014). Model-based boosting in R: A hands-on tutorial using the R package mboost. *Comput. Stat.*, **29**, 3–35.  
<https://doi.org/10.1007/s00180-012-0382-5>
- [31] E. Vigneau and F. Thomas (2012). Model calibration and feature selection for orange juice authentication by 1H NMR spectroscopy. *Chemometr. Intell. Lab. Sys.*, **117**, 22–30.  
<https://doi.org/10.1016/j.chemolab.2011.05.006>
- [32] P. Rinke, S. Moitrier, E. Humpfer, S. Keller, M. Mørtter, M. Godejohann, G. Hoffmann, H. Schaefer and M. Spraul (2007). An 1H-NMR technique for high throughput screening in quality and authenticity control of fruit juice and fruit juice raw materials- SGF-profiling. *Fruit Process*, **1**, 10–18.
- [33] SDBSWeb (2020). Spectral Database for organic Compounds.  
<https://sdb.sdb.aist.go.jp> (National Institute of Advanced Industrial Science and Technology)
- [34] L. Breiman (2001). Random forests. *Mach. Learn.*, **45**, 5–32.  
<https://doi.org/10.1023/A:1010933404324>

# Performance of Malware Detection Classifier Using Genetic Programming in Feature Selection

Heba Al-Harabsheh, Mohammad Al-Shraideh and Saleh Al-Sharaeh.

King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

E-mail: Heba.moh.h@gmail.com, mshridah@ju.edu.jo and ssharaeh@ju.edu.jo

**Keywords:** malware detection, machine learning, feature selection, classifier, genetic programming

**Received:** September 11, 2021

*The term "malicious software," which is commonly referred to as malware, describes malicious software that affects or harms computers, servers, or networks. While the numbers and complexity of malware have rapidly increased, developing a malware detection system is required to detect malware in the world of cybersecurity and test the behavior of its new features. While traditional techniques provide less efficiency in detecting new malware, machine learning techniques are used to achieve rapid malware detection in an intelligent way to improve detection performance, as malware and its application in the industry are constantly increasing. In this study, we developed a malware detection model by detecting malware using machine learning classifiers, after passing a new feature selection technique using genetic programming. We also compared the performance of all classifiers using the most recent feature selection techniques. Results show that Random Forest, Random Forest (4), and Random Tree give the best value in all experiments, while Hoeffding Tree and Decision Stump give lower values for F1-score and accuracy in all experiments. The feature selection method that proposed GPMP gives a better value than Filter-based with little differences. The accuracy and F1-score have the values of 0.881066 and 0.867546 for GPMP, and the values of 0.877624 and 0.862894 for Filter-based, respectively. The experimental results reveal that GPMP used fewer features than Filter-based, and this affected the computation and complexity of the model.*

*Povzetek: Analizirane so bile metode strojnega učenja za povečanje uspešnosti odkrivanja zlonamerne programske opreme.*

## 1 Introduction

Nowadays, the problem of cybersecurity is growing due to the fact that all electronic devices are connected to the Internet. In addition, cybersecurity affects our daily life and the infrastructure of all fields because of the high connectivity between millions of hosts over the Internet.

Malware is considered eligible to modify the target device or application in order to gain full control of the unauthorized access, and the device can have access to other vulnerable devices to steal data.

The main reason of cyber-attack is malware. Accordingly, a malware detection technology must be developed to improve the legacy technology of the industrial security software used for detection. According to Kaspersky's research done in 2020, detecting new malicious files is increased by a rate of 5.2% every day [1].

Therefore, distinguishing between benign and malicious files is the most cybersecurity challenging task, which is used to detect suspicious files with higher accuracy and less time and cost. There are no highly efficient detection methods applied in the traditional methods because malware spreads very quickly on the network. Accordingly, most researchers try to use machine learning to get the best detection accuracy and

reflect it in the new technologies or tools designed for malware detection and network Intrusion [2] [3] [4].

In this paper, we propose a new model using feature selection method and genetic programming that are used in a set of parallel classifiers for a more accurate model to detect malware at the lowest cost. The model is run using five methods of selecting features across ten classifiers, then they will be compared to show the best result at the lowest cost.

## 2 Related studies

Recently, much attention has been given to finding and developing new methods of malware detection, compared to existing methods, to cover the gap of malware detection challenges that arise by the increase of malware over time [5].

Malware detection and analysis help the analyst learning the type, category, and target of malware. Malware detection can be classified into two categories, mainly: static and dynamic analysis. Static analysis is the primary category that analyzes malware and collects data from a file without executing it. Dynamic analysis is the opposite as it executes the suspicious file in an isolated and controlled environment [6].

There are many research papers done to develop malware detection methods. In [7], for example, a detection system using effective low-dimensional features has been proposed. This system used ensemble algorithms for analysis to get better performance. The model applies detection technology to a large number of malwares with faster detection time.

Another research [8] studies two categories of classification in one model. Alotaibi proposed Multi-Level Malware detection using Triad Scale (MLMTS) model that work in multi stages. The first two levels of his proposed method perform static analysis and the third level performs dynamic analysis. The linear regression in machine learning was used in this model as an input of each level. Using MLMTS method in research experiments increases the accuracy and decreases false alarming, compared to other recent models.

The study done by [9] focuses on improving an effective and efficient approach for malware detection by using the behavior of malware families. The authors proposed this methodology because they knew that the attacker could modify API call features with no change in overall behavior. So, they worked on three steps: studying API calls to object operation by analyzing the malware, generating a dependency graph based on the information of these operations, and finally defining the family

dependency graph for each malware. The evaluation results of the proposed approach showed that the approach can help some anti-virus companies to detect malware from a zero-day attack.

Multiple anti-virus scanners detection systems were proposed for enhancement selection performance in the work done by [10]. They proposed multiple anti-virus scanners that attempt to check if increasing the number of scanners affect detection results and how these scanners are able to maximize the accuracy. The experiment shows that there is a small effect of the number of scanners on accuracy, and if the number was increasing, the overall accuracy will be lowered rather than improved. Moreover, the final ranking of the scanners depends on the accuracy and gives the best chance to select the best combination of scanners.

The malware detection model in this study uses a specific feature selection method that is used in several classifiers to compare the scores in order to show the effect of contemporary feature selection on reducing the cost of training time in balanced and unbalanced datasets. The experimental results were obtained by comparing Precision, Recall, Accuracy, and F1-score in all classifiers and by comparing the commuting time as well.

The following Table 1 provides a summary of the related work done on this field of study.

Table 1: Summary of the Related Work.

Paper	Classifiers Algorithms	Features	Feature Selection Method	Result	Objective	Limitations
[6]	Chi-square	APIs/System calls	-	Detecting accuracy up to 96.56%	Proposing a model for recognizing and detecting the malware from benign.	The limitations of this model are related to malware that have an evasion detection technique, and it was used to detect 5 classes of malware only.
[11]	Evolutionary Algorithm	Malware OpCodes	-	Detecting accuracy for all datasets between 85.80% and 87.67%	Using Evolutionary Algorithm to generate graph and compare the similar graph to detect the suspicious files. It was used for categorizing malware and detecting it.	The study shows that the detection approach was used to categorize the malware and detect it, but it does not show if it can detect and cover all classes of malware.
[12]	Hidden Markov Model (HMM), Support Vector Machine (SVM), Decision Tree (J48), and Random Forest (RF)	API-call, operations, and usage system library	Used term and inverse term frequency (TF-IDF) Logarithm for feature extraction	Random Forest classifier gives the best results, while HMM has the lowest performance	Evaluating classification approaches in terms of distinctive dynamic features and finding the best dynamic features.	Malware detection approaches were used to obtain the family classification and malware detection.
[7]	AdaBoost, random forest, XGBoost, rotation trees, and extra trees.	2-gram, 2-gramM, API-DLL, API, and WEM	frequency analysis and Expert knowledge to select a relevant feature	XGBoost reaches the highest rank in AUC-PRC and accuracy	Developing a novel technique to reduce feature dimensionality.	The study does not represent the time used to extract features by frequency analysis and expert knowledge.

[8]	Proposed a model with multi-level linear regression (MLAPAM and MDMLA)	Call sequences, fallouts, and arguments	MLMTS method used to generate a feature set	The proposed method (MLMTS) gives the maximal accuracy and minimum false positive, compared to other methods	Building a model in a Multi-Level for Malware detection using Triad Scale (MLMTS) based on a regression coefficient.	The experiment study was performed using one benchmark malware dataset.
[9]	Comparing the object operation of feature dependency graph and family dependency graph	API call	-	The proposed model gives highly efficient and effective results.	Building a malware detection system based on behavior of the malware family.	The justification of using the behavior-based features and the graphs is time consuming.
[10]	Comparing a multi-scanner as a black box	Features extracted from the malware were not considered. Only the rates from the scanners were	-	Combining multi anti-virus scanners with achieving high accuracy, and the result is having the best combination of scanners	Proposing three models to achieve the best accuracy of multi-scanner detection system and minimize the scanning cost.	The internal mechanism is not clear, and it needs more details about the features and classifiers used in all scanners.
[13]	Gradient Boosting Algorithm	Malware OpCodes	Deep learning-based feature extraction method, word2vec	Detecting accuracy up to 96%.	Developing a model to represent malware that mainly uses the malware opcodes.	The work conducted was on a short range of malware classes. The paper covered 8 different malware classes.

### 3 Datasets information

This section presents all datasets used in our experiments conducted for this study. Our approach needed several datasets to study how they affect malware detection performance. All datasets used are available online.

We used two types of balanced and imbalanced datasets for malware detection domains. They were also categorized into two groups: malicious or benign software, each with a different number of instances and features.

Table 2 shows in detail all information regarding each dataset used in this study in terms of the number of features, the number of classes, the number of instances, characteristics of data, and the type of distribution datasets whether they were balanced or imbalanced.

#### 3.1 PE section headers

The "PE-section" header is a balanced dataset that was developed by Angelo Oliveira to extract dataset features from the "PE-section" portion of a group of PE malware and PE goodware files that appeared in Cuckoo Sandbox reports. This dataset was created for malware detection and classification purposes [14].

#### 3.2 Malware analysis datasets top1000 PE imports

Angelo Oliveira generated "TOP-1000 PE Imports" which is imbalanced dataset that was created from 'pe\_imports' part of Cuckoo Sandbox reports for a group of PE malware and PE goodware files [15].

#### 3.3 API call sequence

The imbalanced "API Call Sequence" dataset contains 42,797 malware and 1,079 goodware of API call sequences gathered by the extracted "calls" part of Cuckoo Sandbox reports [16].

#### 3.4 Malware detection data

This imbalanced dataset was created by Takbiri in June 2019 as a result of his study done on detecting malware using Low-level Architectural Features of malware [17].

#### 3.5 BIG malware dataset from Microsoft

Microsoft team created a balanced dataset from their competition for Malware Classification Challenge which is called "BIG 2015" [18].

Table 2: List of Used Datasets.

Dataset	Alias Name	# of Feature	# of Instances Used	# of Classes	Features Characteristics	Dataset Class Distribution
PE Section Headers	BS1	5	43293	2	Integer, Float, Text	Balanced
TOP-1000 PE Imports	DS2	1001	47580	2	Integer, Float, Text	Imbalanced
API Call Sequence	DS3	101	43876	2	Integer, Float, Text	Imbalanced
Malware Detection Data	DS4	16	70	2	Integer, Float, Text	Imbalanced
BIG Malware Dataset from Microsoft	DS5	69	5210	2	Integer, Float, Text	Balanced
CLaMP (Classification of Malware with PE headers)	DS6	55	5184	2	Integer, Float, Text	Balanced
Malware Executable Detection	DS7	531	373	2	Integer, Text	Imbalanced
Windows Malware Detection (REWEMA )	DS8	631	6271	2	Integer, Text	Balanced
Malware Classification	DS9	56	216352	2	Integer, Text	Imbalanced
Malware Goodware Dataset	DS10	27	50210	2	Integer, Float, Text	Imbalanced

### 3.6 CLaMP (Classification of Malware with PE headers)

The CLaMP balanced dataset is built from portable, executable files in header field values and from a combination of malware and benign samples to be used in the detection system [19].

### 3.7 Malware executable detection

Rumao created a dataset containing a set of features extracted from malware and goodware for Windows executable files. It blends two features of Windows executables: binary hexadecimal system calls feature, and DLL calls as hybrid features, in order to create this dataset. This imbalanced dataset contains 301 malicious programs, while the goodware contains 72 cases [20].

### 3.8 Windows Malware Detection (REWEMA)

Windows Malware Detection Dataset (REWEMA), as a balanced dataset, contains 3136 malicious programs and 3135 benign executable files. Features were extracted from disassembling executable files and selecting a set of useful file attributes [21].

### 3.9 Malware classification

Malware classification dataset uploaded to Kaggle website by Paul. Which is Imbalanced dataset, it contains 75503 malware and 140849 goodware features [22].

### 3.10 Malware goodware dataset

This dataset was uploaded to Kaggle in February 2021. This Imbalanced dataset contains 50210 instances features for malware and goodware files [23].

## 4 Method

### 4.1 Methodology design

The malware datasets described in Section 3 were collected to test the proposed method for the detection system. All ten datasets were classified and categorized into two categories of malware and benign software. In addition, these datasets have been further categorized into two other types: balanced and imbalanced datasets, and this categorization is based on the disproportion of the malware and benign category in each dataset.

Five feature selection techniques, which are described below in Section 4.2, were used in this study, and passed through fourteen machine learning classifiers in parallel. This objective model computes detection performance at the lowest cost. In our approach, we divided the ten datasets into a training and test set with percentages of 70% and 30%, respectively.

In this work, the model is designed and evaluated by making the following main steps: [1] Data cleaning was performed for all datasets before they are split for training and testing to fix all problems in the datasets (missing value, removing outliers, and resolving discrepancies, among others), and [2] five feature selection methods were used (Chi-Square, Filter-based, Wrapper-based, GPM, and GPMP). Then, [3] The number of features was selected for each feature selection method to compare performance, then it was calculated based on the number of features used in each method to test the performance based on how this method extract relevant features that reflect the effect in the overall performance of the discovery model. After that, [4] excessive oversampling SMOTE technique was applied in imbalanced datasets. [5] The release of new datasets was then introduced after applying feature selection and SMOTE methods in the classification model (14 classifiers) to measure

$$X^2 = \sum_i^n \left( \frac{(O_i - E_i)^2}{E_i} \right) \quad , i = 1, 2, \dots, n \quad (1)$$

$$GPM = \sum_i^k (W_{Fk})/k \quad , i = 1, 2, \dots, k \quad (2)$$

$$GPMP = \left( \sum_y^k (W_{Fk})/k \right) - \left( \sum_o^z (W_{Fk-low})/z \right) \quad , y = 1, 2, \dots, k \mid o = 1, 2, \dots, z \quad (3)$$

predictions. [6] The performance evaluation scale for this detection model was accuracy, F1, accuracy, and recall. [7] The rating scale was finally compared for all datasets in all feature selection methods and all classifiers as well.

The result of the model focuses on the performance to obtain the results of balanced and imbalanced datasets. All these steps were performed for the ten datasets (whether balanced or unbalanced) to study whether our proposed approach will obtain good performance in all datasets with different characteristics.

### 4.2 Feature selection.

In this work, two main steps were applied in datasets before running the feature selection technique.

### 4.3 Data cleaning

In this study, we applied a data cleaning for all datasets. It is about preparing raw data to start working on feature selection by drop outliers, cleaning missing values, encoding (text, integer, date, and float, among others), and scaling data [24].

#### 4.3.1 Using data augmentation technique

Synthetic Minority Over-sampling Technique (SMOTE) algorithm is one of the well-known augmentation techniques that are used in imbalanced datasets to solve minority class problems. In the imbalanced dataset, there are too few instances of minority classes that affect model decisions [25].

In this study, we used the SMOTE over-sampling technique to balance the number of classes in the datasets by adding new synthesized instances of the minority class. We also tested another SMOTE technology that is under-sampling by removing the random instances of the majority class, so that it is balanced against the minority class. However, the detection efficacy decreased because some datasets have too few minority classes which results in decreasing the dataset, and this will affect the training and testing phase. Therefore, the main augmentation technique that we used in this study for all imbalanced datasets is the SMOTE over-sampling technique [26].

#### 4.3.2 Feature selection techniques

In this part of our study, we used five methods for feature selection, where three of them were commonly used in machine learning, and they are: Chi-Square, filter-based, and wrapper-based. The remaining methods are Genetic Programming Mean (GPM) and Genetic Programming Mean Plus (GPMP). They were developed in our study

Table 3: Five Feature Selection Methods.

#	Feature Selection Method	Alias name
1	Chi-Square	Chi
2	Genetic programming Mean (GPM)	GPM
3	Genetic programming Mean Plus (GPMP)	GPMP
4	Filter-based	Filter
5	Wrapper-based	Wrapper

using genetic programming (GP) algorithm using the open-source frame-work HeuristicLab (Heuristic and Evolutionary Algorithms Laboratory) [27].

The GP method was used to create a weight for all features in hidden computations and to release the feature at relatively close values. We added two thresholds to the output result of the GP algorithm to find the most important and most relevant feature, in order to get more accuracy in perdition. In the first threshold used in GPM, the mean of all features values was computed, and all features were greater than the threshold.

In GPMP, we changed the threshold by adding a chance for the remaining features whose values are below the mean, and that was done by creating another interim threshold which was added to the original threshold value to add a change for the features where their values are near the original threshold. See equation (1) that defines Chi-Square, where O is the observed value and E is the expected value for all values.

Equation (2) represents the calculation of GPM method, Wfk is the weight for the feature, and the integer number K represents all features y=1, 2, ..., K.

Equation (3) is similar to equation (2), but it subtracts the mean of all weights of features under the total mean as an interim threshold is used to increase the chance for other features that have a value less than the original threshold.

The main difference between these methods is that when we apply them in our approach, we find that a number of some specific features affect the computational cost and model detection performance. Each method evaluated feature values and compared them to the target value to find the strongest relationship between the target values depending on method statistical measures.

Table 3 shows the five feature selection methods used in this study and their alias used in the charts.

We found that each method has its own set of features that are identified to be used in the detection model. The difference in the number of features and the identified features themselves will be certainly reflected in the final

Table 4: Number of features used for all feature selection methods.

Dataset	Number of Features used						Percentage of Features used				
	Chi-Square	GPM	GPMP	Filter-based	Wrapper-based	Total Feature NO	Chi-Square	GPM	GPMP	Filter-based	Wrapper-based
DB1	3	2	4	3	3	5	60%	40%	80%	60%	60%
DB2	948	802	829	113	518	1001	95%	80%	83%	11%	52%
DB3	100	20	33	99	29	101	99%	20%	33%	98%	29%
DB4	15	7	15	14	15	16	94%	44%	94%	88%	94%
DB5	55	12	20	50	61	69	80%	17%	29%	72%	88%
DB6	43	13	16	29	37	55	78%	24%	29%	53%	67%
DB7	483	70	70	133	201	531	91%	13%	13%	25%	38%
DB8	151	59	48	563	611	631	24%	9%	8%	89%	97%
DB9	54	15	18	34	46	56	96%	27%	32%	61%	82%
DB10	19	7	9	20	25	27	70%	26%	33%	74%	93%
<b>Total</b>							<b>79%</b>	<b>30%</b>	<b>43%</b>	<b>63%</b>	<b>70%</b>

results of the detection model. Table 4 shows the differences between the number of features identified in each method.

#### 4.4 Evaluation metrics

To evaluate our proposed detection model approach, we used the common evaluation metrics. These metrics are accuracy, precision, and recall, and we added F1-score because we tested two types of balanced datasets that can be measured using accuracy. In another hand, imbalanced datasets need to be measured using F1-score and accuracy. Equations from (4) to (7) show how these metrics are calculated [28].

F1-score mainly considers the values of both Precision and Recall, while Accuracy represents the percentage of the number of correct predictions in the model to the total number of inputs.

$$F1 - Score = \frac{2 * precision * recall}{precision + recall} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

## 5 Experimental results

In this section, we present the results of our experiment to evaluate the findings of detection over ten datasets.

Based on all experiments, we evaluated the detection model and summarized the results of the study in the conclusion section.

Table 5 shows 14 classifiers that were used in the proposed detection model after applying five feature selection methods in ten labeled malware datasets.

Based on the literature review examining the performance of classifiers, we used 14 classifiers shown in Table 5. We selected these classifiers depending on the efficiency of the literature review. We chose them based on 1) the most common classifier, 2) the least efficient classifier to test our approach, and 3) the most efficient classifier. The diversity of this chosen standardization helps us studying the proposed detection system. In our study, we applied our approach to build our model using four main steps: pre-processing for data cleaning, using augmentation technique for imbalanced datasets, using five-feature selection methods, and applying the data on

Table 5: Classifiers used in proposed model.

NO.	Classifiers	Alias name used in charts
1	Ada Boost.M1	AdaBM1
2	Ada Boost.M1 (4)	AdaBM1(4)
3	AdaBoost	AdaB
4	CatBoost	CatBoost
5	Decision Stump	DStump
6	Hoeffding Tree	HTree
7	k Nearest Neighbors	KNN
8	Naive Bayes	NB
9	Random Committee	RComm
10	Random Committee (4)	RComm4
11	Random Forest	RF
12	Random Forest4	RF4
13	Random Tree	RT
14	Support vector Machines	SVM



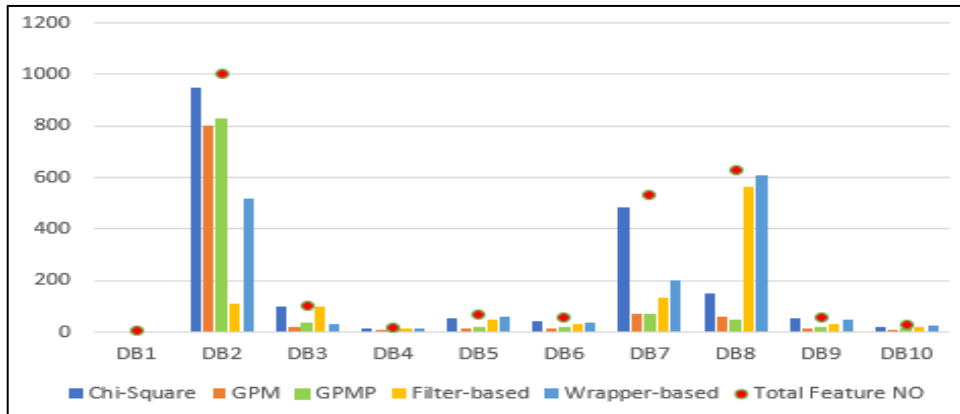


Figure 1: The number of features is used in all Datasets based on the FS methods.

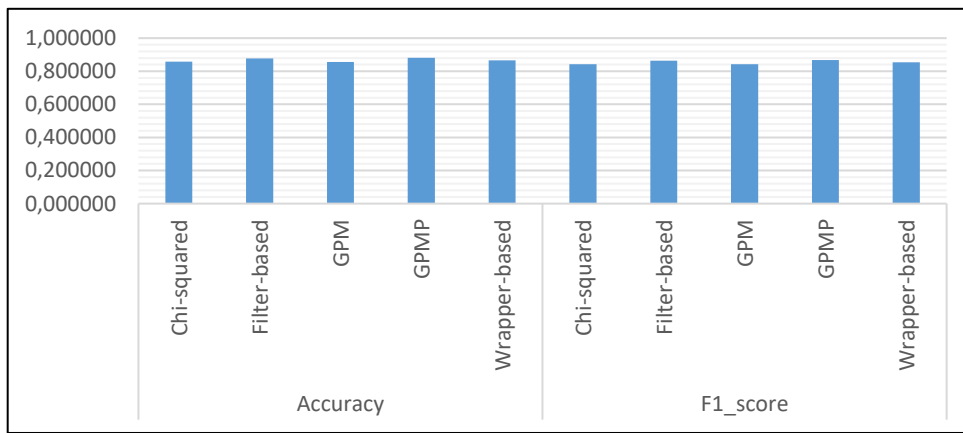


Figure 2: Average accuracy and F1-score summary for ten DS using 14 classifiers.

the model using 14 classifiers. The main objectives of this study focus on:

First: knowing if the new proposed feature selection methods affect the overall performance of the detection model.

Second: Knowing if the proposed methods give good performance of detection in balanced and imbalanced datasets.

Third: Determining which classifiers performs better using new FS methods and comparing them to other state-of-the-art performance methods.

Figure 1 shows the total number of features in all datasets compared to the number of features used in all FS methods in this study. As a Figure 1 appears almost in all datasets, chi-square and wrapper-based used many features in all datasets according to their calculation.

The proposed methods (GPM and GPMP) have a close result to the number of the used features, compared to filter-based. GPM and GPMP used fewer features than filter-based features in seven datasets. Table 4 shows the percentage of features used in ten datasets. GPM and GPMP have the minimum percentage of features used, with a value of 30% and 43%, respectively.

The highest number of features are used in Wrapper-based, in DS8 the percentage of features used are 97% that mean almost all the features are kept and used.

The highest number of features are used in Wrapper-based, and in DS8, the percentage of the used features are 97%. This means that almost all features are kept and used.

Based on the percentages shown in Table 4, and by applying FS on 14 classifiers, it can be noted, after conducting the initial analysis of the results, that the best results of F1-score and accuracy were found after applying the features that were selected by GPMP and Filter-based, with a little difference in values.

The first output of our results shows that the comparison between GPMP and Filter-based must be studied, while GPM gives less performance than these two FS methods.

This finding guided us to check if accuracy and F1-score were affected by these percentages. As shown in figures (3) to (12), the results of the experiment conducted for ten datasets show that we must study if these FS methods give the same performance in balanced and imbalanced datasets. Furthermore, we studied the overall behavior of the performance in all datasets, and we compared the values that were found in balanced and imbalanced datasets after applying SMOTE oversampling technique.

We noted, once we applied SMOTE augmentation technique, that prediction model is able to obtain the best performance based on F1-score and accuracy in the 14 classifiers that were used.

SMOTE is a common oversampling technique that is mainly used to handle the imbalanced datasets, but it may cause the model to need extra time for training and overfitting. However, in this study, oversampling technique

Table 6: Average of accuracy and F1-score for ten DS using 14 classifiers and five FS methods.

	GPM		Filter-based		GPMP		Chi-squared		Wrapper-based		Avg F1-score
	Accuracy	F1_score	Accuracy	F1_score	Accuracy	F1_score	Accuracy	F1_score	Accuracy	F1_score	
AdaBoost Avg	<b>0.897007</b>	<b>0.892153</b>	<b>0.950888</b>	<b>0.950875</b>	0.912717	0.909771	<b>0.913025</b>	<b>0.912015</b>	0.905135	<b>0.910262</b>	<b>0.915015</b>
AdaBoost.M1 Avg	0.877519	0.875979	0.931936	0.931577	<b>0.933636</b>	<b>0.933926</b>	0.897579	0.897521	<b>0.911156</b>	0.910161	<b>0.909833</b>
AdaBoost.M1(4) Avg	0.889907	0.887123	0.920886	0.920435	0.917216	0.920035	0.870939	0.868993	0.902101	0.901588	<b>0.899635</b>
CatBoost Avg	0.844995	0.854525	0.855918	0.855751	0.885714	0.885961	0.898815	0.898688	0.857265	0.857820	<b>0.870549</b>
Decision Stump Avg	0.797667	0.790254	0.793439	0.775139	0.819049	0.812602	0.752604	0.732342	0.771943	0.756560	<b>0.773379</b>
Hoeffding Tree Avg	0.519706	0.381524	0.587115	0.442341	0.548830	0.396545	0.526053	0.386072	0.623355	0.525731	<b>0.426442</b>
KNN Avg	<b>0.904014</b>	<b>0.901189</b>	<b>0.932180</b>	<b>0.932396</b>	<b>0.954862</b>	<b>0.953708</b>	<b>0.932422</b>	<b>0.934905</b>	0.862516	0.852418	<b>0.914923</b>
NB Avg	0.768505	0.736326	0.712549	0.670044	0.735392	0.705738	0.700059	0.648922	0.789419	0.769453	<b>0.706096</b>
Random Committee Avg	0.882569	0.880216	0.908151	0.908101	0.906350	0.906522	0.884793	0.884288	0.825369	0.792877	<b>0.874401</b>
Random Committee(4) Avg	0.877746	0.873598	0.870887	0.871200	0.871380	0.871454	0.887017	0.885551	0.861022	0.858908	<b>0.872142</b>
Random Forest Avg	<b>0.957570</b>	<b>0.955435</b>	<b>0.976959</b>	<b>0.976194</b>	<b>0.979496</b>	<b>0.979747</b>	<b>0.945723</b>	<b>0.944170</b>	<b>0.959321</b>	<b>0.962125</b>	<b>0.963534</b>
Random Forest(4) Avg	<b>0.950536</b>	<b>0.947880</b>	<b>0.975251</b>	<b>0.975478</b>	<b>0.980718</b>	<b>0.979334</b>	<b>0.948085</b>	<b>0.944361</b>	<b>0.966235</b>	<b>0.966801</b>	<b>0.962771</b>
Random Tree Avg	<b>0.948175</b>	<b>0.942662</b>	<b>0.972579</b>	<b>0.972934</b>	<b>0.976031</b>	<b>0.974188</b>	<b>0.939855</b>	<b>0.939396</b>	<b>0.965424</b>	<b>0.962871</b>	<b>0.958410</b>
SVM Avg	0.880227	0.872036	0.898003	0.898045	0.913536	0.916118	0.907813	0.907274	<b>0.915233</b>	<b>0.919567</b>	<b>0.902608</b>
Avg	<b>0.856867</b>	<b>0.842207</b>	<b>0.877624</b>	<b>0.862894</b>	<b>0.881066</b>	<b>0.867546</b>	<b>0.857484</b>	<b>0.841750</b>	<b>0.865392</b>	<b>0.853367</b>	

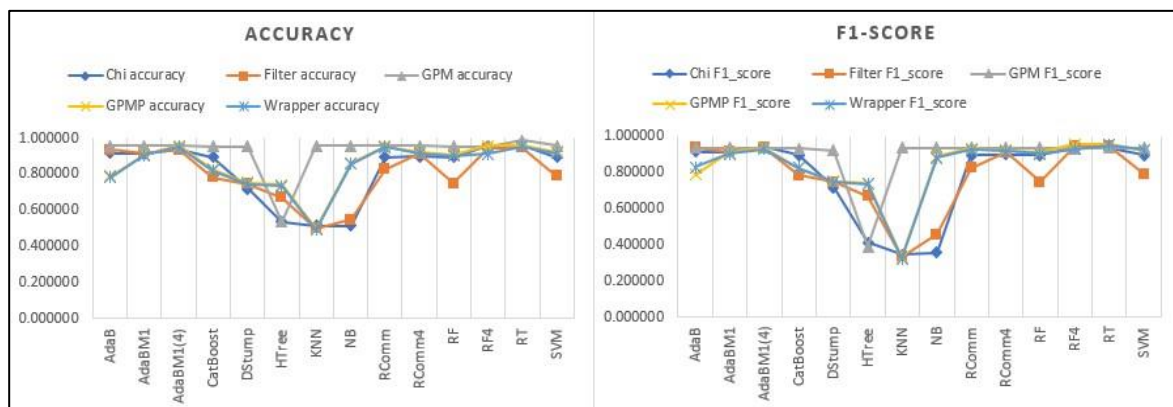


Figure 3: Accuracy and F1-score for DS1.

helps the model to give better performance when compared to balanced datasets.

Figures (3) to (12) illustrate the performance of all of our study objectives. In general, we can see that the balanced and imbalanced datasets are illustrated in similar shapes with little detailed differences occurred after applying SMOTE technique. This means that FS methods have a good result in all datasets regardless of whether they are balanced or imbalanced.

In the final step of our study, we tried to determine which classifier gives better detection performance using the five FS features over ten datasets (balanced and imbalanced).

After applying our approach on ten datasets, results were summarized by computing the average values for F1-score and accuracy for all experiments, as shown in Table 6 and Figure 2. The average of the highest calculated values of F1-score and accuracy shows that it is significant to rank the classifiers based on the efficiency.

We found that there were three datasets that held the best ranks in the average of all conducted experiments. Random Forest, Random Forest (4), and Random Tree are in the lead in accuracy and F1-score values. They are then followed by the other three classifiers, classified as group B of performance, namely: AdaBoost, AdaBoost.M1, and KNN. Additionally, both Hoeffding Tree and Decision Stump give the lowest values of F1-score and accuracy in all experiment. The remaining classifiers are categorized in the middle of giving good performance results scales.

Figure 2 summarizes the average values of accuracy and F1-score for ten DS using 14 classifiers. The average values for all experiments help us concluding our study by saying that GPMP and Filter-based give the best results in all experiments with the average of f1-score values that reach 0.867546 and 0.862894, respectively.

This finding leads us to examine the differences between FS methods. Figure 1 shows the number of features used in all datasets based on FS methods. The

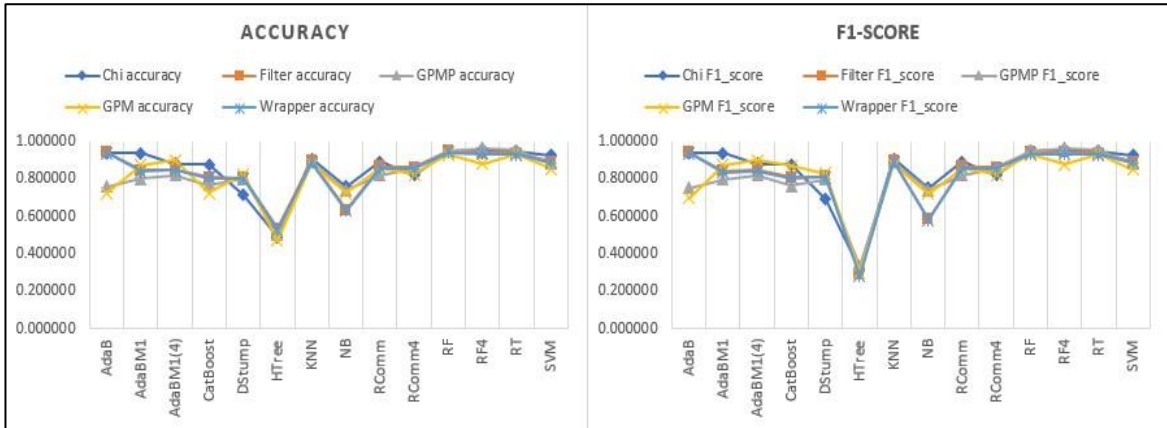


Figure 4: Accuracy and F1-score for DS2.

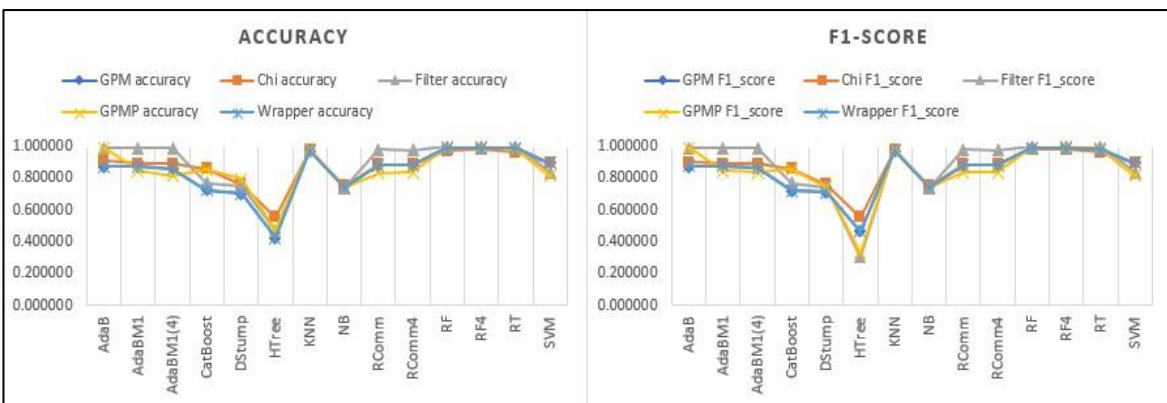


Figure 5: Accuracy and F1-score for DS3.

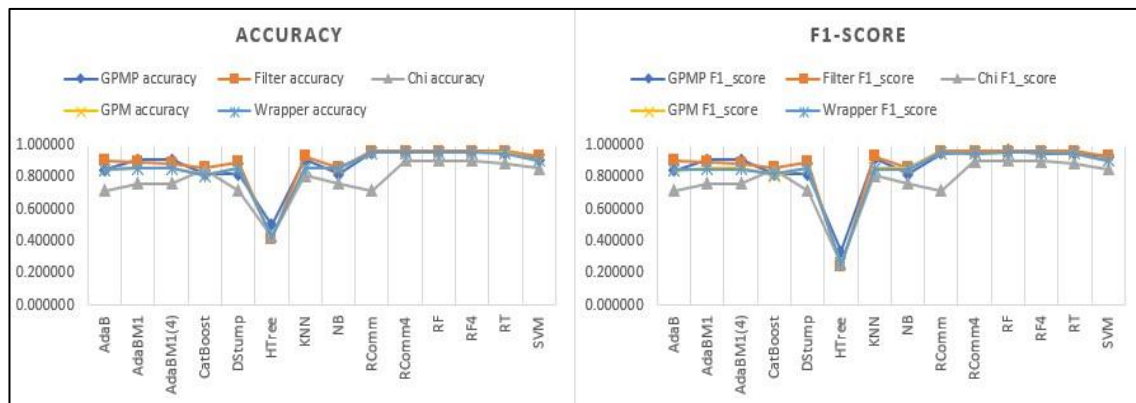


Figure 6: Accuracy and F1-score for DS4.

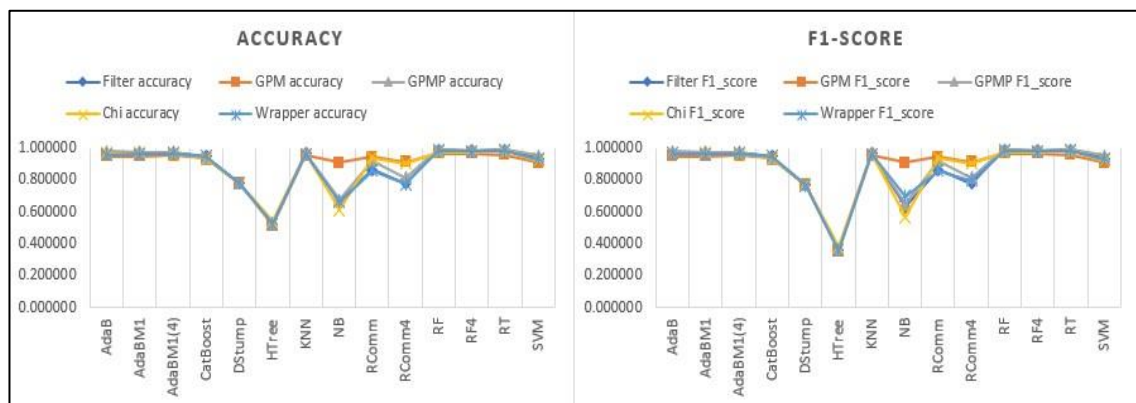


Figure 7: Accuracy and F1-score for DS5.

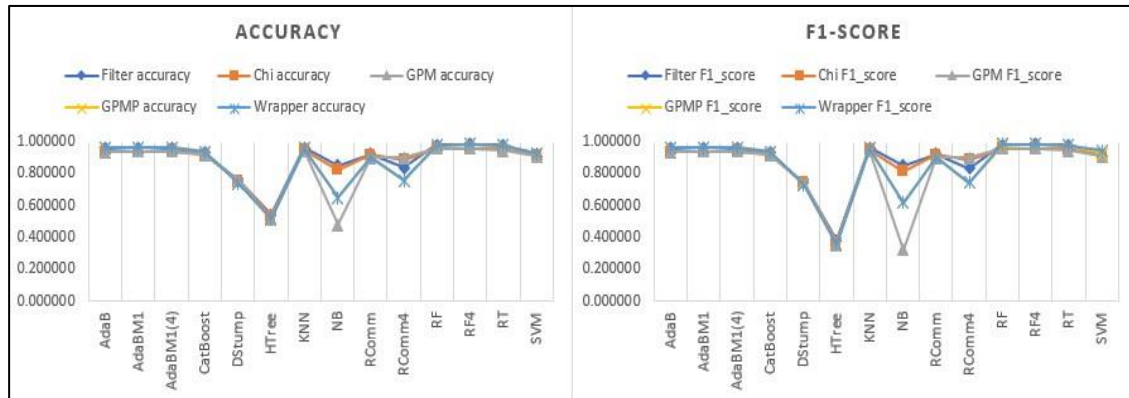


Figure 8: Accuracy and F1-score for DS6.

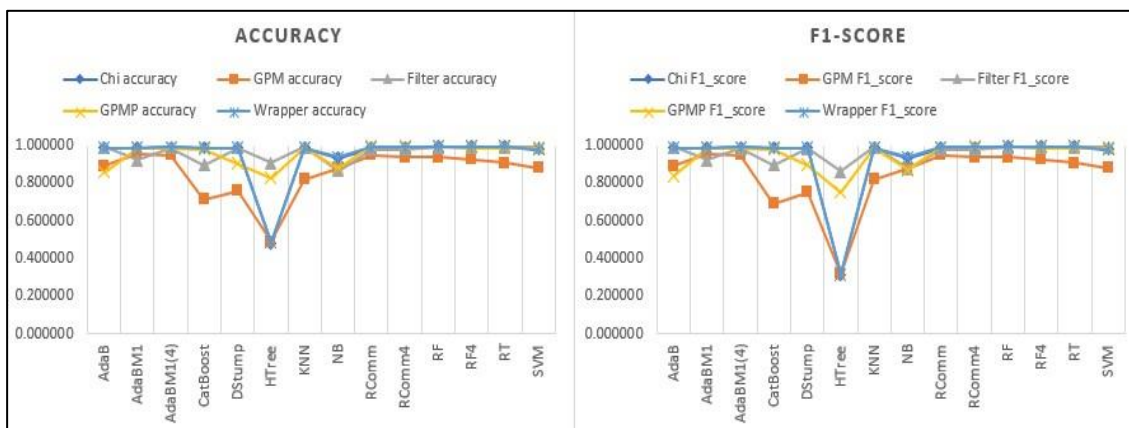


Figure 9: Accuracy and F1-score for DS7.

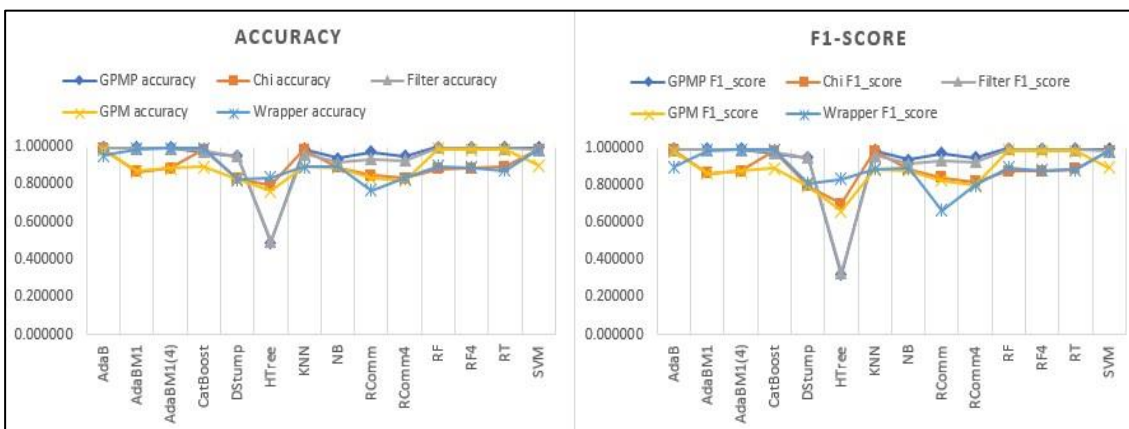


Figure 10: Accuracy and F1-score for DS8.

Figure 1 shows that in most of the datasets, the GPMP used fewer features than Filter-based. This means that the computation used in the model used less time in GPMP than on Filter-based.

Figures (3) to (12) show the F1-score and accuracy of all datasets. The analysis of the figures values shows the same results summarized in Table 6. In all figures, Random Forest, Random Forest (4), and Random Tree are at the top of all experiments. The values of AdaBoost.M1, and KNN are approximately similar, but the values of the Hoeffding Tree and the decision stump are shown in all figures below. These findings can be generalized for all

datasets, whether they are balanced or imbalanced, as previously discussed.

To check the effectiveness of our study, we have implemented our model on ten datasets to get the big picture of our study and the reasons why the proposed model is more effective and efficient.

It is difficult to compare the results of the proposed model with other models because most of the models use a limited number of malware detection features and because there are other limitations such as using a single dataset to make a comparison between the results. This study also covers both balanced and imbalanced datasets and applies the proposed model to them. Most of the

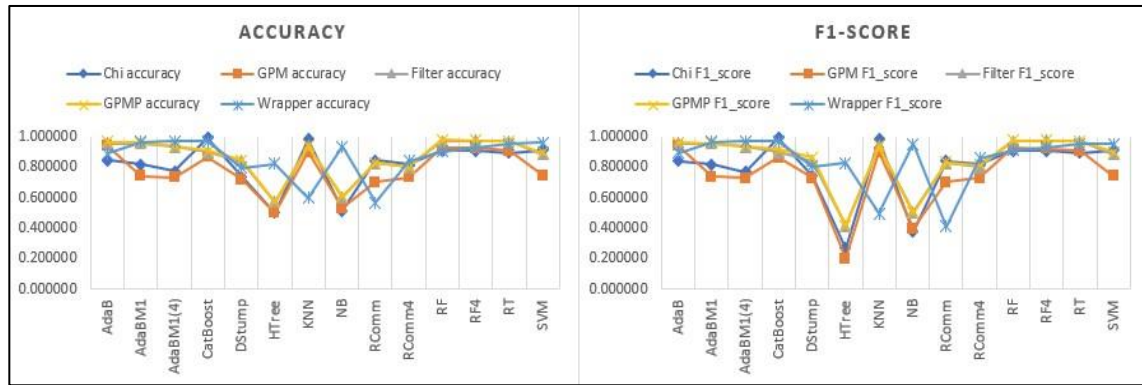


Figure 11: Accuracy and F1-score for DS9.

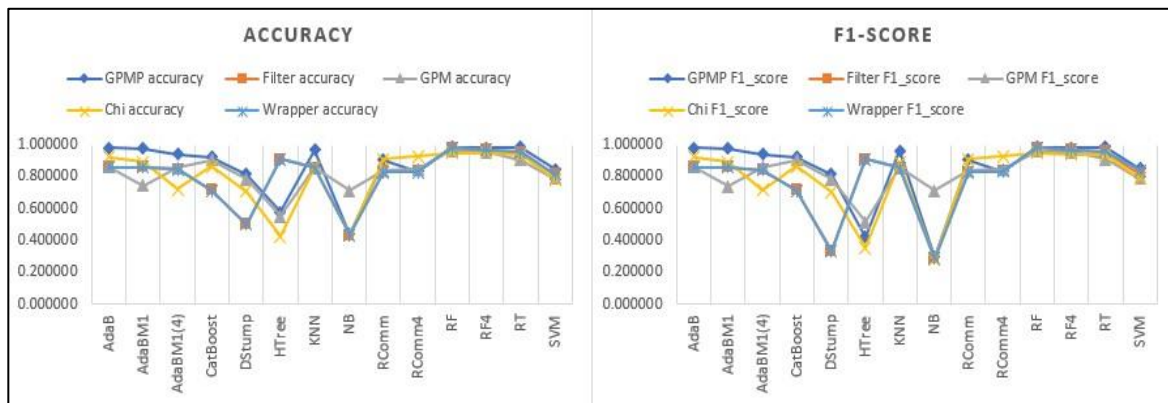


Figure 12: Accuracy and F1-score for DS10.

related works measure accuracy as a performance measurement, but our study does the measures using accuracy and F1-score because we use an imbalanced dataset. However, the results of the proposed model can be evaluated along with other related works by checking the result of F1-score of 0.9635 while we use Random Forest in the average of ten datasets, and this is considered a good value for the detection rate.

We have proposed a malware detection model using 14 classifier algorithms and five feature selection methods, two of which are proposed. Our feature selection methods are compared to other recent methods by applying them to the same datasets to check the differences in accuracy. We found our proposed method to be very effective for distinguishing between benign and harmful programs in relation to their detection.

## 6 Conclusion

This paper presents a model for detecting malware to enhance the detection rate by using five feature selection methods in ten malware datasets and 14 classifiers.

This study examines if this proposed detection method gives better detection value for balanced and imbalanced datasets. The experiments shown throughout the study have no difference in detection values while using balanced and imbalanced datasets after applying SMOTE overfitting technique in imbalanced datasets.

The results of this experiment have confirmed that the proposed GPMP feature selection methods attained high detection values in accuracy and F1-score.

The overall rankings of feature selection methods depending on accuracy and F1-score in this experiment are GPMP, Filter-based, Wrapper-based, and chi-square, respectively.

Results show that GPMP methods used fewer features than other methods with a percentage of 43% in the average of ten datasets. Filter-based that compete GPMP in detection rate used 63% features in an average of ten datasets. This shows how Filter-based affects the complexity and computation in the detection model. The average values of detection rate summarize the performance when using FS methods by saying that GPMP and Filter-based give average F1-score values of 0.867546 and 0.862894, respectively.

The final findings in this study focus on performance ranks for 14 classifiers in an average of all experiments. Random Forest, Random Forest (4), and Random Tree have the highest experiment results in accuracy and F1-score values. The values for these classifiers in F1-score are 0.963534, 0.962771, and 0.958410, respectively.

These values are followed by the values of AdaBoost, AdaBoost.M1, and KNN, while Hoeffding Tree and Decision Stump in all experiments give lower values for F1-score and accuracy.

We intend, in our future work, to apply this presented method in this model on android malware detection in order to study the features of the datasets and the performance of classifiers.

## Reference

- [1] “The number of new malicious files detected every day increases by 5.2% to 360,000 in 2020 | Kaspersky.” [https://www.kaspersky.com/about/press-releases/2020\\_the-number-of-new-malicious-files-detected-every-day-increases-by-52-to-360000-in-2020](https://www.kaspersky.com/about/press-releases/2020_the-number-of-new-malicious-files-detected-every-day-increases-by-52-to-360000-in-2020) (accessed Jun. 14, 2021).
- [2] Y. Jian, X. Dong, and L. Jian, “Detection and recognition of abnormal data caused by network intrusion using deep learning,” *Inform.*, vol. 45, no. 3, pp. 441–445, 2021, doi: 10.31449/inf.v45i3.3639.
- [3] O. F.Y, A. J.E.T, A. O, H. J. O, O. O, and A. J, “Supervised Machine Learning Algorithms: Classification and Comparison,” *Int. J. Comput. Trends Technol.*, vol. 48, no. 3, pp. 128–138, 2017, doi: 10.14445/22312803/ijctt-v48p126.
- [4] A. Chaudhuri, “Parallel fuzzy rough support vector machine for data classificatin in cloud environment,” *Inform.*, vol. 39, no. 4, pp. 397–420, 2015.
- [5] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, “A Survey on Machine Learning Techniques for Cyber Security in the Last Decade,” *IEEE Access*, vol. 8, no. 01, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.
- [6] O. Savenko, A. Nicheporuk, I. Hurman, and S. Lysenko, “Dynamic signature-based malware detection technique based on API call tracing,” *CEUR Workshop Proc.*, vol. 2393, pp. 633–643, 2019.
- [7] S. Euh, H. Lee, D. Kim, and D. Hwang, “Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems,” *IEEE Access*, vol. 8, pp. 76796–76808, 2020, doi: 10.1109/ACCESS.2020.2986014.
- [8] S. S. Alotaibi, “Regression coefficients as triad scale for malware detection,” *Comput. Electr. Eng.*, vol. 90, no. December 2019, p. 106886, 2021, doi: 10.1016/j.compeleceng.2020.106886.
- [9] B. Cheng et al., “MoG: Behavior-Obfuscation Resistance Malware Detection,” *Comput. J.*, vol. 62, no. 12, pp. 1734–1747, 2019, doi: 10.1093/comjnl/bxz033.
- [10] M. N. Sakib, C. T. Huang, and Y. D. Lin, “Maximizing accuracy in multi-scanner malware detection systems,” *Comput. Networks*, vol. 169, p. 107027, 2020, doi: 10.1016/j.comnet.2019.107027.
- [11] F. Manavi and A. Hamzeh, “A new approach for malware detection based on evolutionary algorithm,” *GECCO 2019 Companion - Proc. 2019 Genet. Evol. Comput. Conf. Companion*, pp. 1619–1624, 2019, doi: 10.1145/3319619.3326811.
- [12] A. G. Kakisim, M. Nar, N. Carkaci, and I. Sogukpinar, *Analysis and evaluation of dynamic feature-based malware detection methods*, vol. 11359 LNCS. Springer International Publishing, 2019.
- [13] C. H. Lo, T. C. Liu, I. H. Liu, J. S. Li, C. G. Liu, and C. F. Li, “Malware classification using deep learning methods,” *Proc. Int. Conf. Artif. Life Robot.*, vol. 2020, pp. 126–129, 2020, doi: 10.5954/ICAROB.2020.OS4-4.
- [14] “Malware Analysis Datasets: PE Section Headers | Kaggle.” <https://www.kaggle.com/ang3loliveira/malware-analysis-datasets-pe-section-headers> (accessed Mar. 07, 2021).
- [15] “Malware Analysis Datasets: Top-1000 PE Imports | IEEE DataPort.” <https://iee-dataport.org/open-access/malware-analysis-datasets-top-1000-pe-imports> (accessed Mar. 07, 2021).
- [16] “Malware Analysis Datasets: API Call Sequences | IEEE DataPort.” <https://iee-dataport.org/open-access/malware-analysis-datasets-api-call-sequences> (accessed Mar. 07, 2021).
- [17] “Windows Malware Detection | Kaggle.” <https://www.kaggle.com/sidneylima/rewema> (accessed Mar. 07, 2021).
- [18] Microsoft, “Microsoft Malware Classification Challenge (BIG 2015) | Kaggle,” 2018. <https://www.kaggle.com/c/malware-classification/data>. (accessed Mar. 07, 2021).
- [19] A. Kumar, “ClaMP (Classification of Malware with PE headers),” vol. 1, 2020, doi: 10.17632/XVYV59VWVZ.1.
- [20] “Malware Executable Detection | Kaggle.” <https://www.kaggle.com/piyushrumao/malware-executable-detection> (accessed Mar. 07, 2021).
- [21] “GitHub - rewema/REWEMA.” <https://github.com/rewema/REWEMA> (accessed Mar. 07, 2021).
- [22] “Malware Classification | Kaggle.” <https://www.kaggle.com/kallolkumarpaul/malware-classification> (accessed Mar. 07, 2021).
- [23] “Malware Goodware Dataset | Kaggle.” <https://www.kaggle.com/arbazkhan971/malware-goodware-dataset> (accessed Mar. 07, 2021).
- [24] N. Iqbal and M. Islam, “Machine learning for dengue outbreak prediction: A performance evaluation of different prominent classifiers,” *Informatica*, vol. 43, no. 3, 2019, doi: 10.31449/inf.v43i3.1548.
- [25] S. A. Alsaif and A. Hidri, “Impact of data balancing during training for best predictions,” *Inform.*, vol. 45, no. 2, pp. 223–230, 2021, doi: 10.31449/inf.v45i2.3479.
- [26] J. L. P. Lima, D. MacEdo, and C. Zanchettin, “Heartbeat Anomaly Detection using Adversarial Oversampling,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2019-July, no. July, pp. 1–7, 2019, doi: 10.1109/IJCNN.2019.8852242.
- [27] A. Elyasaf and M. Sipper, “Software review: The HeuristicLab framework,” *Genet. Program. Evolvable Mach.*, vol. 15, no. 2, pp. 215–218, 2014, doi: 10.1007/s10710-014-9214-4.
- [28] E. Amer and I. Zelinka, “A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence,” *Comput. Secur.*, vol. 92, 2020, doi: 10.1016/j.cose.2020.101760.

# Modeling and Performance Analysis of Resource Provisioning in Cloud Computing using Probabilistic Model Checking

Hichem Debbi

Department of Computer Science, University of M'sila, M'sila, Algeria

E-mail: hichem.debbi@univ-msila.dz

**Keywords:** cloud computing, IaaS, performance analysis, resource provisioning, probabilistic model checking, PRISM

**Received:** September 14, 2020

*Cloud computing consists of an advanced set of technologies that allow cloud providers to offer computing resources such as infrastructure, platforms and applications to be accessible over the Internet as services. Cloud computing relies on virtualization of resources in the cloud data centers, where a set of Virtual Machines (VMs) are deployed on Physical Machines (PMs) to provision and serve user requests. Due to the dynamic nature of cloud environments and complexity of resources virtualization, as well as the diversity of user's requests, developing effective techniques to evaluate and analyze the performance of cloud centers has become highly required. In this paper, we propose the use of probabilistic model checking as an effective framework for the evaluation and the performance analysis of resource provisioning in the cloud. Based on an analytical model for resource provisioning in Infrastructure-as-a-Service (IaaS) cloud, we build a stochastic model using the probabilistic model checker PRISM and analyze it against a useful set of probabilistic and reward properties that help to measure and analyze cloud performance in an efficient way.*

*Povzetek: Analizirane so razne komponente računanja v oblaku, npr. modeliranje in performance virov.*

## 1 Introduction

Cloud computing is a novel information technology that provides access to different IT services on demand over the Internet. The services provided through the cloud range into three main categories: Infrastructure as a Service (IaaS), where infrastructure resources such as: servers, storage, network components are provisioned. Platform as a Service (PaaS), which provides an environment for developing, running and managing applications efficiently by reducing the complexity related to infrastructure. Software as a Service (SaaS), which represents the largest cloud market, in which the task of managing software is moved to third-party services. Cloud computing has been treated from different aspects such as: security [22], load balancing [24], storage[7] and consistency [21].

In cloud computing literature, we refer usually to service providing by the technical term, provisioning. In this regard, Vaquero et al. [23] defined cloud as: the provision of computing infrastructure, which aims to shift the location of the computing infrastructure to the network in order to reduce the costs associated to management and maintenance of hardware and software resources. These resources are offered to the customer by cloud providers based on specific legally binding contracts called Service Level Agreements (SLAs), which state Quality of Service (QoS) parameters, such as time, cost, availability and security that should be guaranteed by service providers in or-

der to meet customer's needs and execute service requests. Buyya et al. [5] defined the cloud as: "A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service level agreements (SLA) established through negotiation between the service provider and the customers".

In IaaS cloud, virtualization plays a crucial role in enabling cloud computing services, in fact, it is a principal mechanism that enables cloud providers to cope with multiple requests of users through virtualization of physical machines(PMs). Virtualization refers to the abstraction of computing resources in a way that a single physical machine can run a set of virtual machines(VMs)[3].

However, due to dynamic nature of cloud computing environments and the complexity related to managing infrastructure resources from a side, and the diversity in customers requests from another, addressing the effective ways to instantiate, provision and deploy infrastructure resources to handle user requests and meet QoS requirements is considered as a big challenge and very critical issue in cloud computing. Therefore, performance analysis and evaluation of cloud computing environments have attracted recently much attention and formed an active research area.

Cloud performance analysis is beneficial for both cloud providers and consumers because it helps to get a deep insight on the infrastructure resources and how they should

be provisioned and scaled to execute various customers requests. Among various performance and evaluation methods, analytical modeling-based methods represent the major research that has been done in this area [8]. Since resources provisioning and usage is highly variable and uncertain, and since the arrival of customer requests is stochastic, these methods are stochastic in general, and employ queuing theory with different buffers to cope with a large number of requests given the available resources, and thus, performance measures are quantified using probabilistic methods. These stochastic methods can effectively capture the uncertainty beyond cloud provisioning behavior and estimate perfectly cloud metrics. Hence, SLA can be maintained and an overall optimization can be achieved. Continuous-time Markov Chains (CTMC), Stochastic Reward Net (SRN) and Stochastic Petri Nets (SPN) are all stochastic models that have been used for modeling and analyzing cloud services performance, and they showed promising results.

Performance behavior in the cloud is affected by a large set of parameters, and thus many system variables must be introduced to capture every modeling detail. CTMC models represent a good candidate to model every detail of the system [12, 17]. However, as the system variables under modeling grow up, the analysis could be intractable, since it results in a very large state space, which is known as the state explosion problem. To cope with such a problem, a solution based on decomposing the entire model into small interacting sub-models is proposed to facilitate and speed-up the model generation [12].

Bradley et al. [4] stated that symbolic approaches are very useful for performance and resilience modeling and analysis of massive stochastic systems, and thus they are very suitable for state space representation for cloud computing systems. Symbolic techniques such as Multi-terminal Binary Decision Diagrams are efficiently used to encode CTMCs, and enable steady-state and transient analysis. These techniques are efficiently employed by the probabilistic model checker PRISM [14], whose language features synchronization between modules. These advantages make PRISM a suitable tool for the performance analysis of cloud computing systems.

In this paper, we aim to show how probabilistic model checking can be used for the performance analysis and evaluation of IaaS cloud based on analytical modeling methods using CTMCs. Probabilistic model checking has appeared as an extension of model checking for analyzing systems that exhibit stochastic behavior. These systems are described usually using Discrete-Time Markov Chains (DTMC), Continuous Time Markov Chains (CTMC) or Markov Decision Processes (MDP), and verified against properties specified in Probabilistic Computation Tree Logic (PCTL) [13] or Continuous Stochastic Logic (CSL) [1, 2].

Using the probabilistic model checker PRISM [14], we show that analytical models, even if they are composed of many interacting sub-models, can be easily expressed in

PRISM language and analyzed in an efficient way. The entire model can be generated from interacting sub-models in reasonable time thanks to many numerical solution methods employed by PRISM that can deal perfectly with the state explosion problem. In this paper, we chose the model proposed by [12] as a case study. With probabilistic model checking, we will not be able only to compute probabilities related to QoS metrics, but also we can verify such safety properties and analyze reward-based properties.

The rest of this paper is organized as follows. In Section 2 we present some related works to cloud performance analysis. Section 3 presents some preliminaries and definitions on PRISM language. In section 4, we present the analytical model and its implementation in PRISM with detailed analysis of probabilistic and reward properties. Finally, we conclude the paper in Section 5.

## 2 Related work

The performance analysis and evaluation of cloud computing services can be performed through two ways: measurement-based methods and analytical modeling-based methods. In measurement-based methods [19], both cloud services and performance metrics to be evaluated should be known in prior, and then the benchmark to be tested should be chosen accordingly. After that, the testing experiments can be executed. Actually, this technique suffers from extensive experiments that should be executed with different workloads and system configurations, which may make the construction of appropriate testbeds that can represent realistic cloud services scenarios a costly task. Despite that, some measurements become invalid when cloud service providers upgrade their software and hardware to enhance their services. Therefore, analytical modeling-based methods are considered as a good alternative since they are of low cost, and can cover large parameters of cloud services, especially that these methods can analyze features of services even in early stages of design.

Li et al. [20] addressed the analysis of cloud services by modeling the service as a queuing network consisting of two tandem servers, web server and service server. After service completion at the level of the web server, the request either exits the network or continues to be executed at the service server. Both servers are modelled as  $M/M/1$  queue with an exponential distribution of arrival and service times. The main metric under evaluation in this paper was response time. Based on this measure, a relationship between the number of customers, the minimal service resources and the highest level of services can be easily derived. However, this work lacks an important feature in cloud computing modeling, which is virtualization.

Chen et al. [6] have also considered queuing network to estimate two different performance metrics, which are practically needed more in the context of cloud computing, request completion time (ECT) and rejection probability (RP). The authors in this work consider also two



queues, admission queue, and PM queue. Visualization is addressed considering many parameters, such as buffer size of queues, number of virtual machines, number of physical machines and error/recovery rates. In this work, each job is denoting a VM instance, and each VM is deployed on a single PM.

While previous works assume an exponential distribution of requests, considering heterogeneity in cloud modeling is actually more appropriate to better analyze some dynamic properties. In this regard, Khazaei et al. [16] introduced an embedded Markov model as an approximate analytical model based on  $M/G/m/m+r$  queuing system with single task arrivals and a task buffer of finite capacity. By solving the approximate model, complete probability distribution of the request response time, and important performance indicators such as the mean number of tasks in the system, the blocking probability, and the probability of immediate service can be easily estimated.

Covering more cloud services parameters by performance evaluation model is highly needed. However, sometimes the analysis of such a model tends to be intractable. To deal with this issue, some works [12, 17] proposed a solution based on interacting stochastic sub-models of Continuous-time Markov Chain (CTMC), thus, quantifying performance metrics can be realized in a scalable manner. In [12], the main QoS addressed was service availability and provisioning response delays. The requests or jobs submitted by users can be served in different pools named (hot, warm and cold), the decision in which pool the request should be served is made by a module called resource provisioning decision model, which is a CTMC model consisting of a queue with finite length. Another queue is found at each PM, where some requests/jobs can wait for VM provisioning. While this model is limited to service requests with a single task, Kazai et al. [17] proposed a similar solution, but capable of dealing with batch arrival of requests, where multiple VMs can be provisioned to handle multi-tasks based on a single service request, thus realizing a high degree of visualization.

Probabilistic model checking has already been used for modeling and analysis of cloud computing. Kikuchi and Matsumoto [18] have used PRISM for the performance modeling and analysis of concurrent live migration operations in cloud computing systems. Live migration plays a crucial role in cloud virtualization since it guarantees transporting VMs from a host to another without affecting the performance of the services. The authors described the performance model of concurrent VM live migration operations as a CTMC in PRISM language, and it has been verified against two main quantitative properties regarding the operations that can be stacked in waiting state at sender side, and the operations that are executed at server side.

In [15], the authors defined an interesting set of resource usage patterns in PRISM language as an MDP, and then introduced a set of reward-based properties for analyzing cost variation, and min/max probabilistic properties to analyze deployment's resource usage. These probabilistic patterns

before being generated as MDPs, are first expressed in a higher language called probabilistic pattern modeling language (PPM).

Evangelidis et al. [9] addressed performance modeling and formal verification of auto-scaling policies in PaaS and IaaS to provide performance guarantees to reduce SLAs violations, where two cloud services providers Amazon EC2 and Azure have been considered. The authors considered rule-based auto-scaling policies, where upper and/or lower bound on performance metrics such as CPU are expressed. The dynamics of auto-scaling process are expressed in PRISM as DTMC, and verified against probabilistic properties to estimate CPU utilization and response time violation for each auto-scaling policy, thus refining QoS violation thresholds for the policies.

We summarize the existing related work in Table 1

### 3 PRISM

PRISM is a tool used for formal modeling and analyzing systems that exhibit random or probabilistic behavior [14]. It supports several types of probabilistic models such as DTMCs, CTMCs and MDPs. The analysis is performed on these models against properties specified in PCTL logic [13] for DTMCs and MDPs and Continuous Stochastic Logic (CSL) [1, 2] for CTMCs. PRISM uses several numeric methods for model analysis such as Gauss-Seidel method, Backwards Gauss-Seidel method and Jacobi method. For MDPs and CTMCs, PRISM uses value iteration and uniformisation, respectively. As additional features, PRISM offers a simulation framework for reasoning about probabilities and rewards.

A model in PRISM consists of one or several modules that interact with each other. The module is specified using PRISM language as a set of guarded commands.

$$[\langle action \rangle] \langle guard \rangle \rightarrow \langle updates \rangle$$

Where the guard is a predicate over the variables of the system and the updates describe probabilistic transitions that the module can make if the guard is true. These updates are defined as follows:

$$\langle prob \rangle : \langle atomicupdate \rangle + \dots + \langle prob \rangle : \langle atomicupdate \rangle$$

When representing CTMCs,  $\langle prob \rangle$  will refer to transition rates instead of discrete probabilities. PRISM also supports rewards which are real values associated with states or transitions of the model. Where state rewards can be specified as:  $g : r$ , and transition rewards are represented as:  $[a]g : r$ .

The properties for a CTMC model can be specified in CSL logic that allows the specification of both transient behavior and steady state behavior. We use the P operator for specifying transient properties and S operator for specifying steady state properties. Another interesting operator employed is the R operator that is used to reason on the expected value of rewards.

#### Example

	Analytical models	Analytical Tools	Properties	Scope	Experimental setting
Ghosh et al. [12]	CTMCs	SPHERE	service availability, provisioning response delays	Infrastructure	IBM SmartCloud
Khazaei et al.[17]	CTMCs	Maplesoft	rejection probability, response delays	Infrastructure	Artifex engine
Li et al.[20]	queuing networks	Matlab	completion time, rejection probability, system overhead rate	Infrastructure	–
Chen et al. [6]	queuing networks	–	Rejection probability, task completion time	Infrastructure	XenServer and OpenStack
Kikuchi et al.[18]	CTMCs	PRISM	stacked operations, executed operations	Infrastructure	XenServer
Jhonson et al.[15]	MDPs	PRISM	Cost variation, deployment's resource usage	Infrastructure	–
Evangelidis et al.[9]	DTMCs	PRISM	CPU utilization, response time violation	Infrastructure, Platform	Amazon EC2 and Azure

Table 1: Main related work on cloud performance analysis.

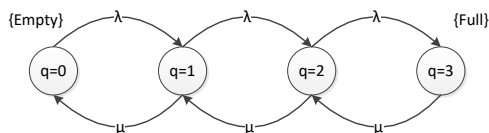


Figure 1: Queue model.

Let us consider the CTMC presented in Figure 1. It represents a queuing system with maximum length 3. The system can move from an empty state, where there is no job to a new state with ( $q = 1$ ) by the arriving of a new job with rate  $\lambda$ , and can return to the previous state by serving the job with a rate  $\mu$ . The same thing applies for the rest of states. The corresponding module for this model is described in Figure 2. We have to declare 3 variables, the integer variable  $lq$  that refers to queue length and two double variables representing the arrival and services rates. The main variable is  $q$ , which represents the possible states of the system through raising two main transitions, *Arrive* and *Service*, with their corresponding rates. We can express probabilistic properties based on the value of time variable  $T$ . For instance, we can express a CSL property that states that the probability of the queue being full with time  $T$  should not exceed the probability 0.5:  $P \leq 0.5[trueU \leq T \text{ "full"}]$ . The property

```

1  ctmc
2  const int lq = 3; //queue length
3  const double lambda = 1/10; //arrival rate
4  const double mu = 1/2; //service rate
5  module Queue
6  q: [0..lq] init 0;
7  [Arrive] (q<lq) -> lambda: (q'=q+1);
8  [Serve] (lq>0) -> mu : (q'=q-1);
9  endmodule
10 label "full" = q=3;

```

Figure 2: Prism model for the queuing system.

can be rewritten in a different way to estimate the probability of the property being true within time unit  $T$  as  $P = ?[trueU \leq T \text{ "full"}]$ .

## 4 Case study

The model that we are going to study concerns data centers that consist of a number of Physical Machines (PMs) [12]. When user requests arrive at a cloud center, a virtual machine or many VMs are deployed on PMs to serve this request. A single VM can be provisioned to serve a single request, however, in reality, multiple VMs can be provisioned on a single or multiple PMs to serve such complex request or super-task [17]. The PMs are grouped into three servers: hot (i.e., running VMs), warm (turned on, but without running VMs) and cold (turned off). It is tried first to provision the request on a hot pool if there is enough capacity, if there is no a hot PM available, there will be a

look-up for a warm PM, if all warm PMs are busy, a PM in the cold pool is used. In the case where no PM is available in all pools, the request will be simply rejected. The strategy of regrouping the PMs into multiple pools results in a good performance by reducing VMs provisioning delay and operational costs. The model proposed consists of three sub-models that refer to the three main steps of cloud servicing, which are resource provisioning decision, VM provisioning, and run-time execution. The overall solution is obtained by interacting over these three sub-models. The steps of provisioning and servicing are presented in Figure 3. In the following, we will describe each of the CTMCs models.

### 4.1 Resource provisioning decision model (RPDM)

This module is responsible for choosing the PM that can accept the request and in which pool. A finite decision queue is employed, where decisions are made on FIFO basis. The arrival to RPDM is modeled as Poisson process with arrival rate  $\lambda$ . The related CTMC model is shown in Figure 5.

The states in the model are presented as pairs  $(i, j)$ , where  $i$  denotes the number of requests being waiting in the global queue, and  $j$  denotes the pool on which the request is under provisioning. The initial state  $(0, 0)$  means that the system is in an empty state, where there is no request, neither in the queue nor under provisioning.  $j$  is set to 'h' if there is at least one hot PM that can accept the job for provisioning. Similarly, when  $j$  is set to 'w' (or 'c'), that means that a warm (or cold) PM can accept the job for provisioning. The waiting queue for this model has a maximum number  $N$ .

From the initial state, by arriving of a new request, the system moves to the state  $(0, h)$  with rate  $\lambda$ , since it tries to find a hot PM first. From this state the following three possible transitions can occur:

- A request is accepted for provisioning in a hot PM, and thus the module returns to the state  $(0, 0)$  with rate  $P_h\delta_h$ .
- Another request arrives, and the system moves to state  $(1, h)$  with rate  $\lambda$ .
- No hot PM can accept the request for provisioning due to insufficient capacity, and thus the system tries to find a warm PM and transits to state  $(0, w)$  with rate  $\delta_h(1 - P_h)$

Now, from the state  $(0, w)$ , the model tries to find an available warm PM to provision the request, if one warm PM is available, the model moves back to the initial state with rate  $P_w\delta_w$ , otherwise, the module tries to find a PM in cold pool by making a transition to  $(0, c)$  with rate  $\delta_w(1 - P_w)$ . Then, from the state  $(0, c)$ , the request can be either accepted in the cold pool, and thus the model moves back to the initial state with rate  $P_c\delta_c$ , or the request is rejected when there is no available cold PM, and thus the model

moves to the same state with a rate  $\delta_c(1 - P_c)$ . The state where  $i \geq 1$  means that  $i$  request is waiting in the queue.

The related prism module of this model is depicted in Figure 6. We use two main variables,  $i$  and  $j$ , where  $i$  refers to the number of jobs waiting in the queue, and  $j$  denotes the type of pool ( $j=1$  for hot,  $j=2$  for warm and  $j=3$  for cold). The commands with *wait* action and rate  $\lambda$  (lines 6, 11 and 16) refer to a new request waiting and staying at the same pool, hot, warm and cold respectively. The other commands of provision refer to the provision in hot, warm and cold respectively with the appropriate rates. The rest of the commands where no action is defined refer to searching for a PM in the next pool. While *wait* actions have no control on the entire model, and they are just used for indication, the other actions (Provision\_hot, Provision\_warm and Provision\_cold) are used for synchronization with the rest of provisioning models (hot, warm and cold).

To build our model we need global as well as local variables. While local variables are defined at each module, global variables are defined at the top of the global model, thus they can be used by all modules. The rates in CTMC models are usually defined as global variables. In addition, we can define some variables that play an important role in defining properties such as the time variable  $T$ . The set of variables with their values are presented in Figure 4. These values are basically adapted from [12, 10].

### 4.2 VM provisioning models

These models capture instantiation, configuration and provisioning of a VM on a PM. The model for provisioning a hot PM is described as a CTMC in Figure 10. In this model, requests, PMs and VMs are all assumed to be homogeneous, and each request is for one VM instance. We also assume that inter-arrival time, service time and VM provisioning time are all exponentially distributed.

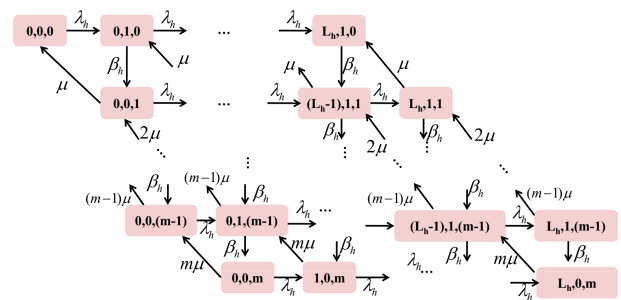


Figure 10: VM provisioning model for each hot PM[12, 11].

States of provisioning model are controlled by three main variables  $i, j$  and  $k$ , where  $i$  presents the number of requests in PM's queue,  $j$  presents the number of VMs currently being provisioned, and  $k$  presents the number of VMs which have already been deployed. There are also input parameters that control the model,  $L_h$  that represents the size of PM's queue,  $j$  can be 0 or 1 if the VMs are

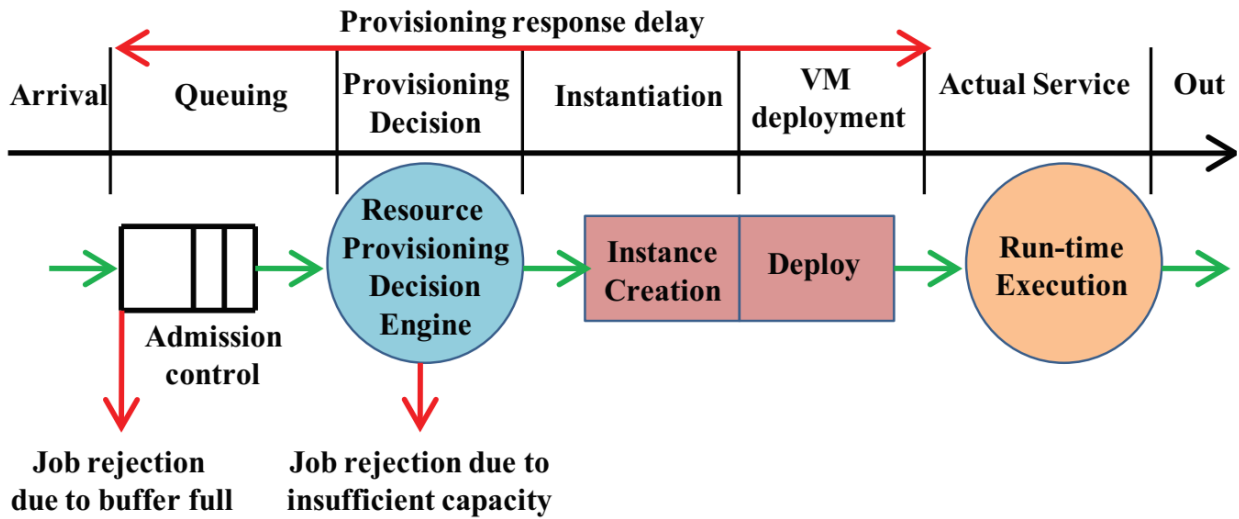


Figure 3: Request provisioning and servicing steps [12].

assumed to be provisioned one at a time, otherwise, a parameter can be introduced to refer to the number of VMs under provisioning. Finally, we have the maximum value of  $k$ , which refers to the maximum number of VMs that can run in parallel ( $m$ ). The other parameters concern rates: effective job arrival rate ( $\lambda_h$ ), VM provisioning rate ( $\beta_h$ ) and service rate ( $\mu$ ).

As we see in Figure 10, when a request arrives, the model moves from state  $(0,0,0)$  to state  $(0,0,1)$  with a rate  $\lambda_h$ , which means that the current request is under provisioning, then it moves to the state  $(0,0,1)$ , with a rate  $\beta_h$ , the last state indicates that one VM is deployed, upon service completion, VM instance is removed and the model goes back state  $(0, 0, 0)$  with service rate  $\mu$ .

The related PRISM mode for a hot PM is presented in Figure 7. We use here three variables:  $xh$  that refers to the state of PM's queue,  $yh$  that refers to the provisioning state and  $zh$  that refers to the number of VMs being deployed. The commands refer in order to provisioning, deployment and service respectively. An additional action has been added just to use it in properties that specify the number of requests being rejected.

The two other CTMCs for warm and cold PMs are similar, though, they can define different arrival and instantiation rates (see Figure 11 and Figure 12). The most important difference concerns provisioning step, where in both warm and cold pools, PMs are turned on but not ready to use, thus, they require additional startup time. Time to make a warm/cold PM ready for use is exponentially distributed with a rate  $\gamma_w/\gamma_c$ .

The related PRISM modules describing warm and cold provisioning models are shown in Figures 8 and 9 respectively. We notice that warm and cold models define the same variables and steps as the hot model does, except with the provisioning step, where the values  $yw$  and  $yc$  have a larger range,  $yw = 2/\gamma_c = 2$  refer to  $1^*$  and

$yw = 3/\gamma_c = 3$  refers to  $1^{**}$ . Thus, compared to the hot model, an additional section has to be added starting from line 11.

For the following values: ( $lq = 6, lh = 1, lw = 1, lc = 1$  and  $m = 2$ ), the model generated by PRISM consists of 72859 states and 289147 transitions. The size of the model may mainly vary to the number of variables used in the module, as well as the range of their values. For instance, if we let the value of  $j$  of warm and cold pools as the same as hot (i.e  $[0..1]$ ), we had to introduce two new Boolean variables to replace the values 2 and 3 of  $j$ . Such a solution could result in an additional large set of states.

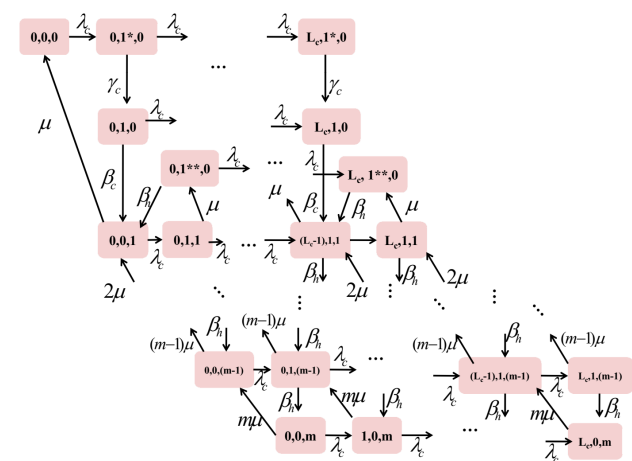


Figure 12: VM provisioning model for each cold PM[11].

### 4.3 Specification

In this section, we will show how we can specify quantitative properties in PRISM to reason about many measures of cloud performance through various operators employed by PRISM, which are the transient operator  $P$ , the steady

```

1 //arrival and service rates
2 const double lambda;
3 const double mu;
4 // provisioning rates
5 const double betaH=1;
6 const double betaW=1/2;
7 const double betaC=1/3;
8 //max VMs deployed, time and global queue size
9 const m=2 ;
10 const double T;
11 const lq =6;
12 // buffer size hot, warm,cold
13 const lh=1;
14 const lw=1;
15 const lc=1;
16
17 const double deltaH=3;
18 const double deltaW=3;
19 const double deltaC=3;
20 // prob. off succes in hot, warm and cold
21 const double ph=0.9 ;
22 const double pw =0.8 ;
23 const double pc=0.7 ;
24
25 const double lamH =deltaH* (1-ph);
26 const double lamW=deltaW * (1-pw);
27 const double lamC =deltaC* (1-pc);
28
29 const double muH=deltaH*ph;
30 const double muW=deltaW*pw;
31 const double muC=deltaC*pc;
32
33 const double lambdaH = lambda/2;
34 const double lambdaW =lambda/4;
35 const double lambdaC =lambda/5;
36
37 const double gammaW =1;
38 const double gammaC=1;
    
```

Figure 4: Global variables.

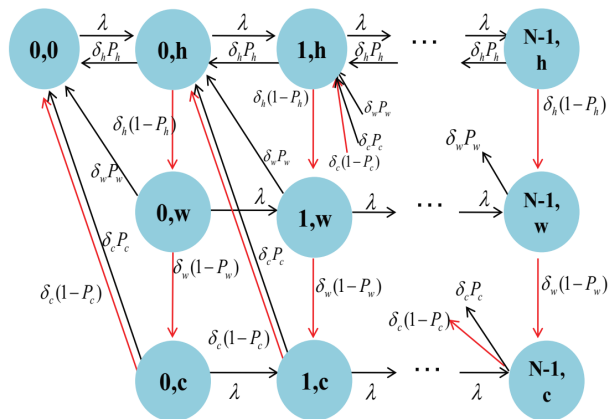


Figure 5: Resource provisioning decision model[12].

operator  $S$  and the reward operator  $R$ . The model checking algorithm used during the analysis phase is Jacobi method, though, we can use other methods such as Gauss-Seidel method. We can use many variations in the model parameters during the analysis, such as the number of PM's, the number of VMs, arrival and service rates, etc. Since each PM is represented by a complete module, to ease the analysis, we are going to fix the number of PMs, so no module duplication will be used.

We can use the simulation framework of PRISM to provide a detailed analysis based on these values. The different graphs to plot will be based on the variation of three main values, arrival rate  $\lambda$ , service rate  $\mu$  and time variable  $T$ . All the values are considered in minutes. We will show that the main measures, job rejection probability and waiting time can be easily computed using probabilistic and reward properties. We will also show how to obtain additional important measures. Before presenting these properties, we should introduce some labels that are employed to express in a better way these properties. The set of labels that we are going to use are presented in Figure 13, and the set of rewards are presented in Figure 14.

**Job rejection probability** As explained before, job rejection could be at the level of the global queue, where the buffer size reaches its limit, or on the level of the provisioning module, where there is no sufficient resources, which means that all PMs queues are full ( $x_h=lh \ \&x_w=lw \ \&x_c=lc$ ). To obtain the rejection probability due to insufficient capacity we use the following steady-state property :

$$S = ?[ "all\_Pools\_Full" ]$$

We fix the arrival rate by  $\lambda = 8$ , and we use different values of the mean service time  $\mu$  to obtain the results presented in Figure 15. From the Figure, we see that increasing the mean service time results in increasing job rejection probability. It's evident that taking more time to serve a request could result in rejecting new arriving requests. We can also use a steady property to estimate the long-run probability of the queue being more than 75% full, this interesting property results in 0.99 and can be expressed as the following:  $S = ?[ i/lq > 0.75 ]$ .

Another important measure can be estimated using steady-state operator is the steady probability that the system is in full provision state ( $y_h = 1 \ \&y_w >= 1 \ \&y_c >= 1$ ), which means that in all pools, there is a request being provisioned. This property is expressed as  $S = ?[ "all\_Provision" ]$  and returns a value of 0.94.

We can reason on minimum and maximum delay time taken for provisioned requests before being served. To do so we use the following reachability reward properties that estimate the reward accumulated along a path until a certain state is reached. The time reward is denoted by "time" in Figure 14 (line 9), where every transition is counted.

$$R "time" = ?[ F(j = 1 \ \&z_h = 1) \{ j = 1 \ \&z_h = 0 \} \{ max \} ]$$

$$R "time" = ?[ F(j = 2 \ \&z_w = 1) \{ j = 2 \ \&z_w = 0 \} \{ max \} ]$$

```

1 module rpdm
2 i: [0..lq];
3 j:[0..3] init 0;
4
5 [] (i=0) & (j=0)-> lambda : (i'=i) & (j'=1);
6 [wait] (i<lq-1) & (j=1)-> lambda : (i'=i+1) & (j'=j);
7 [] (i<lq) & (j=1)-> lamh : (i'=i) & (j'=2);
8 [Provision_hot] (i>0) & (i<lq) & (j=1) -> muH : (i'=i-1) & (j'=j);
9 [Provision_hot] (i=0) & (j=1) -> muH : (i'=0) & (j'=0);
10
11 [wait] (i<lq-1) & (j=2) -> lambda : (i'=i+1) & (j'=j);
12 [] (i<lq) & (j=2) -> lamw : (i'=i) & (j'=3);
13 [Provision_warm] (i>0) & (i<lq) & (j=2) -> muW : (i'=i-1) & (j'=j-1);
14 [Provision_warm] (i=0) & (j=2) -> muW : (i'=0) & (j'=0);
15
16 [wait] (i<lq-1) & (j=3) -> lambda : (i'=i+1) & (j'=j);
17 [] (i>0) & (i<lq) & (j=3)-> lamc : (i'=i-1) & (j'=j-2);
18 [Provision_cold] (i>0) & (i<lq) & (j=3)-> muC : (i'=i-1) & (j'=j-2);
19 [Provision_cold] (i=0) & (j=3) -> muC : (i'=0) & (j'=0);
20 endmodule

```

Figure 6: PRISM model for The RPDM module.

```

1 module vmpsm_hot
2 xh: [0..lh] init 0; // PM queue
3 yh: [0..1] init 0; // provision
4 zh: [0..m] init 0; // deployment
5
6 [Provision_hot] (xh=0) & (yh=0) & (j>=0) & (j<2)-> lambdaH : (yh'=1);
7 [Provision_hot] (xh<lh) & (yh=1) & (j=1)-> lambdaH : (xh'=xh+1);
8 [] (xh>0) & (yh<1) & (j=1)-> lambdaH : (yh'=yh+1) & (xh'=xh-1);
9 [] (yh>0) & (zh<m) & (j=1)-> betaH : (yh'=yh-1) & (zh'=zh+1);
10 [serve_hot] (zh>0) & (zh<=m) & (j=1)-> zh*mu : (zh'=zh-1);
11 [Reject_hot] (xh=lh) & (j=1) -> true;
12 endmodule

```

Figure 7: PRISM model for hot PM.

```

1 module vmpsm_warm
2 xw: [0..lw] init 0; // PM queue
3 yw: [0..3] init 0; // provision
4 zw: [0..m] init 0; // deployment
5
6 [Provision_warm] (xw<lw) & (yw=1) & (j=2)-> lambdaW : (xw'=xw+1);
7 [] (xw>0) & (yw<1) & (j=2)-> lambdaW : (yw'=yw+1) & (xw'=xw-1);
8 [] (yw>0) & (zw<m) & (j=2)-> betaW : (yw'=yw-1) & (zw'=zw+1);
9 [serve_warm] (zw>0) & (zw<=m) & (j=2)-> zw*mu : (zw'=zw-1);
10 // provision steps different than hot
11 [Provision_warm] (xw=0) & (yw=0) & (zw=0) & (j=2)-> lambdaW : (yw'=2);
12 [Provision_warm] (xw<lw) & (zw=0) & (yw=2) & (j=2) -> lambdaW : (xw'=xw+1) & (yw'=2);
13 [] (xw<lw) & (yw=2) & (zw=0) & (j=2) -> deltaW : (xw'=xw) & (yw'=1);
14 [] (zw=0) & (yw=3) & (j=2)-> betaH : (yw'=yw-1) & (zw'=zw+1);
15 [serve_warm] (zw=1) & (yw=1) & (j=2)-> zw*mu : (zw'=zw-1) & (yw'=3);
16
17 [Reject_warm] (xw=lw) & (j=2)-> true;
18 endmodule

```

Figure 8: PRISM model for warm PM.

```

1 module vmpsm_cold
2 xc: [0..lc] init 0; // PM queue
3 yc: [0..3] init 0; // provision
4 zc: [0..m] init 0; // deployment
5
6 [Provision_cold] (xc<lc) & (yc=1) & (j=3)-> lambdaC : (xc'=xc+1);
7 [] (xc>0) & (yc<1)&(j=3)-> lambdaC : (yc'=yc+1) & (xc'=xc-1);
8 [] (yc>0) & (zc<m) & (j=3)-> betaC : (yc'=yc-1) & (zc'=zc+1);
9 [serve_cold] (zc>0) & (zc<=m) & (j=3)-> zc*mu : (zc'=zc-1);
10 // provision steps different than hot
11 [Provision_cold] (xc=0) & (yc=0) & (zc=0) & (j=3)-> lambdaC : (yc'=2);
12 [Provision_cold] (xc<lc) & (zc=0) & (yc=2) & (j=3) -> lambdaC : (xc'=xc+1) & (yc'=2);
13 [] (xw<lw) & (yc=2) & (zw=0) & (j=3) -> deltaC : (xc'=xc) & (yc'=1);
14 [] (zc=0) & (yc=3) & (j=3)-> betaH : (yc'=yc-1) & (zc'=zc+1);
15 [serve_cold] (zc=1) & (yc=1) & (j=3)-> zc*mu : (zc'=zc-1) & (yc'=3);
16
17 [Reject_cold] (xc=lc) & (j=3)-> true;
18 endmodule
    
```

Figure 9: PRISM model for cold PM.

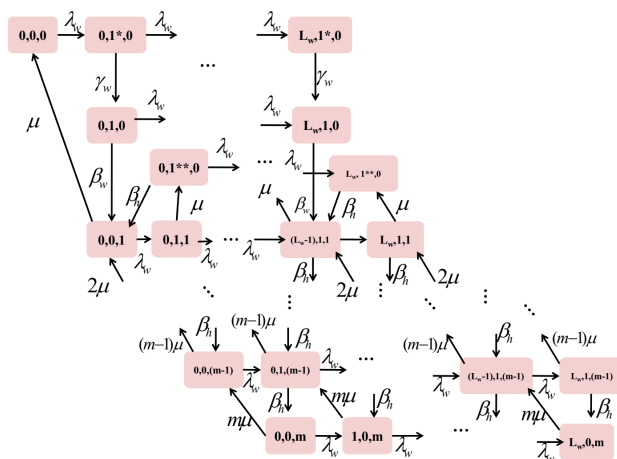


Figure 11: VM provisioning model for each warm PM[11].

```

rewards "queue_size"
true : i;
endrewards
rewards "Provision_queue_full"
[Provision_hot] (i=lq-1) : 1;
[Provision_warm] (i=lq-1) : 1;
[Provision_cold] (i=lq-1) : 1;
endrewards
rewards "time"
true : 1;
endrewards
rewards "Waiting_Pools"
true : xh + xw + xc;
endrewards
rewards "VMs_Deployed"
true : zh + zw + zc;
endrewards
rewards "VMs_Deployed_Hot"
true : zh;
endrewards
rewards "VMs_Deployed_Warm"
true : zw;
endrewards
rewards "VMs_Deployed_Cold"
true : zc;
endrewards
rewards "request_Reject_Warm"
[Reject_warm] true : 1;
    
```

Figure 14: Rewards.

```

label "deployed_Max_In_hot" = (zh = m & j=1);
label "deployed_Max_In_warm" = (zw = m & j=2);
label "deployed_Max_In_cold" = (zc = m & j=3);
label "all_Provision" = (yh=1 & yw>=1 & yc>=1);
label "all_Pools_Full" = (xh=lh & xw=lw
& xc=lc);
label "maximum_Deployment" = (zh=m & zw=m
& zc=m);
    
```

Figure 13: Labels.

$$R^{\text{time}} = ?[F(j = 3 \& zc = 1) \{j = 3 \& zc = 0\} \{max\}]$$

The properties estimate the reward that a state where a request is served can be reached starting from a state where the request is provisioned before being served. Roughly speaking, it computes the complete time between provisioning and service. For the hot pool, PRISM returns a

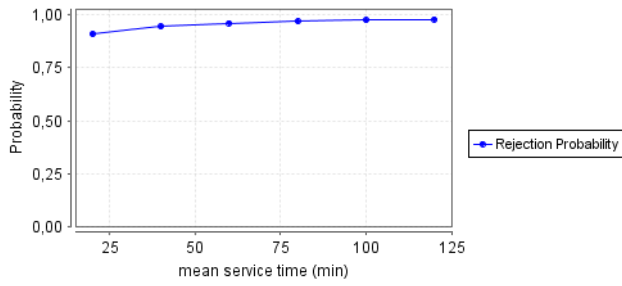


Figure 15: Job rejection probability.

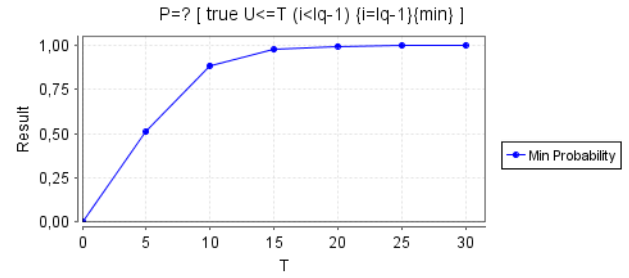


Figure 17: Min probability using filter of full queue.

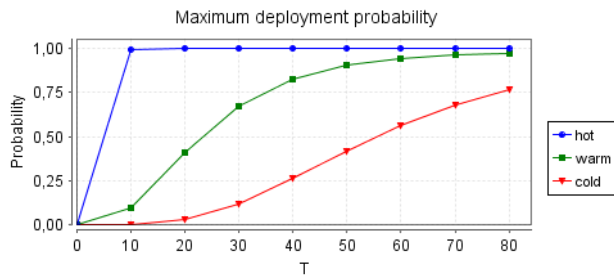


Figure 16: Probability of max deployment over time T.

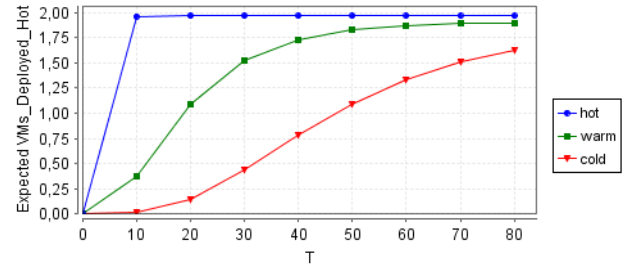


Figure 18: Number of VMs deployed.

result of 3 time units, 15 for the warm pool and 37 for the cold pool. It is evident that the time taken for the request to be served in hot pool is much less than warm and cold, since the last pools require additional provisioning time, and the resource provisioning decision model tries to provision the request in hot pool first. As we can compute maximum reward/probability, we can also compute minimum reward/probability using the feature of filter. For instance, we can compute the minimum probability of the global queue being not full, starting from the state where the queue has been already full. The property is presented as :

$$P = ?[trueU \leq T(i < lq - 1)\{i = lq - 1\}\{min\}]$$

The results of this property for different values of T are presented in Figure 17. A simple reachability property without a filter can be used to compute the probability of reaching the maximum deployment in each pool as follows :

$$P = ?[trueU \leq T"deployed\_Max\_In\_hot"]$$

$$P = ?[trueU \leq T"deployed\_Max\_In\_warm"]$$

$$P = ?[trueU \leq T"deployed\_Max\_In\_cold"]$$

The results of these properties are depicted in Figure 16. We see that the probability of reaching maximum deployment ( $zh = 2$ ) in hot increases faster than warm and cold. For warm pool, the maximum probability value is not reached until approximately  $T = 70$ .

We can also use *Instantaneous reward* properties to reason on the reward of a model at a particular instant of time. This type of properties associates with a path the reward in the state of that path when exactly T time units have elapsed. We can use it to estimate for instance the exact

number of requests waiting in the global queue in an instance T as follows:

$$R"queue\_size" = ?[I = T]$$

As time elapses, the reward will increase until it reaches its limit, which will be at most  $lq - 1$ . Similarly, we can use the property *R"Waiting\_Pools"*  $= ?[I = T]$  to compute the number of requests being waiting. We can use these *Instantaneous reward* properties also to reason about the number of VMs deployed globally at an instance T: *R"VMs\_Deployed"*  $= ?[I = T]$ . For instance, given a value of ( $T = 60$ ), the value returned is 5. For different values of T, we can use the following properties to estimate the number of VMs deployed at each pool.

$$R"VMs\_Deployed\_Hot" = ?[I = T]$$

$$R"VMs\_Deployed\_Warm" = ?[I = T]$$

$$R"VMs\_Deployed\_Cold" = ?[I = T]$$

The graph presented in Figure 18 shows the expected number of VMs being deployed for different values of T. It is evident that always the number of VMs in hot pool is greater, where  $zh$  reaches its limit rapidly before  $zw$  and  $zc$  respectively, because the RPDm tries always to find a hot PM first. The mean service time has a great impact on the results, by increasing its value, it could result in higher values of ( $zh$ ,  $zw$  and  $zc$ ), since each request takes much time to be served.

Unlike the *Instantaneous reward* properties, we can use *Steady-state reward* properties to compute reward in the long-run. To do so, the previous property of queue size can be written as follows: *R"queue\_size"*  $= ?[S]$  and it results in the value of  $lq - 1$ .

The last type of reward properties that can be used by PRISM, is the cumulative reward that associates a reward that is accumulated along the path until a bound T. For in-



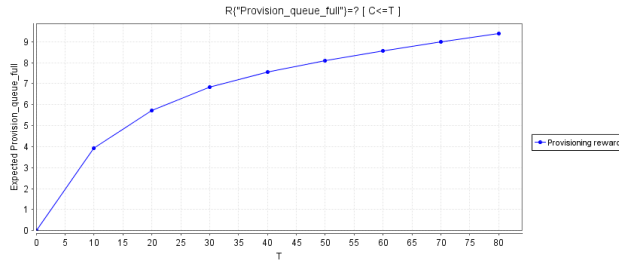


Figure 19: Expected number of requests provisioned after full queue.

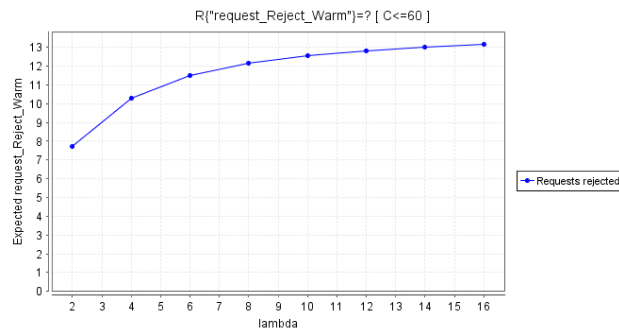


Figure 20: Expected number of requests rejected in warm.

stance, using the property  $R^{\text{Provision\_queue\_full}} = ?[C \leq T]$ , we will be able to compute the expected number of requests being provisioned from the state where the global queue is full. The results over time  $T$  are presented in Figure 19.

Now, we want to use cumulative reward properties to reason on the requests being rejected at the level of each pool with respect to the value of the arrival rate  $\lambda$ . From the previous results, we choose the warm pool for this property, because it knows a medium number of requests being provisioned and served compared to hot and cold. For a fixed period of time of 60 minutes, we try to estimate the number of requests being rejected for different values of  $\lambda$  using the following property:

$$R^{\text{request\_Reject\_Warm}} = ?[C \leq 60]$$

The results of this property as returned by PRISM are depicted in Figure 20. We notice that as the value of  $\lambda$  increases, the expected number of requests rejected in warm increases as well, because as much as requests arrive, more hot PMs start accepting requests and subsequently warm and cold PMs.

#### 4.4 Power consumption analysis

The importance of cumulative reward properties can be clearly shown in the context of power consumption. We use it here for estimating power consumption of the three pools. It is assumed that a hot PM consumes an idle power  $h_l$  when no VM is running, and the power consumption of a VM is assumed to be  $v_a$ . For the hot pool, a reward of  $r(i,j,k) = h_l + K v_a$  is assigned to each state of the hot pool,

where  $k$  represents the number of VMs being deployed. The rest of rewards rates for warm and cold pools can be found at [11], and their rewards as interpreted in PRISM are presented in Figure 21.

We notice that power consumption rates for both warm and cold have much details, since they require much additional startup time to be ready for use. This is represented in the variables  $y_w$  and  $y_c$  that have three possible values. It is assumed here that  $w_{l1} \leq w_{l2} \leq w_{l3} \leq h_l$ , and it is the same case for cold pool:  $c_{l1} \leq c_{l2} \leq c_{l3} \leq h_l$ . The values as adapted from [11] are declared as global variables in PRISM (see Figure 22). The values  $(w_{l1}, w_{l2}, w_{l3})$  are assumed to be within 20 - 50% of  $h_l$ , and the values  $(c_{l1}, c_{l2}, c_{l3})$  0-40% of  $h_l$ . Given these values, we can estimate the power consumption at each pool using the following cumulative reward properties :

$$R^{\text{Power\_hot\_PM}} = ?[C \leq T]$$

$$R^{\text{Power\_warm\_PM}} = ?[C \leq T]$$

$$R^{\text{Power\_cold\_PM}} = ?[C \leq T]$$

The results of these properties over time are presented in Figure 23. We see that power consumption in hot pool is much higher than warm and cold pools, due to the higher rates in hot. In addition, requests are provisioned more in hot, then warm and finally cold. These results can be also explained based on the previous graph (see Figure 18). We notice that after 10 time units, the power consumption in warm starts getting higher, due to a request being provisioned in warm. While warm power consumption could exceed 10% of hot power consumption by time  $T = 80$ , the cold power consumption stays in a low level.

## 5 Conclusion

In this paper we illustrated the use of probabilistic model checking as an effective framework for the evaluation and performance analysis of IaaS clouds. Using PRISM model checker, we implemented an analytical model that consists of many interactive sub-models. The model describes and quantifies the steps of provisioning and serving user requests on virtual machines (VMs), which are deployed on physical machines (PMs) regrouped in different pools. Using transient and steady properties, we were able to compute many important performance measures, such as rejection probability and time delay. In addition, using different types of reward properties, we were able to estimate many reward-based measures, especially the power performance trade-off of the IaaS cloud. The reliable estimations obtained can help cloud providers to get a better insight on cloud performance, thus avoiding SLA violation.

## References

- [1] AZIZ, A., SANWAL, K., SINGHAL, V., AND BRAYTON, R. Model-checking continuous-time markov chains. *ACM Transactions on Computational Logic* 01, 1 (2000), 162–170.

```

1     rewards "Power_hot_PM"
2     (j=1 & zh>0) : h_l + zh*v_a;
3     endrewards
4     rewards "Power_warm_PM"
5     (j=2 & xw=0 & yw=0 & zw=0) : w_l1;
6     (j=2 & (xw>=0 & xw<=lw) & yw=1 & zw=0) : w_l3;
7     (j=2 & (xw>=0 & xw<=lw) & yw=2 & zw=0) : w_l2;
8     (j=2 & (xw>=0 & xw<=lw) & yw=3 & zw=0) : h_l;
9     (j=2 & xw=0 & yw=0 & (zw>=1 & zw<=m) ) : h_l + zw*v_a;
10    (j=2 & (xw>=0 & xw<=lw) & yw=1 & (zw>=1 & zw<=m-1) ) : h_l + zw*v_a;
11    (j=2 & (xw>=0 & xw<=lw) & yw=0 & zw=m) : h_l + m*v_a;
12    endrewards
13    rewards "Power_cold_PM"
14    (j=3 & xc=0 & yc=0 & zw=0) : c_l1;
15    (j=3 & (xc>=0 & xc<=lc) & yc=1 & zc=0) : c_l3;
16    (j=3 & (xc>=0 & xc<=lc) & yc=2 & zc=0) : c_l2;
17    (j=3 & (xc>=0 & xc<=lc) & yc=3 & zc=0) : h_l;
18    (j=3 & xc=0 & yc=0 & (zc>=1 & zc<=m) ) : h_l + zc*v_a;
19    (j=3 & (xc>=0 & xc<=lc) & yc=1 & (zc>=1 & zc<=m-1) ) : h_l + zc*v_a;
20    (j=3 & (xc>=0 & xc<=lc) & yc=0 & zc=m) : h_l + m*v_a;
21    endrewards

```

Figure 21: Power consumption rewards definition.

```

1     // Power hot pm
2     //power consumption in a hot PM (h1)
3     const double h_l=270;
4     //power consumption per VM
5     const double v_a = 16;
6     // Power warm pm
7     const double w_l1=54;
8     const double w_l2=100;
9     const double w_l3=135;
10    // Power cold pm
11    const double c_l1=0;
12    const double c_l2=50;
13    const double c_l3=108;

```

Figure 22: Power consumption rates.

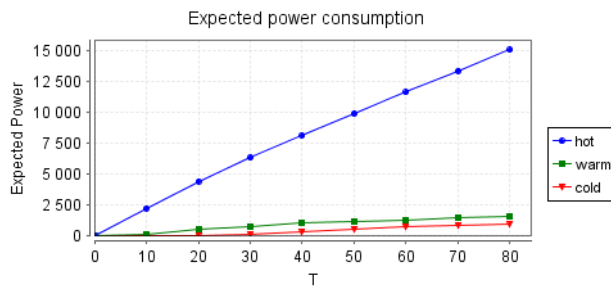


Figure 23: Power Consumption.

- [2] BAIER, C., HAVERKORT, B., HERMANN, H., AND KATOEN, J.-P. Model checking algorithms for continuous-time markov chains. *IEEE Transactions on Software Engineering* 29, 07 (2003), 524–541. <https://doi.org/10.1109/tse.2003.1205180>.
- [3] BARHAM, P., DRAGOVIC, B., FRASER, K., HAND, S., HARRIS, T., HO, A., NEUGEBAUER, R., PRATT, I., AND WARFIELD, A. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review* 37, 05 (2003), 164–177.
- [4] BRADLEY, J. T., CLOTH, L., HAYDEN, R. A., KLOUL, L., REINECKE, P., SIEGLE, M., THOMAS, N., AND WOLTER, K. *Resilience Assessment and Evaluation of Computing Systems*. Springer, 2012, ch. Scalable Stochastic Modelling for Resilience. <https://doi.org/10.1007/978-3-642-29032-9>.
- [5] BUYYA, R., YEO, C. S., VENUGOPAL, S., BROBERG, J., AND BRANDIC, I. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 25, 06 (2009), 599–616. <https://doi.org/10.1016/j.future.2008.12.001>.
- [6] CHEN, P., XIA, Y., PANG, S., AND LI, J. A probabilistic model for performance analysis of cloud infrastructures. concurrency and computation. *Practice and Experience* 27, 17 (2015), 4784–4796. <https://doi.org/10.1002/cpe.3462>.

- [7] DESHPANDE, P., SHARMA, S., AND PEDDOJU, S. Efficient multimedia data storage in cloud environment. *Informatica* 39, 04 (2014), 431–442.
- [8] DUAN, Q. Cloud service performance evaluation: status, challenges, and opportunities – a survey from the system modeling perspective. *Digital Communications and Networks* 03, 02 (2017), 101–111. <https://doi.org/10.1016/j.dcan.2016.12.002>.
- [9] EVANGELIDIS, A., D.PARKER, AND BAHSOON, R. Performance modelling and verification of cloud-based auto-scaling policies. *Future Generation Computer Systems* 87, 04 (2018), 629–638. <https://doi.org/10.1109/ccgrid.2017.39>.
- [10] GHOSH, R. *Scalable Stochastic Models for Cloud Services*. PhD thesis, Duke University, 2012.
- [11] GHOSH, R., NAIKY, V., AND TRIVEDI, K. Power-performance trade-offs in iaas cloud: A scalable analytic approach. In *IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)* (2011), pp. 152–157. <https://doi.org/10.1109/dsnw.2011.5958802>.
- [12] GHOSH, R., TRIVEDI, K., NAIK, V. K., AND KIM, D. End-to-end performability analysis for infrastructure-as-a-service cloud: An interacting stochastic models approach. In *2010 IEEE 16th Pacific Rim International Symposium on Dependable Computing* (2010), pp. pp. 125–132. <https://doi.org/10.1109/prdc.2010.30>.
- [13] HANSSON, H., AND JONSSON, B. logic for reasoning about time and reliability. *Formal aspects of Computing* 6, 5 (1994), 512–535.
- [14] HINTON, A., KWIATKOWSKA, M., NORMAN, G., AND PARKER, D. Prism: A tool for automatic verification of probabilistic systems. In *TACAS* (2006), LNCS, vol. 3920, Springer, Berlin, Heidelberg, pp. 441–444. [https://doi.org/10.1007/11691372\\_29](https://doi.org/10.1007/11691372_29).
- [15] JOHNSON, K., REED, S., AND CALINESCU, R. Specification and quantitative analysis of probabilistic cloud deployment patterns. In *HVC* (2012), LNCS, vol. 7261, Springer, Berlin, Heidelberg, pp. 145–159. [https://doi.org/10.1007/978-3-642-34188-5\\_14](https://doi.org/10.1007/978-3-642-34188-5_14).
- [16] KHAZAEI, H., MISIC, J., AND MISIC, B. Performance analysis of cloud computing centers using m/g/m/m+r queuing systems. *IEEE Transactions on Parallel and Distributed System* 23, 05 (2012), 936–943. <https://doi.org/10.1109/tpds.2011.199>.
- [17] KHAZAEI, H., MISIC, J., AND MISIC, V. A fine-grained performance model of cloud computing centers. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS* 24, 11 (2013), 2138–2147. <https://doi.org/10.1109/tpds.2012.280>.
- [18] KIKUCHI, S., AND MATSUMOTO, Y. Performance modeling of concurrent live migration operations in cloud computing systems using prism probabilistic model checker. In *IEEE 4th International Conference on Cloud Computing* (2011), pp. 49–56. <https://doi.org/10.1109/cloud.2011.48>.
- [19] LI, Z., ZHANG, H., O'BRIEN, L., CAI, R., AND FLIN, S. On evaluating commercial cloud services: A systematic review. *Journal of Systems and Software* 86, 09 (2013), 2371–2393. <https://doi.org/10.1016/j.jss.2013.04.021>.
- [20] LI, Z., ZHANG, H., O'BRIEN, L., CAI, R., AND FLIN, S. Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds. *IEEE Transactions on Automation Science and Engineering* 12, 01 (2015), 162–170. <https://doi.org/10.1109/tase.2013.2276477>.
- [21] MAHFOUD, Z., AND NOUALI-TABOUDJEMAT, N. Consistency in cloud-based database systems. *Informatica* 43, 03 (2019), 313–320. <https://doi.org/10.31449/inf.v43i1.2650>.
- [22] PONNURAMU, V., AND TAMILSELVAN, L. Secured storage for dynamic data in cloud. *Informatica* 40, 01 (2016), 53–62.
- [23] VAQUERO, L., RODERO-MERINO, L., CACERES, J., AND LINDNERS, M. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review* 39, 01 (2009), 50–55. <https://doi.org/10.1145/1496091.1496100>.
- [24] WIDED, A., AND OKBA, K. A novel agent based load balancing model for maximizing resource utilization in grid computing. *Informatica* 43, 03 (2019), 355–262. <https://doi.org/10.31449/inf.v43i3.2944>.



# Layered Architecture for Internet of Things-based Healthcare System: A Systematic Literature Review

Somayeh Nasiri

Department of Health Information Management, School of Health Management and Information Sciences, Iran University of Medical Sciences, Tehran, Iran  
E-mail: nasiri.so@iums.ac.ir, nasiri.him2015@gmail.com

Farahnaz Sadoughi (corresponding author)

Department of Health Information Management, School of Health Management and Information Sciences, Iran University of Medical Sciences, Tehran, Iran  
E-mail: sadoughi.f@iums.ac.ir

Afsaneh Dehnad

Department of English Language, School of Health Management and Information Sciences, Iran University of Medical Sciences, Tehran, Iran  
E-mail: dehnad.a@iums.ac.ir

Mohammad Hesam Tadayon

Iran Telecommunication Research Centre, Tehran, Iran  
E-mail: tadayon@itrc.ac.ir

Hossein Ahmadi

Centre for Health Technology, Faculty of Health, University of Plymouth, Plymouth PL4 8AA, United Kingdom  
E-mail: hossein.ahmadi@plymouth.ac.uk

## Overview paper

**Keywords:** internet of things, architecture, healthcare system

**Received:** June 20, 2021

*Internet of Things (IoT), known as a new paradigm, has shown to have a significant role in healthcare domains including remote vital sign monitoring systems, physical activity tracking, early disease diagnosis, and prevention of disease risks. Therefore, designing an integrated healthcare system based on Internet of Things is highly dependent on designing a layered architecture pattern. However, there are no comprehensive studies on Internet of Things layered architecture in the healthcare industry. The purpose of this study was to identify and scrutinize different types of layered architecture of Internet of Things in healthcare in terms of functions, and technologies. We evaluated studies proposing layered architecture of Internet of Things based on security aspects (security requirements and solutions). A systematic literature review was conducted by searching IEEE, PubMed, Scopus and Web of Science between 2005 and 2019. We were able to find 47 academic studies based on inclusion and exclusion criteria. We systematically reviewed applied functions and technologies and categorized them into three main layers namely, the perception, network, and application layers. This study also presented a comprehensive classification of sensor types. Only 28 out of 47 studies proposing Internet of Things architecture addressed security aspects among which privacy, authentication, and access control, confidentiality, and integrity had the highest rank. The layered architecture of Internet of Things is needed to provide an integrated framework for healthcare system, make better communication, and enhance the information management process. We suggest several potential solutions for future research directions according to technical, management, and security challenges*

*Povzetek: Podan je pregled literatura za zdravstvene sisteme, ki uporabljajo večnivojske arhitekture interneta stvari.*

## 1 Introduction

With the rapid advances in information and communication technologies in recent years, a new paradigm called Internet of Things (IoT) has emerged [1,

2]. IoT is an innovative technology which was first introduced by Kevin Ashton a professor of Massachusetts Institute of Technology (MIT) in 1999 [1, 3]. The term

"IoT" refers to connecting all physical objects and devices in the real world to the Internet [4, 5]. In fact, the main concept of emerging IoT is a network of uniquely identifiable and addressable objects (things) which are embedded with sensors, actuators and microprocessors [6,7]. These objects can communicate with each other and exchange information based on communication interoperability protocols [8-11]. The term "IoT" is often associated with different names such as "internet of objects", "ambient intelligence", "ubiquitous computing", "pervasive computing", "cyber physical systems" and "machine to machine interaction" [12-17]. According to these statements, IoT technology allows interaction between People to People (P2P), People to Machine (P2M), and Machine to Machine (M2M) [18]. Indeed, the idea behind the emergence of IoT was to emphasize on "the connectivity for anyone and anything at anytime and anyplace" [19]. One of the widely applied definitions is provided by Uckelmann et al. [11], describing IoT as "an integrated set of future internet" which can be described as a dynamic worldwide and ubiquitous network infrastructure of intelligent objects with "self-configuring capabilities". In general, the IoT motto is "having a modern and better life and promoting the life quality". This is possible by connecting a large number of intelligent devices, technologies, and applications [5]. Besides, three main elements including hardware (sensing devices), middleware (tools for storage and analyzing of data), and visualization are the key components which make IoT [20].

Gartner estimated that there would be 25 billion devices connected to the internet by 2020 [9]. These connections facilitate the volume of derived data and supply a wealth of intelligence for management, analysis, planning and decisions-making [18]. The IoT technology is currently used in different applications including smart home, grid, agriculture, transportation, logistics and industrial sectors [9, 21]. Among them, healthcare sector is considered as one of the most practical and attractive fields for IoT research [22]. Additionally, the IoT has made a significant potential for many medical domains including remote vital signs, patient monitoring systems, fitness programs, and daily physical activity tracking for the elderly and chronic diseases, continuously monitoring people's physiological and mental conditions [19, 23, 24]. In the near future, the way of providing healthcare services will be altered by developing IoT-based healthcare technologies and services like pervasive and ubiquitous healthcare and telemedicine [3].

Presently, traditional healthcare systems merely focus on patient treatment in healthcare facilities; thereby maintaining traditional healthcare systems are often costly and lacking in quality. Furthermore, patients have to be hospitalized during the treatment processes and delivery of healthcare services in hospitals. However, IoT technology will help change the direction of healthcare services delivery from the hospital-centered approach to the person-centered one. With this new approach, patient treatment takes place at home environment by smart phones and wearable technologies such as smart watches and bracelets [2, 3, 25, 26]. Patient-centered approach can

promote quality of care and educate patients about self-care management and enhance the doctor-patient relationship [2, 27]. This is possible with the help of IoT technology which provides remote patient monitoring by collecting real-time healthcare data and sending critical information to medical staff. It is worth mentioning that many patients need continuous medical monitoring. Additionally, clinicians should have timely access to their patients' medical records in which the information is as an essential and influential resource in improving the healthcare processes and critical decision-making [19, 28]. Hence, IoT supports early disease diagnosis, prevents possible risks, and assists doctors in remote patient monitoring [22, 29].

IoT-based healthcare system is one of the most challenging areas worldwide. The main problem arises from the requirements of IoT technology in which the environment is inherently complex and dynamic and consists of heterogeneous objects [22, 24,30]. Therefore, in order to communicate and share data between different systems, billions of heterogeneous devices should be able to interconnect and interact with each other through the Internet [3, 5]. Moreover, with the development of IoT system, novel healthcare services should be integrated into conventional healthcare systems [25, 28, 31]. It is necessary to design a layered architecture which will be able to overcome these new challenges of IoT [32, 33]. Designing the layered architecture pattern is known as an initial and crucial step for implementing IoT technology [34]. The architecture is a backbone for IoT, which helps developers and designers of systems to provide a cohesive principle for implementing this technology to deliver a quality product [5, 35]. The major goal of designing IoT layered architecture is to provide a common framework for integrating multiple technologies, making better communication, and enhancing information management processes such as sensing, data collecting, transmitting, processing and storage [36-38]. If IoT technologies in each layer are not suitably configured, the IoT system might be exposed to multiple vulnerabilities and problems [30]. Therefore, multi-layered architectural pattern supports different aspects of IoT system deployments such as scalability, modularity, flexibility, configuration, security, and interoperability among heterogeneous systems [5,37,38]. According to this pattern, system components are divided and organized into separated units, called layers [39-41]. This architecture pattern concentrates on grouping relevant functions into distinct layers which present a consistent set of roles and tasks [42].

Although there is evidence of the proposed IoT architectures in the healthcare industry, these architectures are sparse in multiple sources and are only suitable for special application domains, and there is no comprehensive review of IoT layered architecture covering all healthcare domains [6, 43]. Besides, some articles have not investigated IoT layered architecture and they only addressed general detail of IoT structure [44]. There are different architecture styles for IoT including layered, client-server, peer-to-peer, service-oriented, REST and Microkernel architecture. However, many

articles have overlooked the importance of IoT-based healthcare architecture in a layered way. Layered architecture assists system developers to partition the whole system into layers. Hence, it can provide an in-depth examination of the components and functions of each layer. A remarkable capabilities of a layered style is that it can be applied in integration with many other styles, which is not a statement that holds for all styles. Moreover, the advantage of layered architecture is scalability, reusability, and testability. If the layers are inspected in an in depth way, security will be ensured before connecting to healthcare devices and any changes in sensitive areas of the system can be detected [39].

Furthermore, there are no detailed studies demonstrating a well-defined classification of types of sensors, technologies, and functionalities for each layer, and no survey study has separately evaluated architectures in terms of security aspects. Therefore, in an effort to better understand the advancement of IoT technologies and functionalities, a systematic literature review was performed in this area. Thus, the main objective of this study was to present a comprehensive review of related studies regarding the IoT layered architecture in healthcare domains. Accordingly, we formulate four Research Questions (RQS) to be answered on the basis of a comprehensive review as follows:

- RQ1: What are the main application domains of layered architecture for IoT-based healthcare?
- RQ2: What technologies are used in each layer of architecture for IoT-based healthcare?
- RQ3: What main functions are considered in each layer of architecture for IoT-based healthcare?
- RQ4: What are the main aspects of security in the layered architecture for IoT-based healthcare?

The rest of this study is divided into five sections: the research methodology is illustrated in Section 2. The results are presented on the basis of the research aims in Section 3. Finally, discussion and conclusion are explained in Section 4 and 5, respectively.

## 2 Research methodology

This study is a systematic literature review based on a consistent guideline of Preferred Reporting Items for Systematic Reviews and Meta Analyses (PRISMA) provided by Moher et al. [45]. Therefore, we conducted a review of all relevant studies by focusing on well-defined research questions leading to an increase in knowledge and better understanding of the types of layered architecture for IoT-based healthcare.

### 2.1 Search strategy

We conducted a research of four main databases (IEEE, PubMed, Scopus and Web of Science) between 2005 and 2019 to meet the objective of the study. In this way, two major branches of science namely medicine and computer were connected. The above mentioned databases could support a rich source of reliable information regarding the layered architecture for IoT-based healthcare context from different areas of knowledge. Then, we used Boolean

operators such as “OR” and “AND” to combine the three groups of terms in searching the studies and constructing the search string as below:

("internet of things" OR "internet of objects" OR "ambient intelligence" OR "ubiquitous computing" OR "pervasive computing" OR "heterogeneous sensor" OR "cyber physical system" OR "machine to machine communication") AND (architecture OR framework) AND (e-health OR ehealth OR "healthcare" OR "health care" OR medica\* OR health\* OR "smart health")

In the next step, all searched studies were imported into EndNote software and duplicate studies were removed. The results are shown in Fig. 1.

### 2.2 Inclusion and exclusion criteria

On the basis of the research objectives, inclusion and exclusion criteria for our systematic literature review were determined (See Table 1).

I/E	Criteria	Explanation
Inclusion	Language type	Studies written in English-language
	Publication year	Studies published between 2005 and 2019
	Publication venue	Studies published in peer-reviewed journals and international conferences
	Research method	Case studies, experimental studies, surveys, review articles, field studies, simulation and prototyping studies
	Research scope	Studies proposing layered architecture pattern for IoT healthcare.
	Related content	Studies describing technologies and functions used in each layer of IoT.
Exclusion	Without full-text	The full-text of the studies is not available.
	Non-related publication source	Publication source of the studies is a report, brief report, book, thesis and dissertation, editorial letter, commentary, workshop, poster and unpublished working study.
	Vague categorization	Studies contain inadequate and unclear information about the topic.
	Unrelated contents	Studies reporting non-layered architecture patterns, describing technical and semantic issues, and focusing on IoT architectures other than healthcare field (such as manufactory, agriculture, transportation, building, logistics, etc.).

Table 1: Inclusion criteria and exclusion criteria for selection studies.

## 2.3 Study selection

The selection process of studies was carried out in two stages. First, the two authors independently reviewed and screened studies based on titles and abstracts, and according to the defined inclusion and exclusion criteria; consequently, the irrelevant studies were removed. In this regard, if a study referred to IoT architecture in healthcare area, it would be screened for the next stage. Second, in order to determine the articles for eligibility, full texts of extracted studies from the previous stage was investigated independently by the authors of this study. Next, according to our research aims, studies were selected if they followed the layered architectural pattern. As a result, a precise step was taken for study selection to reach a consensus. Eventually, we selected 47 studies regarding the layered architecture for IoT-based healthcare system (See Fig. 1).

## 2.4 Data extraction and synthesis

In this step of review, an initial data extraction form was developed to answer the research questions. Relevant items of IoT were extracted from each study in two sections including general information items (country, publication venues, and research type) and specific information items (architecture application domain, applied technologies and functions in each layer, security aspects and findings). The first two authors reviewed the selected studies independently. Any disagreement was resolved by consensus between the two authors and if necessary, the third and fourth author intervened.

## 3 Results

A total of 6706 studies were identified according to our search strategy, where 47 studies [2, 4, 7, 10, 23, 25, 28,

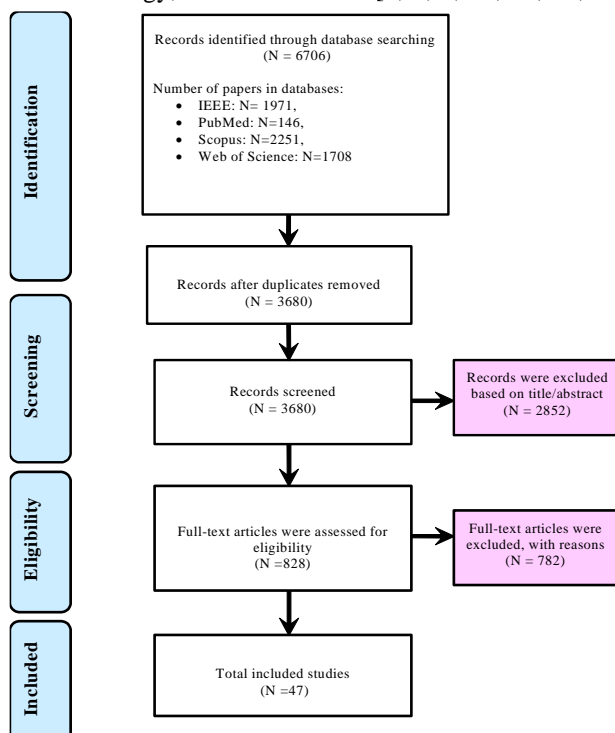


Figure 1: Flow diagram of review process.

31-35, 37, 41, 46-78] according to inclusion and exclusion criteria were included (See Fig. 1). In the following sections, after summarizing and reviewing all studies, we classified the selected studies based on the study characteristics including country, journal and conference names, and research type (Section 3.1). Then, on the basis of the analysis of studies, major findings consisting of application domains of IoT architecture, applied technologies and functions for each layer, and security aspects are presented in Section 3.2.

## 3.1 Overview of study characteristics

### 3.1.1 Distribution of studies by country

As can be seen in Fig. 2, 16 countries have published studies related to IoT layered architecture for various healthcare domains. It is clear from the chart that the majority of studies have been conducted in China (n=16; 34.04%) and India (n=7; 14.89%). Additionally, the detailed information about other countries is presented in Fig. 2.

### 3.1.2 Distribution of studies by the publication venues

Fig. 3 shows the distribution of 47 selected studies on the layered architecture of IoT-based healthcare, published in 29 journal articles and 18 international conferences.

The five hot venues with the larger number of published studies are shown in Fig. 4. "Applied Mechanics and Materials" (n=4; 8.51%) [47, 50, 54, 69] show the highest rank among the published studies, followed by "Future Generation Computer Systems" [2, 60], "Advanced Materials Research" [49, 75], "Journal of Medical Systems" [71, 72] and "E-Health Networking, Application and Services, HealthCom" [33, 74] (n=2; 4.26%).

### 3.1.3 Distribution of studies by research type

As Fig. 5 shows the distribution of 47 academic studies by research type, 28 of which were of the type of conceptual proposal providing the IoT layered architecture for various healthcare domains without performing any testing and evaluation. Only 11 out of 47 studies validated and experimented the performance of the layered architecture to pre-implementation phase (validation research), and 8 out of 47 studies evaluated the IoT layered architecture to post-implementation phase (evaluation research).

## 3.2 Overview of major study findings

### 3.2.1 Distribution of studies by application domains of the IoT layered architecture (RQ1)

Distribution of studies based on application domains of IoT layered architecture is shown in Table 2. Majority of studies which proposed IoT layered architecture, focused on specific domains of healthcare including diseases (n=7, 14.89%), ambient assisted living for elderly and disabled



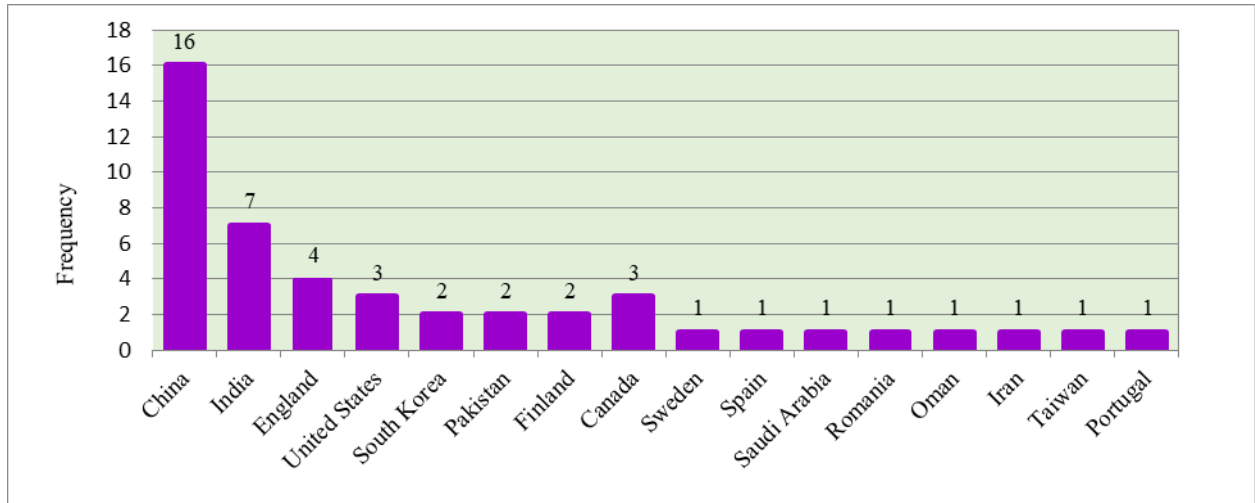


Figure 2: Frequency distribution of selected articles by country.

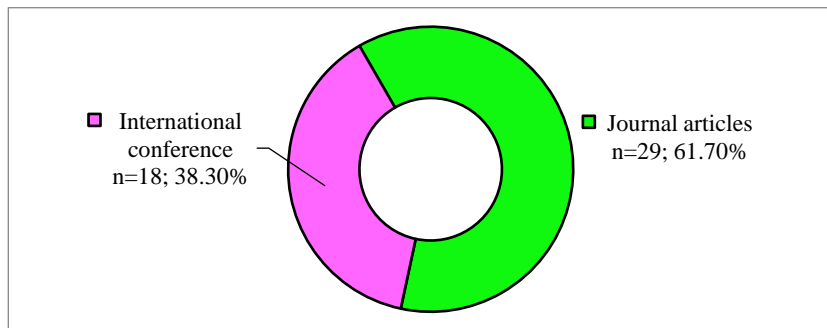


Figure 3: Frequency distribution of selected studies by venue types.

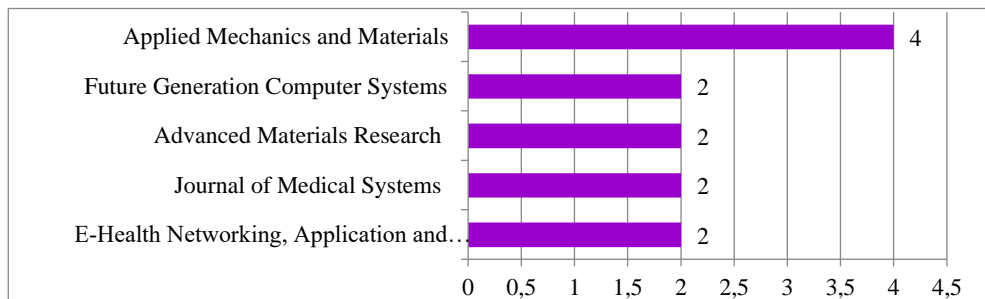


Figure 4: The hot five venues with the highest number of studies.

people (n=5, 10.64%), smart sports (n=2, 4.26%) and physical activity (n=2, 4.26%). Additionally, among the architectures presented, 17 studies (36.17%) specifically addressed service management in the healthcare sector including smart hospitals, medical equipment, ICU monitoring, nursing care, smart medication, mental health education, post discharge care management, medical emergency, personalized healthcare services and smart community healthcare services. Moreover, the results showed that 14 out of 47 studies (29.79%) proposing IoT layered architecture, belonged to general real-time or remote health monitoring, measuring various parameters including body physiology (e.g., temperature, BP, ECG, EEG, etc.).

### 3.2.2 The main technologies and functions in the layered architecture for IoT-based healthcare system (RQ2 and RQ3)

One of main contributions of this study is that it has systematically identified and categorized IoT-based healthcare technologies and functionalities into a layered architecture. Hence, to answer RQ2 and RQ3, we reviewed 47 studies regarding the IoT architecture in healthcare domains according to functionalities (See Fig. 6) and technologies (Table 3-5 and Fig. 7) and then categorized them into three main layers namely, the perception, network, and application layers. Due to lack of a unified architecture for IoT-based healthcare, and to better understand architectural layers, we followed the architecture proposed by International Telecommunication Union (ITU) [79, 80]. Detailed information on technologies and functionalities identified

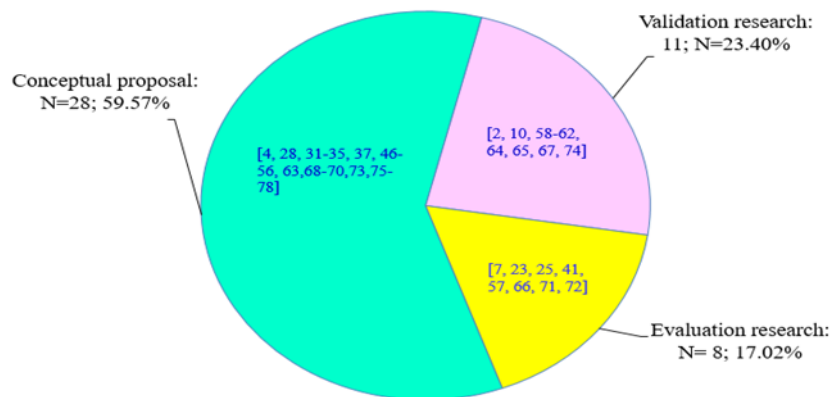


Figure 5: Frequency distribution of selected studies by research type.

in each layer of IoT architecture for healthcare industry are illustrated as follows:

**Perception layer**

The first layer of the IoT architecture is called the perception layer which is also known as "sensing" [74], "data collection" [59], "smart objects" [10], "hardware" [41], and "physical layer" [7]. As seen in Fig. 6, a large number of the studies describe the main functions of this layer including object recognition and identification, data sensing and acquisition. This layer consists of physical objects and various types of sensors, mobile devices (e.g., cell phones, tablet PCs, smart phones, PDAs, laptops and pocket PCs), and wearable devices (e.g., smart watches, wristband and fitness band). Depending on the functionality this layer provides, it can be divided into two sub-layers including perception nodes and perception networks [79]. Our findings show that most sensors act as tools for measuring and collecting the data related to the medical status, movement, location and position of a person and environmental conditions. Table 3 shows four categories of the most popular sensors used in IoT-based healthcare system. This category consists of physiological sensors, motion sensors, environmental sensors and position sensors. The following section reports the most important sensors used in the selected studies.

**A. Physiological Sensors**

We identified 12 types of *physiological* sensors used in IoT system as shown in Table 3. In general, *these* sensors are capable to sense and monitor physical parameters such as body temperature, Electrocardiogram (ECG), Electroencephalogram (EEG), Electromyogram (EMG), Blood Glucose (BG), Blood Pressure (BP), pulse rate and body weight. Some sensors measure emotional parameters and stress level of people. All of these parameters, which are known as vital indicators, continuously measure health condition by wearable devices [81]. According to Table 3, the most common types of *physiological sensors* belongs to ECG (27 studies), followed by body temperature (22 studies), BP (24 studies), and pulse rate (19 studies); whereas the *sensors* with the lowest type of frequency is Electrooculography (EOG) (1 study) and weight (3 studies).

**B. Motion Sensors**

In Table 3, different types of motion sensors, also called inertial sensors, are shown to have a considerable role in monitoring the motion and activities of human body [82]. Based on our analysis of selected studies, a wide range of

Application domains	Number of studies	Reference
Cardiovascular disease	5	[4, 41, 49, 57, 59]
Cerebrovascular disease	1	[67]
Diabetes management	1	[58]
Ambient Assisted living (AAL)	5	[10, 32, 46, 73, 74]
Smart hospital management	3	[33, 51, 61]
Medical equipment management	1	[35]
ICU monitoring	1	[71]
Medical emergency management	1	[72]
Nursing care management	3	[28, 47, 50]
Post discharge care management	1	[31]
Smart medication management	1	[25]
Physical activity monitoring	2	[55, 77]
Smart sports	2	[23, 65]
Mental health education	1	[54]
Personalized healthcare services	2	[48, 68]
Smart community healthcare services	3	[34, 52, 75]
General real-time/remote health monitoring	14	[2, 7, 37, 53, 56, 60, 62-64, 66, 69, 70, 76, 78]

Table 2: Application domains of the IoT-based healthcare layered architecture.

motion sensors including accelerometer (n= 15), gyroscope (n=8), magnetometer (n=6), barometric pressure (n=6), piezo vibration (n=1), strain gauge (n=1), impact (n=1) and tilt meter (n=2) were identified. Table 3 shows that accelerometer sensors have the highest rank, while gyroscope, magnetometer, and barometric pressure sensors are ranked second and third among motion sensors.

### C. Environmental Sensors

According to Table 3, most of environmental **sensors** are used for monitoring the environmental conditions and changes, such as room temperature, oxygen level and humidity. In addition, some sensors are able to detect and diagnose any smoke, chemical and toxic substances and light. Other environmental **sensors** are capable of controlling open and close doors or windows, and water leak. As can be observed from Table 3, **the lowest and highest** types of environmental sensors are related to water leak sensors (n=1) and room temperature (n=12), respectively.

### D. Position sensor

Positioning systems consist of various sensors which are capable of continuously detecting and tracing position, location, and proximity of a person or an object in real time [83]. According to Table 3, our findings show an extensive range of positioning technologies including GPS (n=11), ultrasonic devices (n=1), laser scanners (n=4), and image sensors (n=12) (e.g., camera, video camera, *webcam*, etc.).

### Network layer

The network layer, known as core layer, is placed in the second layer of IoT architecture. As can be seen in Fig. 6, this layer is responsible for communicating and transmitting the data securely from the lower layer (perception layer) to upper layer (application layer). Network layer contains three sub-layers including access network, core network, and local and wide area network [79]. Fig. 6 depicts the functions of network layer referring to interoperability, pre-processing, data buffering and aggregation. This layer can filter unnecessary data from a great volume of data and ensure a secure communication.

In this section, we present a summary of types of short-range and long-range communication network technologies shown in Table 4 and Table 5. These tables illustrate key properties of technologies according to the type of network, standard, frequency, data rate, range, topology, power consumption and cost [3, 23, 77, 83-89]. According to Alarifi et al. [83], a group of technologies act as network-based positioning system. Positioning technologies are also employed in network connectivity and communications including RFID, Bluetooth, Near-Field Communication (NFC), Wi-Fi, Infrared, *Ultra-wideband* (UWB), cellular, ZigBee, Z-wave, Low-power Wireless Personal Area Networks (*6LoWPAN*) (See Table 4).

### Application layer

The application layer is the topmost layer of the IoT architecture. According to Fig. 6, this layer can be divided

into two subsets namely, application support layer and IoT applications.

#### A. Application support layer

The Application support layer is responsible for storage, analysis and processing of received data from the lower layer (network layer). This layer is called with different names such as “cloud layer” [28], “processing and storage layer” [66], “service supporting” [67], and “management layer” [65]. As illustrated in Fig. 7, the selected studies show different technologies including cloud computing (n=21), data mining (n=13), and Decision Support System (DSS) (n=16). With the help of these technologies, significant and valuable information has been used for a specific purpose such as knowledge discovery, exploratory analysis, and intelligent decision-making (See Fig. 6). According to Fig. 7, only four studies reported fog computing and big data as effective technology for this layer. There were 20 studies discussing data centers as integrated tools for deploying and maintaining all stored data in various sources (e.g., web server, application server, analytic server, database server and storage systems). A data center as a software platform is an indispensable component of IoT architecture to handle and manage received data from gateway networks [10, 90]. Support layer enables management of the entire IoT system such as activities and services. In this respect, 12 studies reported the business process as one of the most important capabilities of the support layer. In IoT-based healthcare systems, the business process can be addressed for determining particular requirements and standards, and defining policies in how data flow is managed, processed, integrated and controlled [31, 74]. Additionally, this layer is the main body of coordination of all activities such as management of the patient medical records (for real-time accessing to history and necessary information), healthcare facilities, equipment and materials and financial issues.

#### B. IoT Application layer

IoT application layer is responsible for the delivery of diverse applications and services according to the user's request. According to Fig. 6, basic function of this layer is to display and visualize the information on the central monitoring systems such as nurse station, workstation, touch screen, BeneVision mobile viewer and dashboard. The outcome of this layer will be a visual representation of information in the format of texts, tables, pictures and graphs. Different entities are resided in the application layer as the user (e.g., patient, doctor, nurse, caregiver, administrator, patient's family, technical support team, etc.), location (e.g., hospital, emergency center, clinic, pharmacy supply chain, government agency, home health agency, insurance and other organizations) and technologies [51, 61, 62, 72]. Application layer involves a range of e-health technologies including Electronic Medical Record (EMR) [35, 69], smart e-health monitoring system [7, 53], Tele-consultation [47, 52-54], Tele-surgery [66], e-prescription [25], hospital information system [33, 50, 53, 61], and *medication reminder* system [56]. Clinicians and medical staff can remotely observe and monitor a patient's vital sign

Category	Sensor type	Function	Reference
Physiological sensors	ECG	Measuring heart's rhythm and electrical activity	[10, 23, 25, 28, 34, 41, 47, 49, 52, 55-57, 59-63, 65, 66, 68-71, 73, 74, 76, 77]
	BP	Measuring blood pressure (systolic and diastolic)	[7, 34, 37, 47, 49, 50, 52, 53, 58, 60-62, 64-66, 68-74, 76, 77]
	Body temperature	Measuring body temperature scale of person	[7, 34, 37, 47, 49, 51-53, 55, 56, 60-62, 64-66, 68, 70-74]
	Pulse rate	Measuring heart beats per minute	[7, 10, 23, 28, 34, 37, 47, 49, 50, 52, 53, 55, 56, 58, 63, 65, 72-74]
	SpO2	Monitoring and controlling blood oxygen (oxygen saturation) level in blood.	[23, 28, 34, 37, 49, 52, 53, 60, 61, 63-65, 71, 73, 76]
	Respiratory air flow	Measuring the activity and function of lung, respiratory rate and lung volume	[23, 34, 37, 56, 61, 65, 68, 71-74, 76, 77]
	BG	Measuring the glucose level in blood sample	[53, 58, 60, 62, 69, 72-74, 76]
	EEG	Measuring electrical signals of the brain	[25, 60, 62, 65, 68, 71, 73, 76]
	GSR	Measuring secretion of sweat gland, and explore emotional stress and anxiety level	[23, 34, 53, 60, 65, 71, 77]
	EMG	Measuring electrical activity of muscle during contractions or at rest	[34, 63, 65, 73, 76]
	Weight	Measuring body weight scale of a person	[50, 58, 73]
EOG	Measuring eye movements	[77]	
Motion Sensors	Accelerometer	Measuring linear acceleration of body motion, and monitoring during walking, standing and sitting	[10, 23, 28, 37, 53, 55, 56, 62, 63, 65, 68, 73, 74, 76, 77]
	Gyroscope	Measuring angular velocity and maintaining orientation	[23, 55, 56, 63, 65, 68, 76, 77]
	Magnetometer	Measuring the strength and direction of the magnetic field at a particular location such as detecting human movement direction during watching TV	[23, 52, 55, 65, 68, 77]
	Barometric pressure	Monitoring human behaviors during climbing up and down stairs and fall detection	[10, 23, 55, 63, 65, 77]
	Tilt meter	Monitoring vertical rotation, deflection, and deformation	[65, 73]
	Strain gauge	Monitoring the motion of the person's body such as vibration of the vocal cords, movements of joints	[65]
	Impact sensor	Detecting the position of the patient such as fall	[65]
	Piezo vibration	Measuring flexibility, vibration, impact and touch	[65]
Environmental Sensors	Room temperature	Measuring ambient air temperature (indoor/outdoor)	[10, 23, 50, 52, 53, 63, 68, 71, 74, 76-78]
	Hygrometer (humidity)	Measuring ambient moisture (indoor/outdoor)	[23, 50, 63, 71, 74, 77, 78]
	Smoke and toxic substance	Sensing and detecting the smoke, fire, toxic and chemical substance and monitors events or malfunctions lead to raise alarm conditions	[37, 46, 50, 71, 74, 76]
	Noise	Detecting the sound intensity in ambient environment	[23, 37, 52, 71]
	Open/close sensor	Detecting window or door open/close state	[65, 68, 74, 77]
	Light	Detecting the amount of light in the vicinity	[23, 50, 77]
	Oxygen level	Measuring the amount of Oxygen in the environment	[23, 71]
	Leak sensor	Detecting water leak	[68]
Position sensors	Image sensors	Tracking, identifying location, proximity of people or object in indoor or outdoor environment	[2, 7, 10, 32, 46, 47, 50, 51, 53, 54, 64, 67, 68]
	GPS		[10, 32, 46, 47, 49, 50, 56, 64, 68, 70, 77]
	Laser scanners		[32, 49, 53, 63]
	Ultrasonic		[46]
Other sensors and wearable devices	e.g. mobile devices, smart cards, wrist band, smart watch	[2, 4, 10, 31, 46-50, 55, 56, 60, 67, 68, 70, 75, 77, 78]	

Abbreviations: ECG= Electrocardiogram, EEG= Electroencephalogram, EMG= Electromyogram, EOG=Electrooculography, BG= Blood Glucose, BP= Blood Pressure, GSR= Galvanic Skin Response, SpO2= Pulse Oximeter Saturation, GPS= Global Positioning System

Table 3: The types of sensors used in IoT-based healthcare system by theirs functions.

Technology	Properties									Function			Reference
	Type of network	Standard	Frequency	Data Rate	Range	Topology	Power consumption	Cost	Identification	Positioning	Communication		
NFC		PAN	ISO/IEC 18092	13.56MHz (ISM)	up to 424 kbps	20cm	P2P	Low	Low	✓	✓	✓	[23, 61, 65, 68, 77]
RFID		PAN	ISO/IEC 15,693	125kHz–2.45 GHz	40–640 kbps	30–100 m	P2P	Low	Low	✓	✓	✓	[7, 23, 25, 32–35, 37, 46, 47, 49, 51, 53–55, 60, 65–68, 70, 71, 74, 77]
WSN		PAN	IEEE 802.15.4	902–928 MHz	20–250 Kb/s	20–100 m	Bus, Tree, Star, Ring, Mesh	High	High	✓	✓	✓	[34, 63, 67]
Bluetooth		PAN	IEEE 802.15.1	2.4 GHz	1Mb/s	<30 m	Star	Medium	Low	-	✓	✓	[2, 10, 23, 25, 31, 34, 37, 49, 50, 56, 58, 61]
BLE		PAN	IEEE 802.15.4	2.4GHz	1Mb/s	5–10m	Star	Very low	Low	-	✓	✓	[33, 41, 47, 57, 60, 73]
IrDA		PAN	IrDA	850–900nm	14.4 kbps	0–1m	P2P	Low	Low	-	✓	✓	[23, 49, 52, 65, 68, 76, 77]
UWB		PAN	IEEE 802.15.4a and ECMA-368	3.1G–10.6 GHz	100–500 Mb/s	<10 m	P2P	Low	High	-	✓	✓	[23, 51, 65]
ZigBee		PAN	IEEE 802.15.4	868/915 MHz, 2.4 GHz	20 k–250 kbps	10–100 m	Ad-hoc, P2P, star, or mesh	Very low	Medium	-	✓	✓	[31–33, 35, 46, 50–53, 55, 60, 61, 67–69, 72, 73, 77]
6LOWPAN		PAN	IEEE 802.15.4	868Mhz (EU) 915Mhz (USA) 2.4Ghz (Global)	40–250 Kb/s	10–20 m	Mesh, star	Very low	Low	-	✓	✓	[31, 32, 65, 67]
Z-Wave		PAN	Z-Wave alliance	900MHz	100 Kbps	30 m	Mesh	Very low	Very low	-	✓	✓	[74]
Wi-Fi		LAN	IEEE 802.11	2.4G–5 GHz	11–1730 Mbps	10–100 m	Star, mesh	Low	High	-	✓	✓	[7, 28, 32, 33, 35, 46, 47, 50, 51, 53, 55, 57–60, 62, 65, 67–70, 72, 77, 78]
Ethernet		LAN	IEEE 802.3	100 MHz	100 Mbps–10 Gbps	100m	Bus, star, P2P	Low	Low	-	✓	✓	[35, 47, 51, 60, 65, 72]

Table 4: Short-range communication technologies for IoT-based healthcare system.




Technology		Properties									Function			Reference
		Type of network	Standard	Frequency	Data Rate	Range	Topology	Power consumption	Cost	Identification	Positioning	Communication		
WiMAX		MAN	IEEE 802.16	2-66 GHz	1 Mb/s –1 Gb/s (fixed) 50–100 Mb/s (mobile)	<50 Km	mesh	Medium	High	-	✓	✓	[32, 46, 67, 68, 72, 78]	
Mobile Communication Network		WAN	2G-GSM, CDMA 2.5-GPRS 3G-UMTS CDMA 2000 4G-LTE 5G	450 MHz–2.6 GHz	1 Gbps	70km	Not available	High	Medium	-	✓	✓	[4, 7, 10, 31-35, 37, 41, 46, 47, 49-51, 53, 58-60, 62, 65, 67-69, 72, 74, 75, 77, 78]	
Satellite networks		WAN	IEEE 521	30-300 GHz	1 Mbps	6000 km	Star	Low	High	-	✓	✓	[53, 65, 75]	

Table 5: Long-range communication technologies for IoT-based healthcare system.

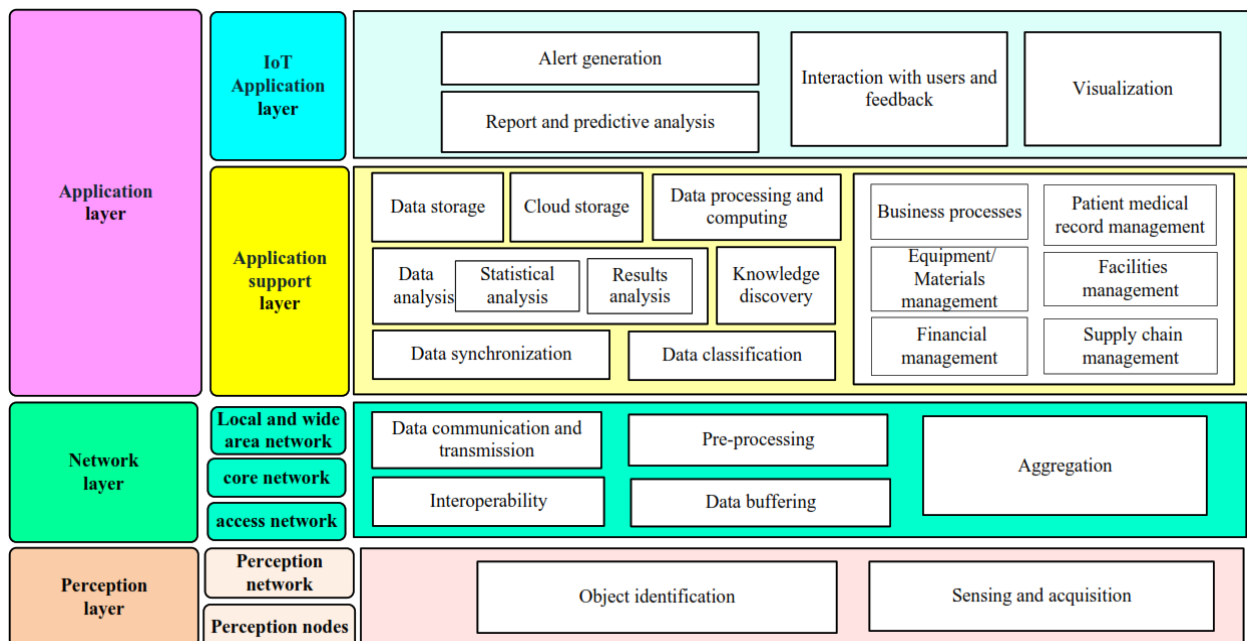


Fig. 6. The main functions used in the layered architecture for IoT-based healthcare system [2, 4, 7, 10, 23, 25, 28, 31-35, 37, 41, 46-78].

parameters, when the values of the parameters exceed the normal range; alert is automatically sent to medical center and feedback and advisory are provided to the users.

Reporting and prediction are other important functions of this layer described in selected studies.

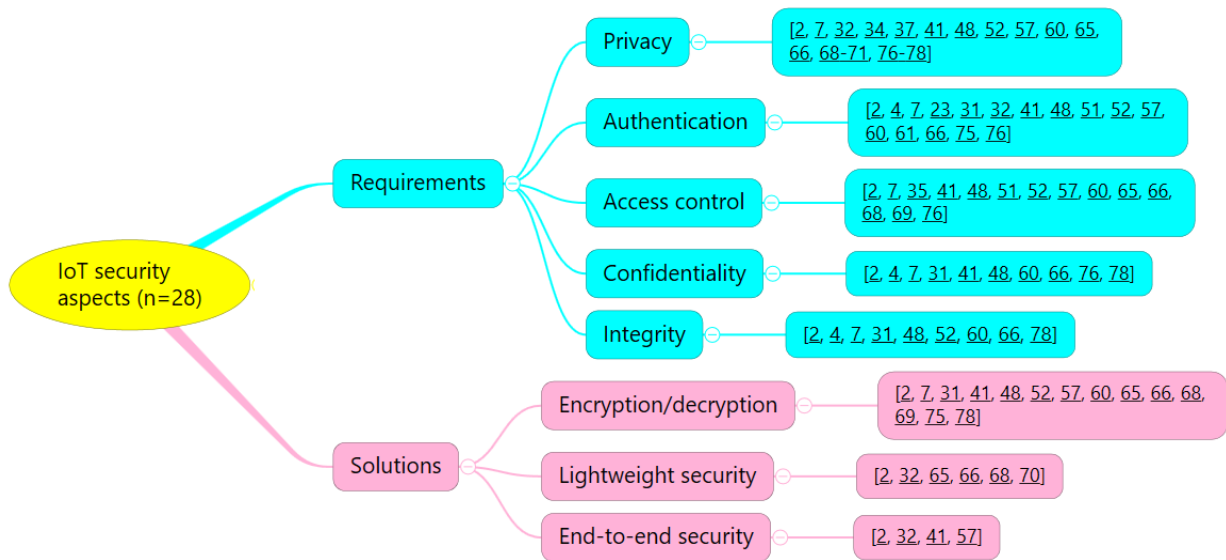


Figure 9: The selected studies about IoT-based healthcare architecture by security requirements and solutions [2, 4, 7, 23, 31, 32, 34, 35, 37, 41, 48, 51, 52, 57, 58, 60, 61, 65, 66, 68-71, 74-78].

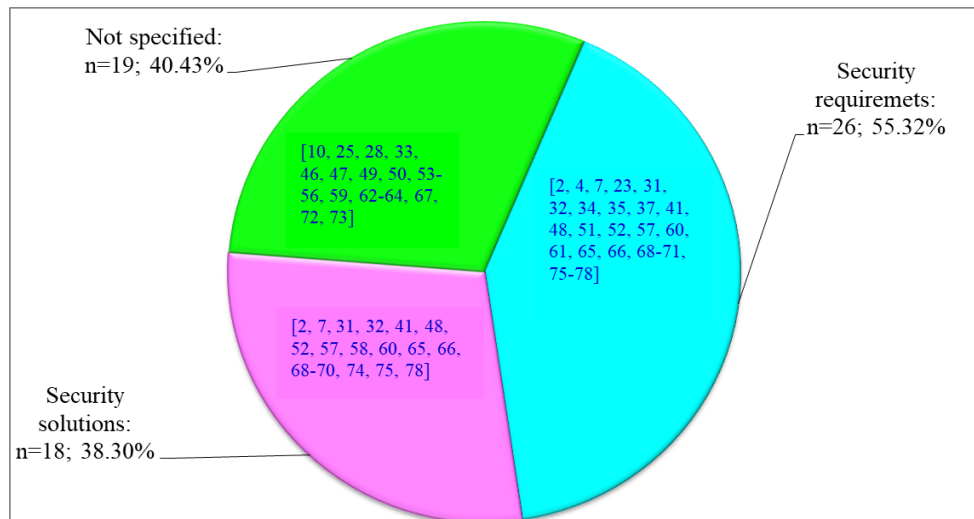


Figure 8: Frequency distribution of selected studies by main security aspects.

### 3.2.3 Distribution of studies by main security aspects in IoT architecture (RQ4)

As shown in Fig. 8, 28 out of 47 studies have included the security aspects of layered IoT architecture. Of these, 26 studies have reported security requirements and 18 studies have reported security solutions. However, 19 of these studies have not addressed any security aspects of IoT architecture. According to Fig. 9, 15 major security requirements in 26 selected studies were identified, among which privacy (n=19), authentication (n=16), access control (n=14), confidentiality (n=10), and integrity (n=9) have the highest rank among other security requirements. Besides, 16 out of 47 studies have focused on security requirements of IoT architecture, as well as providing security solutions including encryption/decryption, lightweight security and end-to-end security.

## 4 Discussion

In our systematic review, 47 academic studies were identified with respect to our formulated research questions between 2012 and 2019. According to our findings, selected studies were from three continents including Asia, Europe and America. It is noteworthy that Asian countries especially China and India have made the most contribution regarding the IoT-based healthcare layered architecture, suggesting an enormous potential and opportunities for research on this topic in Asian countries. Likewise, Talavera et al. [91] showed similar results about IoT layered architecture in agricultural domain. It seems that Asian countries particularly China has the most progress and activities with regard to the projects for IoT layered architecture in different fields.

Based on our analysis of research types, 23.40% of studies have investigated the validity of the proposed IoT architecture by experiments, simulations, or prototyping.

Besides, 17.02% studies evaluated the proposed IoT architecture in the post-implementation phase. Accordingly, one of the strengths of these studies is that they have practically evaluated the accuracy and performance of the architecture or the proposed framework during, before, and after implementation phases of the system. However, over half of the selected studies (28 studies) did not perform any validation or evaluation of the proposed architecture.

#### 4.1 Application domain (RQ1)

In terms of application domains of the layered IoT architecture, a large number of published studies ( $n=33$ ; 70.21%) have suggested specific application domains of healthcare including diseases, ambient assisted living for the elderly and disabled population, sport and physical activities and management of smart services. Similarly, an architecture has been proposed by Al-Taee et al. [58], mainly focusing on diabetes management. However, few studies have focused on general health monitoring, and measuring different parameters of patient status in real time (e.g., ECG, BG, temperature, BP, respiratory rate, etc.) [2, 7, 37, 53, 56, 60, 62–64, 66, 69, 70, 76, 78]. The results of this study are consistent with the systematic review conducted by Gonzalez et al. [92], remarking that most studies regarding the mHealth systems architecture, focus on diseases, elderly population, disabled people and the athletes.

#### 4.2 Technologies and functionalities (RQ2 and RQ3)

Majority of studies did not provide a comprehensive taxonomy of different sensor types. It is important to note that our review study has divided sensor types into four main categories and 31 sub-categories. Based on our analysis, we have identified four groups of fundamental sensors for IoT-based healthcare architecture such as physiological, environmental, motion, and position. Depending on sensor types and their functions, they allow us to perform continuous monitoring of medical parameters, measuring environmental conditions, detecting motion and behavior of people especially the elderly and tracking location, position and proximity of humans or objects [23, 50, 52, 63, 65, 68, 71, 93, 94].

More importantly, based on our findings, BLE technology has a considerable merit over Bluetooth in terms of low energy consumption for wireless networks. BLE is one of the wireless PAN technologies, known as Bluetooth smart which needs a small battery to be able to run the device for a long time [6, 20].

Network layer contains a various types of gateways which are essential to data transmission and are known as intermediate tools for the connectivity between sensors and cloud [18,28,35,93]. According to our results, this layer encompasses well-known short-range and long-range communication technologies which act as a gateway network [65]. The most substantial function of network layer is interoperability which facilitates an effective communication and information exchange across

heterogeneous devices [6, 38]. Based on our analysis of selected studies, it is surprising that most of the studies have proposed IoT architectures without considering the interoperability among devices. Only 25.53% of the selected studies ( $n=12$ ) [2, 25, 31, 32, 35, 37, 60, 61, 65, 68, 77, 78] have highlighted the interoperability issue for IoT architecture. Due to an increase in the number of connected objects, interoperability has faced many complexities and barriers such as devices compatibility and identification issues affecting different levels of healthcare systems [6,18, 38]. In this regard, the need for well-defined communication protocols is a critical and basic component of IoT architecture, allowing devices to interact with each other. It is important to develop universally accepted standards to overcome interoperability problems. In this regard, allocation of a unique Internet Protocol (IP) address such as IPv4 and IPv6 for each device should be taken into consideration. However, IPv4 address alone cannot respond to identify, and protect objects because of scalability and mobility of IoT-based healthcare system with the increasing number of internet connected devices and applications. Consequently, mobile IP for IPv6 address can guarantee billions of large scalable connected devices. This process was facilitated through integrating IPv6 infrastructure and 6LowPAN protocol [22, 32, 38]. On the other hand, Domingo [46] has remarked that the above-mentioned protocols are not suitable in terms of energy efficiency, cost and computation. Therefore, it is necessary to conduct further research on adapting existing protocols and discovering innovative solutions.

Based on our analysis of the selected studies, the third layer has the greatest functionalities and tasks compared with other layers. Overall, the most important task in this layer is processing, computing, storing, analyzing, management and business [4,31,65,66,69,95]. One of the most central roles in this layer is the business process. Consequently, it is important that healthcare organizations pay more attention to determining policies and requirements regarding the data management, recourse and service management, equipment management and facilities management [18, 22, 31, 32, 37, 61, 65, 67, 93, 96]. However, many reviewed studies ignored the importance of business activities and provided no solution for business technology algorithms. It is worth mentioning that some proposed architectures for IoT in areas other than healthcare domain have considered the business layer as a separate layer [5, 93]. However, findings from the two studies of Patel et al. [18] and Darwish et al. [97] are in line with our findings which suggest that business module is an integrated part of service management layer. Therefore, the role of business support systems is essential in the third layer.

In application support layer, cloud computing and big data analytics are known as two novel and ideal technologies [98, 99]. In our review, only 8.51% of selected studies used these two technologies to overcome the challenges of big data. Given that IoT ecosystem faces massive volume of data generated by medical sensors and numerous devices in real-time, the process and analysis of all the data is necessary [100, 101]. In healthcare industry,



cloud computing is a perfect technique to support recourse scalability, flexibility, data storage and computing, and cost saving [100]. Consequently, it is worthwhile to create a smart environment in healthcare industry by emerging cloud computing and IoT. Jalali et al. [34] have remarked that cloud is an imperative and integrated part of an IoT-based healthcare system. Existing cloud computing in IoT architecture facilitates ubiquitous access to resources and services in demand over the network and meets the needs of different medical centers [22]. Guo et al. [102] have suggested that integration of cloud computing and IoT technologies have created a new idea of “cloud of things”. It can be inferred that in the near future, cloud computing will have a strong impact on the development of IoT-based information systems. Additionally, big data analytic tools contain different methodologies that can solve data processing and analytics problems in IoT environment and are part of the IoT system requirements [22, 90]. However, it is surprising that big data technology is used in a small number of proposed architectures.

In application support layer, depending on patient situations and required functions, data processing is done in two forms. Some of the events periodically collect sensory data and perform batch processing at a specific time. In the case of emergency units, it is necessary to immediately collect data via sensor devices, to have real-time processing and giving a fast response to the patient’s critical situation [18, 28, 71, 72, 103]. For this reason, IoT medical device manufacturers and developers should take into account both batch and real-time processing to meet users requirements.

Our findings showed that only few studies noted fog computing as the main technology in this layer. Almehmadi et al. [76] discussed that fog computing is one of main IoT technologies expanding cloud computing to new services at the edge of the network. Fog computing supports a variety of services such as latency reduction, real-time systems, mobility, heterogeneity, and interoperability accompanied by the cloud computing. IoT-based healthcare systems should be capable of efficiently functioning to provide continuous vital sign monitoring and real-time medical services without any delay and interruption [2, 76]. Based on our analysis, we believe that it is eminent to consider emerging fog computing in IoT architecture. Azimi et al. [57] evaluated a fog-assisted computing architecture for healthcare IoT systems in terms of response time and latency. These indicators are a critical measure to generate alerts and notifications regarding the status of patients in cases of emergency.

Based on our review, few related studies applied middleware technologies in IoT architecture [33, 50, 54]. Nonetheless, middleware technology, which has an important role in supporting a system, allows overcoming the problems related to IoT-based healthcare systems such as heterogeneity, dependability, interoperability, and decision-making [6, 33, 90].

The result of our review demonstrated that 27.66% of the studies have emphasized on data mining playing a crucial role in extracting useful information and knowledge discovery in IoT architecture [23, 31, 33, 34,

63, 65, 66, 68, 71, 99]. Nowadays, data mining is used mainly for predicting a range of diseases, assisting with diagnosis and advising physicians in making clinical decisions. But, the potential of data mining is even greater; it concentrates on anomaly-based discoveries to create more informed decisions, and predictive modeling. Overall, application layer is the most important and practical layer which acts as a user interface in terms of providing personalized services to meet the needs of different users including doctors, medical professionals, and patients [9, 28, 97, 98, 104]. In this layer, the required modules for controlling, monitoring, and producing the alert of the IoT-based healthcare systems should be considered [9]. Because the application layer is in direct association with system users, the authors of this review believe that designing a secure platform for the workstations, smart phones and embedded PCs is essential to the safe protection of the device and the trusted visualization and presentation of data types (texts, images, sounds, reports, etc.).

### 4.3 Security aspects (RQ3)

Considering that IoT-based healthcare devices and applications deal with vital and personal information of patients, they must be protected against any security threats and attacks. Due to the mobility nature of IoT devices connecting objects to global information networks for their access at anytime and anyplace, a wide range of security challenges arise in the healthcare industry. [22]. As a result, IoT security is one of the serious issues in such uncertain and unpredictable environments and can affect the adoption of IoT system. The findings of this review shows that almost half of the studies [10, 25, 28, 33, 46, 47, 49, 50, 53-56, 59, 62-64, 67, 72, 73] proposing IoT architecture have not provided any comprehensive security solutions or requirements while patient information is sensitive and can be susceptible to hackers during transfer or synchronization stages [105]. Hossain et al. [105] have argued that wearable devices, which are based on IoT, and continuously collect data from patients’ ECG, are subject to security breaches. In order to ensure a safe and high quality IoT-based healthcare service, data security and privacy of patients must be protected against any illegal access.

A few of the selected studies have considered IoT security aspects; however, these aspects are only limited to some layers of the IoT architecture [4, 7, 32, 58, 61, 67, 68, 71]. In this regard, Bilal et al. [106] have remarked that all layers of IoT architecture face security challenges and threats. As a result, it is essential that security requirements be considered in all layers of IoT system. For instance, application layer of IoT architecture faces challenges like user privacy and access controls during data sharing, phishing, malware and injection attacks while network layer faces major security problems such as integrity and data confidentiality [106]. According to the research performed by Vijayalakshmi et al. [107] the main threats in network layer are eavesdropping, and Denial of Services (DoS)/Distributed Denial of Service (DDoS).

These threats breach integrity, confidentiality, and availability.

Bilal et al. [106] have suggested that the existing conventional security requirements and standards for IoT architecture is not adequate to protect vulnerable intelligent devices. This finding indicated that a dynamic defense-based mechanism is required for IoT medical devices. Based on our results, it is surprising that some security requirements such as autonomy, safety and fault tolerance are very negligible, whereas according to National Institute of Standards and Technology (NIST) guidelines [108], these requirements are necessary for establishing a resilient system against cyber-attacks. Due to resource constraints of IoT sensors and devices, it is worth mentioning that encryption mechanisms proposed should be lightweight to meet the security requirements [32, 66, 109]. Zhou et al. [110] suggested a secure IoT scheme based on elliptic curves cryptosystem. This scheme is more appropriate and economical for limited resources (e.g. computing complexity, bandwidth, cost, etc.) in IoT environment. It is surprising that only six studies have used lightweight encryption solution for IoT architecture. However, the authors of this study believe an additional work regarding the IoT-based healthcare security should be conducted to further investigate security requirements, vulnerabilities, and mechanisms. IoT in Healthcare industry is a complex area with different users, such as patients, doctors, researchers, insurance payers, and pharmaceutical companies, dealing with large amounts of patient data. Bublitz [78] et al. have offered Blockchain as a distributed and trusted mechanism which can improve issues in the healthcare industry such as security, transparency, data fragmentation, data exchanging, and interoperability. Blockchain can be combined with IoT as an extraordinary solution to enhance patient log control and facilitate secure access to the electronic records. Considering the importance of healthcare information for continuously monitoring patients and the decisions making based on it, we should not ignore the security and quality of the services and information. In this regard, Ebrahimi et al. [111] suggested decentralized trust management model as an efficient mechanism for protecting for IoT based healthcare devices. Alam [112] described IoT with blockchain enables ensure secure data storage and transactions in several industries like e-banking, healthcare monitoring, robotics through a peer-to-peer communication utilizing a shared database without third-party agents.

#### 4.4 Comparison of related studies

We reviewed IoT layered architectures from different aspects. In comparison to previous literature on IoT, our study is a strong review in terms of selected keywords, number of databases, various domains of healthcare and extensive research.

Previous studies have only focused on one specific domains of IoT architecture in healthcare industry. For instance, some studies have proposed architectures for nursing care [28], personalized healthcare system [68], disabled people [32], smart sport [65], and physical

activities [77] while our study is not exclusive to a specific domain of the disease, population, and service. The present study is a comprehensive review of all IoT architectures in healthcare domains. On the other hand, some studies have not precisely investigated functions and technologies of each layer. Besides, some reviews have used a limited list of terms for search strategies [3, 28, 68]. For example, in a survey study of advanced internet of things for personalized healthcare system, Qi et al. [68] searched only three databases including IEEE Xplore, ACM digital library and ScienceDirect, whereas our review has systematically covered an extensive and complete search of terms related to the IoT and has addressed more databases.

Lopes et al. [32], Meng et al. [69], Mohammed et al. [41] have concentrated on few technologies of IoT architecture, and the review study by Ahmadi et al. [3] have not provided a robust classification of sensors types and functions. This review study has identified and synthesized all the relevant publications on applied technologies in each layer of IoT architecture and presented a comprehensive taxonomy of useful sensors in healthcare industry. More importantly, the present study has scrutinized and evaluated the security aspects in each of the studies proposing layered architecture.

#### 4.5 Limitations

In our review study, there are several limitations which can be considered as potential suggestions for future research. First, the data were extracted from the academic journals and international conferences, in which some documents were not closely examined. This includes reports, brief reports, books, theses and dissertations, commentaries, and unpublished studies. Second, the selected studies belong to four main databases (IEEE, PubMed, Scopus, and Web of Science), and other databases (such as ACM, Springer, Willey, etc.) were not searched. Third, we did not perform manual search in the reference lists of the selected studies. Hence, it is possible that some potential information may have been missed. For this reason, further review studies can be conducted to cover other documents. Forth, in our review, studies written in non-English languages were excluded. It is possible that other relevant studies written in different languages might have been missed.

### 5 Conclusion and future work

This study systematically reviewed the current knowledge about different layered architecture of IoT on the basis of the various application domains of healthcare. We were able to identify and summarize types of functions and technologies in each layer and security. The layered architecture perspective of IoT in the healthcare industry covers three main layers: perception layer, network layer, and application layer. In sum, the results of this review are expected to be useful and effective for a large group of communities in the area of the IoT-based healthcare system including academic groups (researchers), healthcare groups (nurses, doctors and medical team

staff), engineering groups (IT engineers and software developers), and governmental groups (policymakers, managers and decision makers).

As discussed earlier, layered architecture is known as an important and essential step for designing and implementing IoT integrated healthcare system. In fact, multi-layered architectural pattern provides a framework with the aim of integrating various technologies and maintaining interaction of system components with each other to support platforms and application services. Moreover, this architecture facilitates communication and electronic information sharing, and improves remotely healthcare monitoring.

Our research is a landscape for developing more research, providing quality services and real-time access to patient information, designing a suitable platform, and making strategic decisions and investment for IoT deployment. However, we are aware that achieving these goals is not easily reachable, because there are still many challenges for IoT architecture and system deployment. Thus, challenges related to IoT provide directions for further research. The findings of this study provide several recommendations that create opportunities and motivate researchers in future research as follows:

- According to our findings of this review study, there is not a solitary IoT architecture which addresses all issues, including reliability, latency, scalability, and security in the healthcare industry. Therefore, the authors suggest that technical and semantic issues related to IoT-based healthcare system architecture such as scalability, interoperability, standards and communication protocols be considered. Scalable architecture ensures the proper functioning of IoT networks and supports the connectivity of increasing number of devices and sensor nodes. In the healthcare sector, various applications, devices and protocols are delivered by different vendors. Hence, interoperable architecture approaches are recognized as solutions to overcome the heterogeneous networks allowing data sharing throughout IoT system. More importantly, security and privacy issues are the significant problems in IoT architecture. The security issue is related to all layers of IoT architecture which are exposed to different types of attacks and threats, which are areas for investigating in healthcare industry.
- The IoT architecture deals with diversity of high and low cost technologies which also have constraints in terms of power consumption and data storage. Thus, designing a model for cost analysis of IoT devices and choosing an economical solution in terms of energy management is essential.
- The findings of this study indicate that business processes, data flow management and policy determination are main functions which should be considered for IoT-based healthcare architecture. As a result, designing a comprehensive business model for IoT-based healthcare can be considered as a valuable work.
- Given that IoT has faced a massive volume of data which are generated from heterogenous IoT devices, emerging technologies such as cloud computing and big data with IoT would provide a significant opportunity to expand more research in healthcare systems.
- In order to achieve the successful implementation and widespread adoption of IoT system in healthcare industry, it is suggested that proposed IoT architecture be tested and evaluated before implementing the actual system.

## Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this article.

## Acknowledgements

This study has been funded and supported by Iran University of Medical Sciences (Grant Number: 94-04-136-26859).

## References

- [1] Avachat V, Gupta P. A study of Semantic Middleware for Internet of Things. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2014; 4(12): 23-7.
- [2] Moosavi SR, Gia TN, Nigussie E, Rahmani AM, Virtanen S, Tenhunen H, et al. End-to-end security scheme for mobility enabled healthcare Internet of Things *Future Generation Computer Systems* 2016; 64: 108-24.  
<https://doi.org/10.1016/j.future.2016.02.020>
- [3] Ahmadi H, Arji G, Shahmoradi L, Safdari R, Nilashi M, Alizadeh M. The application of internet of things in healthcare: a systematic literature review and classification. *Universal Access in the Information Society* 2019; 18(4): 837-69.  
<https://doi.org/10.1007/s10209-018-0618-4>
- [4] Ungurean I, Brezulianu A. An Internet of Things Framework for Remote Monitoring of the HealthCare Parameters. *Advances in Electrical and Computer Engineering* 2017; 17(2): 11-6.  
<https://doi.org/10.4316/aece.2017.02002>
- [5] Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communications Surveys & Tutorials* 2015; 17(4): 2347-76.  
<https://doi.org/10.1109/comst.2015.2444095>
- [6] Sethi P, Sarangi SR. Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*. 2017; 2017(1):1-25.  
<https://doi.org/10.1155/2017/9324035>
- [7] Sharma D, Jinwala D. Functional encryption in IoT E-Health care system. *Lecture Notes in Computer Science (including subseries Lecture Notes in*

- Artificial Intelligence and Lecture Notes in Bioinformatics), 2015. p. 345-63. [https://doi.org/10.1007/978-3-319-26961-0\\_21](https://doi.org/10.1007/978-3-319-26961-0_21)
- [8] Pticek M, Podobnik V, Jezic G, Beyond the internet of things: The social networking of machines, *International Journal of Distributed Sensor Networks* 2016; 12(6): 1-15. <https://doi.org/10.1155/2016/8178417>
- [9] Ray PP. A survey on internet of things architectures. *Journal of King Saud University - Computer and Information Sciences* 2016; 30(3): 291-319. <https://doi.org/10.1016/j.jksuci.2016.10.003>
- [10] Yang L, Ge Y, Li W, Rao W, Shen W, A home mobile healthcare system for wheelchair users, In: *Proceeding of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2014, pp. 609-14. <https://doi.org/10.1109/cscwd.2014.6846914>
- [11] Uckelmann D, Harrison M, Michahelles F. *An Architectural Approach Towards the Future Internet of Things*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011, pp. 1-24. [https://doi.org/10.1007/978-3-642-19157-2\\_1](https://doi.org/10.1007/978-3-642-19157-2_1)
- [12] Katole B, Sivapala M, Suresh V. Principle elements and framework of internet of things. *International Journal Of Engineering And Science* 2013; 3(5): 24-9. <https://doi.org/10.4028/www.scientific.net/amm.303-306.2144>
- [13] Konstantinidis EI, Bamparopoulos G, Billis A, Bamidis PD. Internet of things for an age-friendly healthcare, *Studies in health technology and informatics* 2015; 210: 587-91. <https://doi.org/10.3233/978-1-61499-512-8-587>
- [14] Madakam S, Ramaswamy R, Tripathi S. Internet of Things (IoT): A literature review. *Journal of Computer and Communications* 2015; 3(05): 164-73. <https://doi.org/10.4236/jcc.2015.35021>
- [15] Pang Z, Chen Q, Han W, Zheng L. Value-centric design of the internet-of-things solution for food supply chain: Value creation, sensor portfolio and information fusion. *Information Systems Frontiers*. 2015; 17(2): 289-319. <https://doi.org/10.2991/emim-15.2015.61>
- [16] Aeman N, Patil DVM. Internet of things is an advanced concept of ICT for better human life. *International journal of pure and applied research in engineering and technology* 2016; 4(9): 1158-63.
- [17] Ray PP. Towards an Internet of Things based architectural framework for defence. In: *Proceeding of the 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2015, pp. 411-6. <https://doi.org/10.1109/ICCICCT.2015.7475314>
- [18] Patel KK, Patel SM, Professor PSA. Internet of Things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *Int J Eng Sci Comput* 2016; 6(5): 6122-31. <https://doi.org/10.4010/2016.1482>
- [19] Chiuchisan I, Costin HN, Geman O. Adopting the Internet of Things technologies in health care systems. In: *Proceeding of the 2014 International Conference and Exposition on Electrical and Power Engineering (EPE)*, 2014, pp. 532-5. <https://doi.org/10.1109/ICEPE.2014.6969965>
- [20] Hegde SG, Naveen Soumyalatha. Internet of things (IoT): A study on architectural elements, communication technologies and applications. *International Journal of Advanced Research in Computer and Communication Engineering* 2016; 5(9): 189-93. <https://doi.org/10.17148/IJARCC.2016.5943>
- [21] Ara T, Shah PG, Prabhakar M. Internet of Things Architecture and Applications: A Survey. *Indian Journal of Science and Technology* 2016; 9(45): 1-7. <https://doi.org/10.17485/ijst/2016/v9i45/106507>
- [22] Islam SMR, Kwak D, Kabir MH, Hossain M, Kwak KS, The internet of things for health care: A comprehensive survey, *IEEE Access* 2015; 3: 678-708. <https://doi.org/10.1109/ACCESS.2015.2437951>
- [23] Bhatia M, Sood SK. A comprehensive health assessment framework to facilitate IoT-assisted smart workouts: A predictive healthcare perspective. *Computers in Industry* 2017; 92–93: 50-66. <https://doi.org/10.1016/j.compind.2017.06.009>
- [24] Kolenik T, Gams M. Persuasive technology for mental health: One step closer to (mental health care) equality? *IEEE Technology and Society Magazine*. 2021;40(1):80-6. <https://doi.org/10.1109/MTS.2021.3056288>
- [25] Yang G, Xie L, Mäntysalo M, Zhou X, Pang Z, Xu LD, et al. A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor, and Intelligent Medicine Box. *IEEE Transactions on Industrial Informatics* 2014; 10(4): 2180-91. <https://doi.org/10.1109/TII.2014.2307795>
- [26] Qiu T, Chen N, Li K, Atiquzzaman M, Zhao W, How Can Heterogeneous Internet of Things Build our Future: A Survey. *IEEE Communications Surveys & Tutorials* 2018; 20(3) 1-18. <https://doi.org/10.1109/COMST.2018.2803740>
- [27] Ahmadi M, Jeddi FR, Gohari MR, Sadoughi F, A review of the personal health records in selected countries and Iran, *Journal of medical systems* 2012; 36(2): 371-82. <https://doi.org/10.1007/s10916-010-9482-3>
- [28] Mieronkoski R, Azimi I, Rahmani AM. Aantaa R, Terava V, Liljeberg P, et al. The Internet of Things for basic nursing care-A scoping review, *International journal of nursing studies* 2017; 69: 78-90. <https://doi.org/10.1016/j.ijnurstu.2017.01.009>
- [29] Darshan KR, Anandakumar KR. A comprehensive review on usage of Internet of Things (IoT) in healthcare system. In: *Proceeding of the 2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, 2015, pp. 132-6. <https://doi.org/10.1109/erect.2015.7499001>

- [30] Rahman AFA, Daud M, Mohamad MZ. Securing sensor to cloud ecosystem using internet of things (IoT) security framework. In: *Proceeding of the International Conference on Internet of things and Cloud Computing*, Cambridge, United Kingdom, 2016, pp. 1-5.  
<https://doi.org/10.1145/2896387.2906198>
- [31] Vargheese R, Viniotis Y. Influencing data availability in IoT enabled cloud based e-health in a 30 day readmission context, In: *Proceeding of the 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2014, pp. 475-80.  
<https://doi.org/10.4108/icst.collaboratecom.2014.257621>
- [32] Lopes NV, Pinto F, Furtado P, Silva J. IoT architecture proposal for disabled people. In: *Proceeding of the International Conference on Wireless and Mobile Computing, Networking and Communications*, 2014, pp. 152-8.  
<https://doi.org/10.1109/WiMOB.2014.6962164>
- [33] Pir A, Akram MU, Khan MA. Internet of things based context awareness architectural framework for HMIS. In: *Proceeding of the 2015 17th International Conference on E-Health Networking, Application and Services, HealthCom 2015*, 2015, pp. 55-60.  
<https://doi.org/10.1109/HealthCom.2015.7454473>
- [34] Jalali R, El-Khatib K, McGregor C, Smart city architecture for community level services through the internet of things, In: *Proceeding of the 2015 18th International Conference on Intelligence in Next Generation Networks, ICIN 2015*, 2015, pp. 108-13. <https://doi.org/10.1109/ICIN.2015.7073815>
- [35] Mi YW, Wu RC, Li YY, Zhang K, Wang W, Li T. Architecture and the key technologies of medical equipment management system based on the internet of things. *J Comput Theor Nanosci* 2016; 13(12): 9722-7.  
<https://doi.org/10.1166/jctn.2016.5914>
- [36] Muhammad AP, Akram MU, Khan MA. Survey based analysis of internet of things based architectural framework for hospital management system. In: *Proceeding of the 2015 13th International Conference on Frontiers of Information Technology (FIT)*, 2015, pp. 271-6.  
<https://doi.org/10.1109/FIT.2015.54>
- [37] Gupta N, Saeed H, Jha S, Chahande M. Pandey S, Implementation of an IOT framework for smart healthcare. In: *Proceeding of the 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, 2017, pp. 622-7.  
<https://doi.org/10.1109/ICECA.2017.8203613>
- [38] Tokognon CJA, Gao B, Tian GY, Yan Y. Structural Health Monitoring Framework Based on Internet of Things: A Survey. *IEEE Internet of Things Journal* 2017; 4(3): 619-35.  
<https://doi.org/10.1109/jiot.2017.2664072>
- [39] Croes E, Hoepman J-H. *Software architectural styles in the internet of things*. Netherlands: Radboud University Nijmegen; 2015.
- [40] Sancho G, Tazi S, Villemur T. A semantic-driven auto-adaptive architecture for collaborative ubiquitous systems. In: *Proceeding of the Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology*, 2008, pp. 650-5.  
<https://doi.org/10.1145/1456223.1456354>
- [41] Mohammed J, Lung CH, Ocneanu A, Thakral A, Jones C, Adler A. Internet of Things: Remote Patient Monitoring Using Web Services and Cloud Computing. In: *Proceeding of the 2014 IEEE International Conference on Internet of Things (iThings)*, and *IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 2014, pp. 256-63. <https://doi.org/10.1109/iThings.2014.45>
- [42] Meier JD, Hill D. *Microsoft application architecture guide*, 2nd Edition. United States: Microsoft Corporation; 2009. p. 560.
- [43] Abdmeziem MR, Tandjaoui D, Romdhani I. *Architecting the Internet of Things: State of the Art*. ed. Springer International Publishing, Cham, 2016.  
[https://doi.org/10.1007/978-3-319-22168-7\\_3](https://doi.org/10.1007/978-3-319-22168-7_3)
- [44] Mustafa T, Varol A. Review of the internet of things for healthcare monitoring. 2020 8th International Symposium on Digital Forensics and Security (ISDFS); 1-2 June 2020. <https://doi.org/10.1109/ISDFS49300.2020.9116305>
- [45] Moher D, Liberati A, Tetzlaff J, Altman DG. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *BMJ* 2009; 339: b2535.  
<https://doi.org/10.1136/bmj.b2535>
- [46] Domingo MC. An overview of the Internet of Things for people with disabilities. *Journal of Network and Computer Applications* 2012; 35(2): 584-96. doi:  
<https://doi.org/10.1016/j.jnca.2011.10.015>
- [47] Huang JM. Research on Application of Internet of Things in Nursing Home. *Applied Mechanics and Materials*. 2013;303–306:2153–6.  
<https://doi.org/10.4028/www.scientific.net/amm.303-306.2153>
- [48] Lee JD, Yoon TS, Chung SH, Cha HS. Service-Oriented Security Framework for Remote Medical Services in the Internet of Things Environment. *Healthcare Informatics Research* 2015; 21(4): 271-82. <https://doi.org/10.4258/hir.2015.21.4.271>
- [49] Lu DX, Liu T. The Developing of Medical System in Information Age. *Advanced Materials Research*. 2013;749: 139–43. <https://doi.org/10.4028/www.scientific.net/amr.749.139>
- [50] Wang D, Ge WC, Sun LP, Li JH. Neonatal Nursing Information System Based on Internet of Things. *Applied Mechanics and Materials*. 2013; 401–403:1927–30. <https://doi.org/10.4028/www.scientific.net/amm.401-403.1927>
- [51] Yu L, Lu Y, Zhu X. Smart hospital based on internet of things. *Journal of Networks* 2012; 7(10): 1654-61.

- <https://doi.org/10.4304/jnw.7.10.1654-1661>
- [52] Wu Z, Liu LH, Huang YS, Wu XM. A community health service architecture based on the internet of things on health-care. In: *Proceeding of the IFMBE Proceedings*, 2013, pp. 1317-20. [https://doi.org/10.1007/978-3-642-29305-4\\_345](https://doi.org/10.1007/978-3-642-29305-4_345)
- [53] Wang X. The architecture design of the wearable health monitoring system based on internet of things technology. *Int J Grid Util Comput* 2015; 6(3-4): 207-12. <https://doi.org/10.1504/IJGUC.2015.070681>
- [54] Wang Y. The Construction of the Psychological Health Education Platform Based on Internet of Things. *Applied Mechanics and Materials*. 2014; 556–562:6711–5. <https://doi.org/10.4028/www.scientific.net/amm.556-562.6711>
- [55] Qi J, Yang P, Fan D, Deng Z. A survey of physical activity monitoring and assessment using internet of things technology. In: *Proceeding of the Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUC 2015, 13th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2015 and 13th IEEE International Conference on Pervasive Intelligence and Computing, PICom 2015*, 2015, pp. 2353-8. <https://doi.org/10.1109/CIT/IUC/DASC/PICOM.2015.348>
- [56] Keh HC, Shih CC, Chou KY, Cheng YC, Ho HK, Yu PY, et al. Integrating unified communications and internet of M-health things with micro wireless physiological sensors. *Journal of Applied Science and Engineering* 2014; 17(3): 319-28. <https://doi.org/10.6180/jase.2014.17.3.12>
- [57] Azimi I, Anzanpour A, Rahmani AM, Pahikkala T, Levorato M, Liljeberg P, et al. HiCH: Hierarchical Fog-Assisted Computing Architecture for Healthcare IoT. *ACM Trans Embed Comput Syst* 2017; 16(5s): 1-20. <https://doi.org/10.1145/3126501>
- [58] Al-Tae MA, Al-Nuaimy W, Al-Ataby A, Muhsin ZJ, Abood SN. Mobile health platform for diabetes management based on the Internet-of-Things. In: *Proceeding of the 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 2015, pp. 1-5. <https://doi.org/10.1109/AEECT.2015.7360551>
- [59] Li J, Zhou H, Zuo D, Hou KM, De Vault C. Ubiquitous health monitoring and real-time cardiac arrhythmias detection: a case study. *Bio-medical materials and engineering* 2014; 24(1): 1027-33. <https://doi.org/10.3233/bme-130900>
- [60] Farahani B, Firouzi F, Chang V, Badaroglu M, Constant N, Mankodiya K. Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Generation Computer Systems* 2018; 78: 659-76. <https://doi.org/10.1016/j.future.2017.04.036>
- [61] Thangaraj M, Ponmalar PP, Anuradha S. Internet Of Things (IOT) Enabled Smart Autonomous Hospital Management System - A Real World Health Care Use Case with the Technology Drivers. In: *Proceeding of the 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2015, pp. 174-81. <https://doi.org/10.1109/ICCIC.2015.7435678>
- [62] Khan SF. Health Care Monitoring System in Internet of Things (IoT) by Using RFID. In: *Proceeding of the IEEE's 6th International Conference on Industrial Technology and Management (ICITM 2017)*, 2017, pp. 198-204. <https://doi.org/10.1109/ICITM.2017.7917920>
- [63] Ullah K, Shah MA, Zhang SJ. Effective Ways to Use Internet of Things in the Field of Medical and Smart Health Care. 2016, pp. 372-9. <https://doi.org/10.1109/intelse.2016.7475151>
- [64] Jingjing Y, Shangfu H, Xiao Z, Benzhen G, Yu L, Beibei D, et al. Family health monitoring system based on the four sessions internet of things. *Telkomnika Telecomun Compt Electr Control* 2015; 13(1): 314-20. <https://doi.org/10.12928/TELKOMNIKA.v13i1.1265>
- [65] Ray PP. Generic Internet of Things architecture for smart sports. In: *Proceeding of the 2015 International Conference on Control Instrumentation Communication and Computational Technologies, ICCICCT 2015*, 2015, pp. 405-10. <https://doi.org/10.1109/ICCICCT.2015.7475313>
- [66] Kocabas O, Soyata T, Aktas MK. Emerging Security Mechanisms for Medical Cyber Physical Systems. *IEEE/ACM Trans Comput BioL Bioinf* 2016; 13(3): 401-16. <https://doi.org/10.1109/TCBB.2016.2520933>
- [67] Wang X, Bie R, Sun Y, Wu Z, Zhou M, Cao R, et al. The architecture of an automatic eHealth platform with mobile client for cerebrovascular disease detection. *J Med Internet Res* 2013; 15(8). <https://doi.org/10.2196/mhealth.2550>
- [68] Qi J, Yang P, Min G, Amft O, Dong F, Xu L. Advanced internet of things for personalised healthcare systems: A survey. *Pervasive and Mobile Computing* 2017; 41: 132-49. <https://doi.org/10.1016/j.pmcj.2017.06.018>
- [69] Meng XJ, Cui HQ, Hua R. An IoT-Based Remote Health Monitoring and Management System. In: *Proceeding of the Applied Mechanics and Materials*, 2014, pp. 1176-9. <https://doi.org/10.4018/978-1-7998-2584-5.ch014>
- [70] Wan J, Al-Awlaqi M, Li MS, O'Grady M, Gu X, Wang J, et al. Wearable IoT enabled real-time health monitoring system. *Eurasip J Wireless Commun Networking* 2018. <https://doi.org/10.1186/s13638-018-1308-x>
- [71] Bhatia M, Sood SK. Temporal Informative Analysis in Smart-ICU Monitoring: M-HealthCare Perspective. *J Med Syst* 2016; 40(8): 190. <https://doi.org/10.1007/s10916-016-0547-9>

- [72] Rathore MM, Ahmad A, Paul A, Wan J, Zhang D. Real-time Medical Emergency Response System: Exploiting IoT and Big Data for Public Health. *J Med Syst* 2016; 40(12).  
<https://doi.org/10.1007/s10916-016-0647-6>
- [73] Ray PP. Home Health Hub Internet of Things (H(3)IoT): An Architectural Framework for Monitoring Health of Elderly People. In: *Proceeding of the 2014 International Conference on Science Engineering and Management Research (ICSEMR)*, 2014, pp. 1-3.  
<https://doi.org/10.1109/ICSEMR.2014.7043542>
- [74] Ngo Manh K, Saguna S, Mitra K, Ahlund C. IReHMo: An efficient IoT-based remote health monitoring system for smart regions, In: *Proceeding of the 2015 17th International Conference on E-Health Networking, Application and Services, HealthCom 2015*, 2015, pp. 563-8.  
<https://doi.org/10.1109/HealthCom.2015.7454565>
- [75] Xiao ZR, Lv BG, Wang X, Zhao YJ. A Healthcare Service System Based on Internet of Things. In: *Proceeding of the Advanced Materials Research*, 2013, pp. 1903-7. <https://doi.org/10.4028/www.scientific.net/AMR.774-776.1903>
- [76] Almehmadi T, Alshehri S, Tahir S. A Secure Fog-Cloud Based Architecture for MIIoT. In: *Proceeding of the 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, 2019, pp. 1-6.  
<https://doi.org/10.1109/CAIS.2019.8769524>
- [77] Qi J, Yang P, Waraich A, Deng Z, Zhao Y, Yang Y. Examining sensor-based physical activity recognition and monitoring for healthcare using Internet of Things: A systematic review. *Journal of Biomedical Informatics* 2018; 87: 138-53.  
<https://doi.org/10.1016/j.jbi.2018.09.002>
- [78] Bublitz FM, Oetomo A, Sahu KS, Kuang A, Fadrique LX, Velmovitsky PE, et al. Disruptive technologies for environment and health research: An overview of artificial intelligence, blockchain, and internet of things. *International journal of environmental research and public health* 2019; 16(20): 3847.  
<https://doi.org/10.3390/ijerph16203847>
- [79] Gupta BB, Quamara M. An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurr Comput.* 2018.  
<https://doi.org/10.1002/cpe.4946>
- [80] Telecommunication standardization sector of ITU. *Global information infrastructure, internet protocol aspects and next-generation networks next generation networks—frameworks and functional architecture models*. Geneva, Switzerland: International Telecommunication Union, 2012. p. 1-14.
- [81] An BW, Shin JH, Kim S-Y, Kim J, Ji S, Park J, et al. Smart sensor systems for wearable electronic devices. *Polymers*. 2017; 9(8): 303.  
<https://doi.org/10.3390/polym9080303>
- [82] Ni Q, Garcia Hernando AB, de la Cruz IP. The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development. *Sensors (Basel, Switzerland)* 2015; 15(5): 11312-62.  
<https://doi.org/10.3390/s150511312>
- [83] Alarifi A, Al-Salman A, Alsaleh M, Alnafessah A, Al-Hadhrami S, Al-Ammar MA, et al. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors (Basel, Switzerland)* 2016; 16(5): 1-36.  
<https://doi.org/10.3390/s16050707>
- [84] Gholamhosseini L, Sadoughi F, Ahmadi H, Safaei A. Health Internet of Things: Strengths, Weakness, Opportunity, and Threats. In: *Proceeding of the 2019 5th International Conference on Web Research (ICWR)*, Tehran, Iran, Iran 2019, pp. 287-96.  
<https://doi.org/10.1109/ICWR.2019.8765286>
- [85] Al-Sarawi S, Anbar M, Alieyan K, Alzubaidi M. Internet of Things (IoT) communication protocols: Review. In: *Proceeding of the 2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685-90.  
<https://doi.org/10.1109/ICITECH.2017.8079928>
- [86] Chakkor S, Cheikh E, Mostafa B, Hajraoui A. Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Applications, *International Journal of Advanced Computer Science and Applications*. 2014; 5.  
<https://doi.org/10.14569/IJACSA.2014.050413>
- [87] Verma A, Prakash S, Srivastava V, Kumar A, Mukhopadhyay SC. Sensing, Controlling, and IoT Infrastructure in Smart Building: A Review, *IEEE Sensors J* 2019; 19(20): 9036-46.  
<https://doi.org/10.1109/JSEN.2019.2922409>
- [88] Shakerighadi B, Anvari-Moghaddam A, Vasquez JC, Guerrero J. Internet of Things for Modern Energy Systems: State-of-the-Art, Challenges, and Open Issues. *Energies* 2018; 11.  
<https://doi.org/10.3390/en11051252>
- [89] Aqeel-ur-Rehman, Mehmood K, Baksh A. Communication Technology That Suits IoT - A Critical Review. In: *Proceeding of the, Berlin, Heidelberg*, 2013, pp. 14-25.  
[https://doi.org/10.1007/978-3-642-41054-3\\_2](https://doi.org/10.1007/978-3-642-41054-3_2)
- [90] Aly H, Elmogy M, Barakat S. Big Data on Internet of Things: Applications, Architecture, Technologies, Techniques, and Future Directions, *Int J Comput Sci Eng* 2015; 4: 300-13.
- [91] Talavera JM, Tobón LE, Gómez JA, Culman MA, Aranda JM, Parra DT, et al., Review of IoT applications in agro-industrial and environmental fields, *Computers and Electronics in Agriculture* 2017; 142, Part A: 283-97.  
<https://doi.org/10.1016/j.compag.2017.09.015>
- [92] Gonzalez E, Pena R, Avila A, Vargas-Rosales C, Munoz-Rodriguez D. A Systematic Review on Recent Advances in mHealth Systems: Deployment Architecture for Emergency

- Response. *Journal of healthcare engineering*. 2017; 2017: 1-13. <https://doi.org/10.1155/2017/9186270>
- [93] Khan R, Khan SU, Zaheer R, Khan S. Future internet: the internet of things architecture, possible applications and key challenges. In: *Proceeding of the 2012 10th International Conference on Frontiers of Information Technology (FIT)*, 2012, pp. 257-60. <https://doi.org/10.1109/FIT.2012.53>
- [94] Khoi NM, Saguna S, Mitra K, Ahlund C. IReHMo: An efficient IoT-based remote health monitoring system for smart regions. In: *Proceeding of the 2015 17th International Conference on E-Health Networking, Application and Services, HealthCom*. 2015, pp. 563-8. <https://doi.org/10.1109/HealthCom.2015.7454565>
- [95] Datta SK, Bonnet C, Gyrard A, Costa RPFd, Boudaoud K. Applying Internet of Things for personalized healthcare in smart homes. In: *Proceeding of the 2015 24th Wireless and Optical Communication Conference (WOCC)*, 2015, pp. 164-9. <https://doi.org/10.1109/WOCC.2015.7346198>
- [96] Firdausi A. Overview the internet of things (IoT) system security, applications, architecture and business models. Indonesia: Universitas of Electrical Engineering; 2016.
- [97] Darwish D. Improved Layered Architecture for Internet of Things. *International Journal of Computing Academic Research (IJCAR)*. 2015; 4(4): 214-23.
- [98] Mijac M, Androcec D, Picek R. Smart city services driven by IoT: A systematic review. *Journal of Economic and Social Development*. 2017; 4(2): 40-50.
- [99] Tsai C-W, Lai C-F, Vasilakos AV. Future Internet of Things: open issues and challenges. *Wireless Networks*. 2014; 20(8): 2201-17. <https://doi.org/10.1007/s11276-014-0731-0>
- [100] Erfannia L, Sadoughi F, Sheikhtaheri A. The advantages of implementing cloud computing in the health industry of Iran: A qualitative study. *International Journal of Computer Science and Network Security*. 2018; 18(1): 198-206.
- [101] Thangaraj M, Ponmalar PP, Sujatha G, Anuradha S. Agent based semantic internet of things (IoT) in smart health care. In: *Proceeding of the Proceedings of the The 11th International Knowledge Management in Organizations Conference on The changing face of Knowledge Management Impacting Society, Hagen, Germany, 2016* pp. 1-9. <https://doi.org/10.1145/2925995.2926023>
- [102] Guo H, Ren J, Zhang D, Zhang Y, Hu J. A scalable and manageable IoT architecture based on transparent computing. *Journal of Parallel and Distributed Computing* 2018; 118: 5-13. <https://doi.org/10.1016/j.jpdc.2017.07.003>
- [103] Dalli A, Bri S. Acquisition devices in internet of things: RFID and sensors, *J Theor Appl Inf Technol* 2016; 90(1): 194-200.
- [104] Pacheco J, Zhu X, Badr Y, Hariri S. Enabling Risk Management for Smart Infrastructures with an Anomaly Behavior Analysis Intrusion Detection System. In: *Proceeding of the 2017 IEEE 2nd International Workshops on Foundations and Applications of Self\* Systems (FAS\*W)*, 2017, pp. 324-8. <https://doi.org/10.1109/FAS-W.2017.167>
- [105] Hossain MS, Muhammad G. Cloud-assisted Industrial Internet of Things (IIoT) - Enabled framework for health monitoring. *Computer Networks* 2016; 101: 192-202. <https://doi.org/10.1016/j.comnet.2016.01.009>
- [106] Bilal M, A Review of Internet of Things Architecture, Technologies and Analysis Smartphone-based Attacks Against 3D printers. arXiv preprint arXiv:170804560 2017.
- [107] Vijayalakshmi A, Arockiam L, A study on security issues and challenges in IoT. *Int J Eng Sci Manage Res*. 2016;3(11):34-43.
- [108] Ross R, Graubart R, Bodeau D, McQuaid R. *Systems Security Engineering: Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems*. United States: NIST Special Publication 800-160, 2018 p. 1-142.
- [109] Turkanovi'c M. Authentication and Key Agreement Protocol for Ad Hoc Networks Based on the Internet of Things Paradigm. 2016; 40 (1):153-4.
- [110] Zhou X, Jin Z, Fu Y, Zhou H, Qin L. Short Signcryption Scheme for the Internet of Things. *Informatica*. 2011;35 (4): 521-30.
- [111] Ebrahimi M, Sayadhighighi M, Jolfaei A, Shamaeian N, Tadayon MH. A secure and decentralized trust management scheme for smart health systems. *IEEE J Biomed Health Inform*. Forthcoming 2021. <https://doi.org/10.1109/JBHI.2021.3107339>
- [112] Alam T. IBchain: Internet of Things and Blockchain Integration Approach for Secure Communication in Smart Cities. 2021; 45(3):477-86. <https://doi.org/10.31449/inf.v45i3.3573>



# Keyphrase Extraction Model: A New Design and Application on Tourism Information

Ngo Le Huy Hien

School of Built Environment, Engineering, and Computing, Leeds Beckett University, Leeds, England

E-mail: n.hien2994@student.leedsbeckett.ac.uk

Ho Minh Hoang and Nguyen Van Hieu

The University of Danang - University of Science and Technology, Danang, Vietnam

Est Rouge Technologies JSC, Vietnam

E-mail: hoanghm@tech.est-rouge.com, nvhieuqt@dut.udn.vn

Nguyen Van Tien

The University of Danang - University of Science and Technology, Danang, Vietnam

E-mail: vannt808@gmail.com

**Keywords:** elasticsearch, keyphrase extraction, conditional random field, BERT, BiLSTM-CRF, long short-term memory

**Received:** April 5, 2021

*Keyphrase extraction has recently become a foundation for developing digital library applications, especially in semantic information retrieval techniques. From that context, in this paper, a keyphrase extraction model was formulated in terms of Natural Language Processing, applied explicitly in extracting information and searching techniques in tourism. The proposed process includes collecting and processing data from tourism sources such as Tripadvisor.com, Agoda.com, and vietnam-guide.com. Then, the raw data was analyzed and pre-processed with labeling keyphrase and fed data forward to Pretrained BERT model and Bidirectional Long Short-Term Memory with Conditional Random Field. The model performed the combination of Bidirectional Long Short-Term Memory with Conditional Random Field in order to solve keyphrase extraction tasks. Furthermore, the model integrated the Elasticsearch technique to enhance performance and time of looking up tourism destinations' information. The outcome extracted key phrases produce high accuracy and can be applied for extraction problems and textual content summaries.*

*Povzetek: Predstavljen je pristop na osnovi ključnih fraz za uporabo v turističnih sistemih.*

## 1 Introduction

In the science of natural language processing, the analysis of sentences into phrases, labeling, and marking has been a point of interest in research and application in various aspects. Keyphrase Extraction is the process of extracting key phrases that contain important content of a document. Keyphrases are used to solve information extraction content clustering, text classification, and text summary problems [16]. Numerous studied methodologies have been widely applied in academic issues such as Key2Vec [2] - automatically extracting keywords from scientific articles, Sequence Labeling [1] - extracting keyphrase from scholarly documents. The process normally used the BiLSTM [1] model, combining a pre-trained model to extract corresponding keywords of a dataset. Then, the search engine operated through API using NoSQL Elasticsearch, which uses scoring techniques from the keyphrases of documents corresponding to the database [19].

Research that applies in traveling newspapers and documents would support tourism information searching

from many traveling sites. From there, the Keyphrase Extraction method allows visitors to easily select and quickly search based on their own words without clearly understanding their desired places. Furthermore, based on official data analysis from tourist sites, visitors will avoid unreliable information of some locations related to their own needs. The method would help increase the experience and satisfaction when visitors come and learn information about the city.

From the aforementioned method and benefits, this study proposed a new design and application to assist in searching tourist information. The application can be implemented as a phone app or a tourist information website that optimally serves tourist demand for their first steps in a new destination and acquire typical characteristics of the sit shown on media. This research can play a novel and practical keyphrase extraction model and contribute to the science of extraction applications and textual content summaries.

## 2 Related works

### 2.1 Keyphrase extraction

Keyphrase Extraction is a subfield of Information extraction problem to extract keyphrases from documents according to given requirements. Currently, there are many studies and methods given in this problem with many different approaches, as mentioned below.

#### 2.1.1 Unsupervised method

The unsupervised learning method provides probability models based on the word input that determines the frequency and importance of keywords in the text to identify keyword phrases. Some of the unsupervised learning techniques such as TextRank [3], Sgrank [4], and WikiRank [5] help extract keyphrases based on a narrow context of context (identifying the meaning of the word or by probability). Therefore, these methods' accuracy is also limited, but this is a basic method and has been applied in many problems to instantly perform labeling for supervised learning methods.

#### 2.1.2 Self-supervised method

Recent studies focus on self-supervised learning in the field of information extraction. When the data is initially trained with the labels, the machine will then label and learn itself based on the relationships between the new input information and the previously trained information. These are superior studies such as SRES [6] - extracting information on the Web, and SelfORE [7] - extracting from natural language sentences their open-domain relation facts. Studies have introduced new approaches to training as well as solutions for information extraction.

#### 2.1.3 Supervised method

In this method, the data is labeled corresponding to the keyphrases before training through a machine learning model. Parallel with the development in data and computing hardware; deep learning has been increasingly popular and widely used to optimize. For example, the research about Long Short-Term Memory (LSTM) [8] effectively covers the neighbor contexts [17]. BiLSTM-CRF [9] used the Glove representation model to embed input words and return positive results when experimenting in an academic dataset.

## 2.2 Contextual embedding

Before training, input data have to be normalized into sets of vectors. Word embedding is a form of word representation, representing words with related meanings to have similar representations.

There are many studies and experiments in implementing algorithms supporting word embedding and vocabulary modeling. And Contextual Embedding is one of the SOTA techniques to vectorize documents based on meaning and contextual relations. The enhancement of Contextual embedding compared with others embedding models such as Word2Vec [10], Glove [14] is the addition

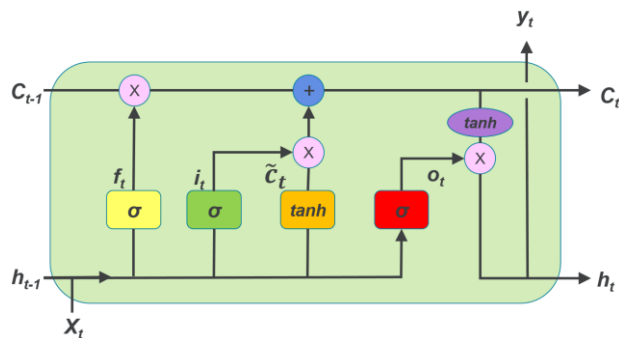


Figure 1: Structure of 1 cell LSTM.

of context for vectors generated through position, thereby increasing the accuracy in terms of context and semantics.

BERT [11] was introduced as a breakthrough in the field of natural language processing, with improvements in text modeling with the application of Transformer architecture to train context-based word representations.

In this article, the authors combined the improvement of word representation models with sequence labeling techniques for extracting keyphrases in tourism documents.

## 3 Methodology

Let  $d = \{w_1, w_2, \dots, w_n\}$  be input document, and  $w_i$  represents the  $i^{th}$  token. Each word in  $d$  was labeled into 3 classes of set  $Y = \{K_B, K_I, K_O\}$ , where  $K_B$  indicates that  $w_i$  is the beginning keyphrase,  $K_I$  denotes that  $w_i$  is in the keyphrase, and  $K_O$  marks that  $w_i$  is out of the keyphrase.

### 3.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) [8] is a form of Recurrent Neural Network (RNN) model [13] for solving problems of sequence data based on previously learned information to predict the current information in the sequence. LSTM is a solution to resolve the Vanishing Gradient issue of a primitive RNN network when information is learned from far away in the chain and lost its importance. In order to achieve this, LSTM uses several "gates" that store information remotely. Especially, Bidirectional LSTM (BiLSTM) is a generalization technique covering the context information in both directions [12].

Each word in the text was mapped for embedding size vector  $x_i$ , so that the sequence  $d$  of length  $n$  will be represented by a vector.

$x = \{x_1, x_2, \dots, x_n\}$  was labeled accordingly with  $y = \{y_1, y_2, \dots, y_n\}$  where  $y_i \in Y$ .

The input of LSTM is a  $[h_{t-1}, x_t]$  vector at time  $t$ , with the cell state of the network  $c_t$ , and the output vector between the two times  $t$  and  $t+1$  is  $h_t$ .

LSTM unit has 4 gates: forget gate  $f_t$ , input gate  $i_t$ , output gate  $o_t$ , and memory cell  $c_t$ , which are represented by the following equations:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f); \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); \tag{2}$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}, x_t] + b_c); \tag{3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o); \tag{4}$$

$$h_t = o_t * \tanh(c_t); \tag{5}$$

in which the activation function are:  $\sigma$  (sigmoid) and  $\tanh$ , and  $*$  is element-wise multiplication.  $W$  and  $b$  are model parameters, and  $h_t$  is hidden state.

In the BiLSTM model, 2 used LSTM architectures progress simultaneously and independently to model the input sequence in 2 directions: from left to right (direction  $\vec{h}_t$  - Figure 2a) and from right to left (direction  $\overleftarrow{h}_t$  - Figure 2b).

Where  $\vec{h}_t$  represents the information from preceding word of  $w_t\{w_1, w_2, \dots, w_{t-1}\}$ , and  $\overleftarrow{h}_t$  represents the information from succeeding words of  $w_t\{w_{t+1}, w_{t+2}, \dots, w_n\}$ . Vector  $\overleftrightarrow{h}_t$  represents for word  $w_t$  in input sequence  $d$  when concatenating 2 vectors  $\vec{h}_t$  and  $\overleftarrow{h}_t$ .

$$\overleftrightarrow{h}_t = [\vec{h}_t; \overleftarrow{h}_t]$$

Then the results were mapped to vector  $f_t$  where

$$f_t = W_a \overleftrightarrow{h}_t$$

in which  $W_a$  is weight vector that has a shape of

$$|Y| \times |\overleftrightarrow{h}_t| = 3 \times |\overleftrightarrow{h}_t|$$

The output vector of BiLSTM model after multiplying weight matrix is

$$f = \{f_1, f_2, \dots, f_n\}$$

in which  $f$  is the input of the CRF layer.

### 3.2 Conditional Random Field

Conditional Random Field (CRF) is a probabilistic model for structured predictive problems and has been used very successfully in machine learning areas. CRF is used in conjunction with deep learning models to increase the efficiency for segmentation and sequence data labeling [18].

As the input data in CRF is sequential, the previous context must be considered before predicting a data point, thereby increasing the model's accuracy. For example, if the previous label is B-P (begin phrase), the following tag is most likely I-P.

In this study, a 378-dimensional vector was used representation for each word following the BERT (BERT-base) model. A BERT's pre-trained model's architecture and some layers were added to match the problem. Then, the original layer parameters were fine-tuning, and the additional layer parameters were re-trained from the beginning. In this way, the proposed model could reduce the training time while ensuring its accuracy.

### 3.3 Elasticsearch (NoSQL)

This study used the NoSQL Elasticsearch database management system because of its ability to analyze data and statistics. A node is an Elasticsearch server, which is logically independent of each other. In fact, a node can run on one (usually in a development or test environment) or multiple physical servers (usually in a production

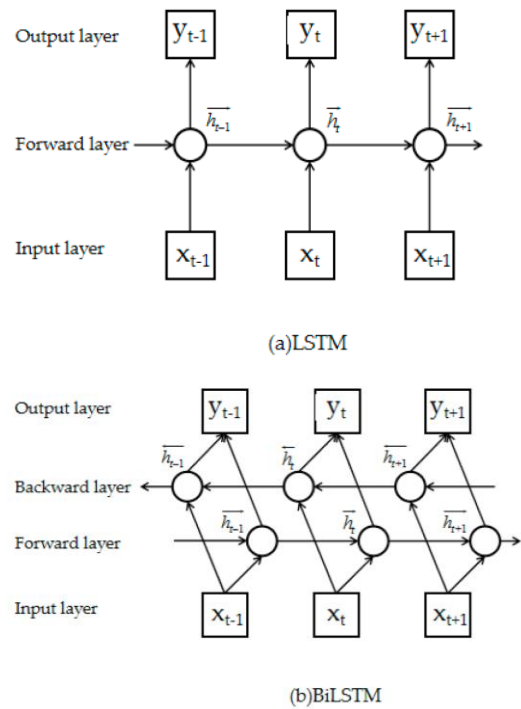


Figure 2: The architecture of LSTM and BiSLTM.

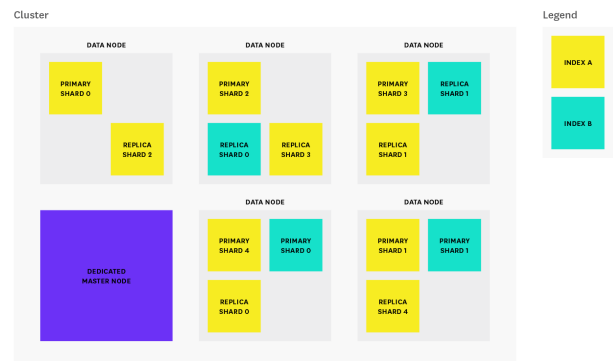


Figure 3: Elasticsearch architecture.

environment). A collection of nodes working together forms a cluster; each node in the cluster contains a portion of that cluster's data. And all the data of a cluster will be divided among the nodes [20].

Nodes have three different types: master, data, and client. A cluster automatically selects a node as the master from its nodes. The master node will be responsible for coordinating the work of the cluster, such as distributing shards and creating/deleting indexes. Only the master node has the ability to update the cluster's state. In essence, Apache's Lucene - a full-text search - uses a data structure called an inverted index to perform searches with high performance. The architecture of Elasticsearch is indicated in Figure 3 [21][22].

Elasticsearch operates on a private server, communicates through RESTful APIs, and provides near real-time.

## 4 Proposed method and architecture

This paper reveals a new method in keyphrase extraction from travel documents, using the sequence labeling technique with pre-trained model BERT (Bidirectional Encoder Representations from Transformers). Then, the model applies BiLSTM with Conditional Random Field in the training phase to enhance its result.

### 4.1 Proposed model

**Pre-processing:** The input data was pre-processed by encoding each word (token), along with whether the word tag corresponds to the keyphrase or not. The labels are B-P (Begin Phrase), I-P (In Phrase), and O (Out Phrase). Those labels were encoded with the input data into numeric labels 0, 1, 2, respectively.

For example:

Restaurant has a big view of natural landscape.

O O O B-P I-P I-P I-P I-P

Pre-train model BERT: There are currently many different versions of the BERT model. All versions are based on the transformation of the Transformer architecture, focusing on 3 parameters:

L: The number of block sub-layers in transformer,

H: Embedding vector size (or hidden size),

A: The number of heads in a multi-head layer, each head operates one self-attention.

The research used the pre-trained BERT base uncased model (L = 12, H = 768, A = 12) to represent the input vocabulary into vectors containing information about the vocabulary and its context. The BERT model input consists of a sequence of coded words, and the output is a lexical vector representing each input word.

**BiLSTM-CRF model:** Output vectors of BERT are the inputs of the BiLSTM-CRF model. They were passed through the 2-dimensional LSTM network, and the information will be trained in two dimensions of the context, in terms of firm magnetism and context. Next, the output was passed through the CRF layer with labels marked previously to train and extract key phrase information in the text.

### 4.2 The process

To implement the data effectively, the proposed model applied a process that is depicted in Figure 5. In the beginning, raw data were pre-processed by filters and removed noise data, including HTML, tag, link, unrelated text, etc.

In the training phase, each sentence of the text was labeled. If the phrase is at the beginning, it would be labeled as B-P (begin phrase), the rest of the keyphrase should be labeled as I-P (in phrase), and other words considered as O-P (out phrase). After that, the processed data was fed into the BERT model in Contextual Embedding stages before forwarding to BiLSTM layers. Then the data was fed into CRF layers after attaching labels. The output weight was used in the evaluation stage

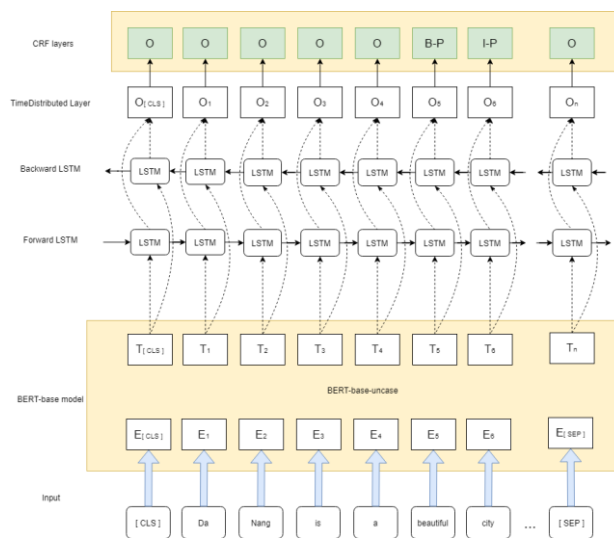


Figure 4: Proposed model BERT-BiSTM-CRF.

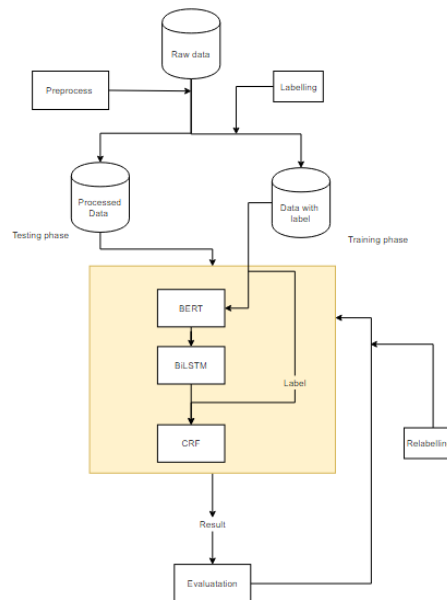


Figure 5: The proposed model process.

with new input; new input results were relabeled and became the architecture input to re-train.

In the testing phase, the real raw data was also pre-processed and then fed into the model to predict the keyphrase.

### 4.3 Application architecture

The output keyphrases were stored in a database and synchronized to a NoSQL database before applying the search engine of Elasticsearch. Since Elasticsearch operates on a private server through RESTful APIs, the proposed model can fit for a large feature set with the real-time processing ability of Elasticsearch.

The system architecture is presented in Figure 6 with an artificial intelligence system integrated with an API layer. The API contributes as a communication channel

between the server and the client to perform querying, processing, and returning results to UI [23]. First, the data collected from travel websites will be trained and stored in a NoSQL database (MongoDB). Then the data synchronized from Cluster Mongo, containing collected data for Elasticsearch. The APIs were used in retrieving data from Elasticsearch to fetch and return the results to the user since Elasticsearch is only effective in retrieving data.

## 5 Results

### 5.1 Training result

The study was conducted on a dataset with two models: one applies Glove embedding, and the other uses BERT embedding. After the experiment, Table 1 shows that BERT embedding as a pre-trained model indicates better results with the Recall value of about 0.891.

The two graphs in Figure 7 indicate the loss and accuracy of the proposed approach based on the number of epochs of the training phase. The Loss and Accuracy indexes at the first two epochs present the ideal trends

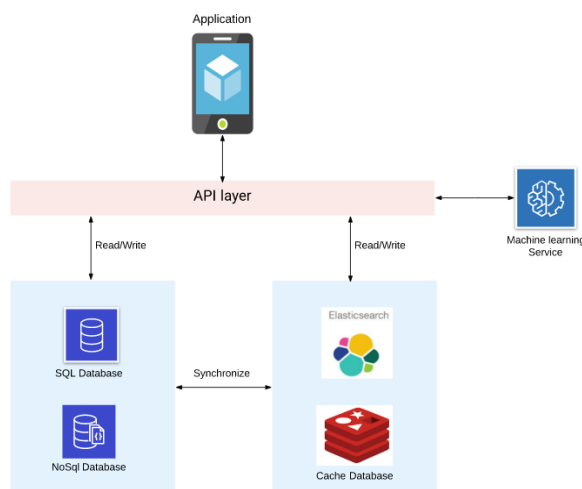


Figure 6: The proposed application architecture.

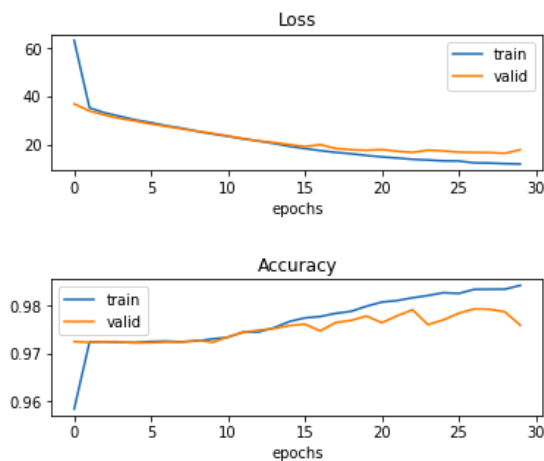


Figure 7: Training results based on number of epochs.

while the next epochs take a slight improvement in the Loss and Accuracy results.

### 5.2 Test result

The study also analyzed the proposed model with travel datasets that were pre-processed and labeled. The datasets were crawled into different text blocks [15] from tourism resources, including newspapers, descriptions, documents, and comments on Tripadvisor and Agoda about destinations and attractions.

The data was pre-processed by removing noise information such as images, HTML tags, web page scripts, and other irrelevant comments. In this case, the experiment focused on English data after removing other languages' information.

The results are shown in Tables 2 and 3, corresponding to different amounts of training paragraph.

It is recognized from Tables 2 and 3 that when the training and testing data increase, the predicted

	Precision	Recall	F1 score	Exact match
BERT Embedding	0.954	0.891	0.921	0.282
Glove Embedding	0.956	0.724	0.824	0.211

Table 1: Comparison between BERT embedding and Glove embedding.

Training paragraph	Original Keyphrase (test)	Predicted Keyphrase	Accuracy	Training Time (s)
200	4681	1724	161	9.13
400	4681	3122	900	13.91
600	4681	2683	1075	17.30
873	4681	3574	2033	20.78

Table 2: Prediction results based on training data.

Training paragraph	Precision	F1 score	Recall
200	0.577	0.310	0.212
400	0.650	0.613	0.580
600	0.715	0.727	0.739
873	0.874	0.794	0.729

Table 3: Result indices corresponding to testing data based on training data.

Sgrank	BERT-BiLSTM-CRF
‘Danang is a developing coastal city in the central part of <u>Vietnam</u> and is known as one of the <u>largest cities</u> alongside <u>Hanoi</u> and <u>Ho Chi Minh city</u> . Visiting <u>Da Nang</u> , you will be astounded by the <u>amazing natural landscape</u> , the <u>friendly locals</u> and a <u>countless number of great attractions</u> around the city.’	‘Danang is a <u>developing coastal city</u> in the central part of Vietnam and is known as one of the largest cities alongside Hanoi and Ho Chi Minh city. Visiting Da Nang, you will be astounded by the <u>amazing natural landscape</u> , the <u>friendly locals</u> and a countless number of great attractions around the city.’

Table 4: Result in application<sup>1</sup>.

keyphrases and the correct keyphrases also increase, indicating the efficiency of the model. Moreover, the training time for each experiment is applicable in real practice.

## 6 Application

The study examined the BiLSTM-CRF model combined with the BERT Embedding layer and compared the result with the Sgrank method.

Table 4 showcases the actual results of the model when predicting a completely new input. It is recognized that the proposed model focuses on phrases of nouns and adjectives from B-P and I-P labels. Although the Sgrank method produces many phrases, it has a high error rate and focuses on nonspecific adjectives and nouns.

## 7 Conclusion

In this article, a keyphrase extraction method was proposed, which uses the BiLSTM-CRF deep learning model with the BERT pre-trained model's lexical representation. The output of the BERT (encoder) model became the input of the BiLSTM-CRF model to perform the keyphrase extraction task.

With a supervised learning method, the proposed method has outweighed previous models in terms of accuracy of words' context and meaning. In addition, the study has built an API system for applications integrated with actual text extraction. The presented method has helped extract key phrases in the text with high accuracy (from 40% on sample data), thereby can be applied for extraction problems and textual content summaries.

Future work may gear towards expanding the model and proposing a software architect to conduct an application supporting tourism for different cities around the world. Based on each characteristic of each destination (from keyphrases), a recommendation system could be developed to support users in finding their next desired destinations. Furthermore, the authors aim to continue expanding the applications of the model into different languages (rather than English) and various fields, not only in tourism.

## 7.1 Acknowledgement

This research is funded and implemented for the Mercury project of Est Rouge Technologies JSC, Vietnam.

## References

- [1] Alzaidy, Rabah; Caragea, Cornelia; GILES, C. Lee. Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In: The world wide web conference. 2019. p. 2551-2557. <https://doi.org/10.1145/3308558.3313642>.
- [2] Mahata, Debanjan, et al. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018. p. 634-639. <http://dx.doi.org/10.18653/v1/N18-2100>.
- [3] Mihalcea, Rada; Tarau, Paul. Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing. 2004. p. 404-411.
- [4] Danesh, Soheil; Sumner, Tamara; Martin, James H. Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In: Proceedings of the fourth joint conference on lexical and computational semantics. 2015. p. 117-126. <http://dx.doi.org/10.18653/v1/S15-1013>.
- [5] Yu, Yang; Ng, Vincent. Wikirank: Improving keyphrase extraction based on background knowledge. arXiv:1803.09000, 2018.
- [6] Feldman, Ronen, et al. Self-supervised relation extraction from the web. In: International Symposium on Methodologies for Intelligent Systems. Springer, Berlin, Heidelberg, 2006. p. 755-764. <https://doi.org/10.1007/s10115-007-0110-6>.
- [7] HU, Xuming, Et Al. SelfORE: Self-supervised Relational Feature Learning for Open Relation Extraction. arXiv:2004.02438, 2020.
- [8] Hochreiter, Sepp; Schmidhuber, Jürgen. Long short-term memory. Neural computation, 1997, 9.8: 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [9] Alzaidy, Rabah; Caragea, Cornelia; Giles, C. Lee. Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In: The world wide web conference. 2019. p. 2551-2557. <https://doi.org/10.1145/3308558.3313642>.
- [10] Mikolov, Tomas, et al. Efficient estimation of word representations in vector space. arXiv:1301.3781, 2013.
- [11] Devlin, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805, 2018.
- [12] Graves, Alex; Fernández, Santiago; Schmidhuber, Jürgen. Bidirectional LSTM networks for improved phoneme classification and recognition. In: International conference on artificial neural

- networks. Springer, Berlin, Heidelberg, 2005. p. 799-804. [https://doi.org/10.1007/11550907\\_126](https://doi.org/10.1007/11550907_126).
- [13] David, Rumelhart. *Recurrent Neural Networks*, 1986.
- [14] Pennington, Jeffrey; Socher, Richard; Manning, Christopher D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014. p. 1532-1543. <http://dx.doi.org/10.3115/v1/D14-1162>.
- [15] Hien, Ngo Le Huy; Tien, Thai Quang; Van Hieu, Nguyen. Web Crawler: Design and Implementation for Extracting Article-Like Contents. *Cybernetics and Physics*, 2020, 9.3: 144-151. <https://doi.org/10.35470/2226-4116-2020-9-3-144-151>.
- [16] Witten, Ian H., et al. Kea: Practical automated keyphrase extraction. In: *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*. IGI global, 2005. p. 129-152. <https://doi.org/10.4018/978-1-59140-441-5.ch008>.
- [17] Hien, Ngo Le Huy; Van Hieu, Nguyen. Recognition of Plant Species using Deep Convolutional Feature Extraction. *International Journal on Emerging Technologies*, 2020, 11.3: 904-910. <https://doi.org/10.14445/22315381/IJETT-V68I4P205S>.
- [18] Zhang, Chengzhi. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 2008, 4.3: 1169-1180.
- [19] Hien, Ngo Le Huy; Huy, Luu Van; Van Hieu, Nguyen. Artwork style transfer model using deep learning approach. *Cybernetics and Physics*, 2021, 10.3: 127-137. <https://doi.org/10.35470/2226-4116-2021-10-3-127-137>.
- [20] Munezero, Myriam, et al. Automatic detection of antisocial behaviour in texts. *Informatica*, 2014, 38.1: 3-10.
- [21] Azam, Irfan, and Sajid Ali Khan. Feature extraction trends for intelligent facial expression recognition: A survey. *Informatica*, 2018, 42.4: 507-514. <https://doi.org/10.31449/inf.v42i4.2037>.
- [22] Chen, Feng, and Shi Zhang. Information Visualization Analysis of Public Opinion Data on Social Media. *Informatica*, 2021, 45.1: 157-162. <https://doi.org/10.31449/inf.v45i1.3426>.
- [23] Menai, Mohamed El Bachir. Word sense disambiguation using an evolutionary approach. *Informatica*, 2014, 38.3: 155-169.





# Reduced Number of Parameters for Predicting Post-Stroke Activities of Daily Living Using Machine Learning Algorithms on Initiating Rehabilitation

Ali Mohammad Alqudah and Munder Al-Hashem

Department of Biomedical Systems and Informatics Engineering, Yarmouk University, Irbid, Jordan

E-mail: ali\_qudah@hotmail.com, munderalhashem@gmail.com

Amin Alqudah

Department of Computer Engineering, Yarmouk University, Irbid, Jordan

E-mail: amin.alqudah@yu.edu.jo

**Keywords:** Barthel Index scale (BI), Activities of Daily Living (ADL), stroke, features selection, Machine Learning (ML)

**Received:** June 01, 2021

*The estimation of the Barthel Index scale (BI) is a significant method for measuring the performance of Activities Daily Living (ADL), where the prediction of ADL is crucial for providing rehabilitation care management and recovery for patients after stroke, therefore in this paper, nine various Machine Learning (ML) algorithms were implemented in a medical dataset contains 776 records from 313 patients 208 of them are men: 208 and 150 are women with multiple features collected from them for predicting and classifying the BI status as clinical decision support for determining the ADL of post-stroke patients. Meanwhile, we have applied feature selection using the chi-squared test to reduce the number of features in the dataset. The results showed that the Decision Tree (DT), XGBoost (XGB), and AdaBoost (ADB) classifiers performed the highest performance achieved with 100% correctness in terms of accuracy, sensitivity, specificity, error, and Area Under Curve (AUC) on both the full and reduced features datasets.*

*Povzetek: V prispevku je predlagana metodologija za zmanjšanje števila parametrov za napovedovanje primernih aktivnosti po možganski kapi.*

## 1 Introduction

A stroke is a medical condition that occurs when the blood flow to the brain's parts is reduced or interrupted and frustrating the brain tissues from receiving nutrients and oxygen which leads to dying the brain cells in few minutes [1]. Stroke is a dilemma that needs urgent care and attention from around. Many people in the whole world are suffering from stroke and almost two-thirds of those individuals survive but need rehabilitation and recovery. Patients with long-standing disabilities after stroke could face many load difficulties in their life living whether physical, society, family and mental, therefore, the main object of rehabilitation and recovery is by observing the patients functions after a stroke and monitoring the level of independence to achieve the greatest potential Activities of Daily Living (ADL) [2].

Predicting the ADL is pivotal and crucial for effective use and careful deal with patients after stroke especially in the first months. Furthermore, ADL provides an overview of how the person's independence in its life and the sufficient support and the health care provided by both the government and patient's family [2]. For example, in 2016 only every 40 seconds in the USA only there is a new onset of stroke events appeared [3], meanwhile, in 2012 in Taiwan there was around 230 patients were admitting hospital every day due to the occurrence of acute stroke

[3]. Due to this, if the disabilities and impairments remained for a long time in patients whose affected by a stroke this may lead to a massive mental, physical, and also a heavy financial load for the patients and their families and this all can be override by just make a rehabilitation for stroke patients as fast as possible [3]. The main target of rehabilitation the stroke patient is to ensure that they returned to their life after sophisticated training of them to make them able to make activities of daily living (ADL) independently. Moreover, the first stage of rehabilitation is to accurately predict ADL because the accurate prediction of ADL is vital to make optimization for stroke management in the first two to three months following a stroke attack [3].

During the rehabilitation process of the patient with strokes, different assessments and parameters should be checked for patients of Post-acute Care-Cerebrovascular Diseases (PAC-CVD) program to provide important and detailed information about the patient's health from various aspects [3]. The collected parameters include Barthel index (BI), modified Rankin scale (MRS), functional oral intake scale (FOIS), mini nutrition assessment (MNA), instrumental activities of daily living scale (IADL), Berg balance test (BBT), gait speed, 6-min walk test (6MWT), Fugl-Meyer upper extremity

assessment (FuglUE), modified Fugl–Meyer sensory assessment (FuglISEN), mini-mental state examination (MMSE), motor activity log (MAL), and concise Chinese aphasia test (CCAT). Using these multiple assessment parameters (features) we can generate and predict more comprehensive data and a conclusion about the current statuses of patients than single assessment features. However, providing an accurate tool for the prediction of ADL is not available due to the contribution of other features on the prognosis of ADL. To solve this problem, machine learning (ML) systems can be used [1].

ML is considered as an application of Artificial Intelligence (AI) that provides the ability to automatically learn and improve from the experienced learned. ML directly focuses on building a mathematical model to be used in the prediction process based on patterns extracted from the experienced learned data whether to be a large or complex dataset. Latterly, ML algorithms are extensively used in real-life applications and the healthcare field. The Healthcare field is directly related to human life that's because using the ML will intensify, enhance, improve, and reduce the error level of rates in the medical diagnosis [1, 2, 3]. This research presented several ML algorithms for predicting the ADL after the stroke of patients to give information about the person's independence based on multiple features by predicting BI score. Where the BI score is a widely used and very well-known parameter and tool to assess the independence of functions of ADL especially with post-stroke patients. To predict apply machine learning we use a dataset of 18 parameters 17 of them as features and the last one represents the BI score where the dataset contains 776 records from 313 patients 208 of them are men: 208 and 150 are women. We used nine different ML algorithms which are as follows; Logistic Regression (LR), K Nearest Neighbor (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), XGBoost (XGB), AdaBoost (ADB), and Artificial Neural Network (DNN). The algorithms are estimated by using training and testing methodology and the performance evaluation is calculated utilizing the accuracy, sensitivity, specificity, error, Receiver Operating Characteristic (AUC), and confusion matrix.

This paper is organized as follows: Section II describes the related work; Section III explains the materials and method of the post-stroke dataset used in our study and various ML algorithms applied to the multiple features. In section IV we present the results of the suggested methodology; Section V represents a discussion of the results of the proposed research. Finally, we conclude the paper in Section VI.

## 2 Literature review

In this section, the most recent works related to the topic of stroke rehabilitation prediction using machine learning are discussed. Janne et al [2] presented a study on the early post-stroke stage for final ADL to recognize the variables and the outcome of Activities of Daily Living (ADLs) after stroke. The method was by distinguishing the high and low quality for the risk of bias and qualitative

synthesis. The results showed that the median risk of bias was 17 out of 27.

Yin Lin et al [3] proposed a method for predicting the activities of daily living (ADL) of post-stroke patients. Many Machine Learning (ML) algorithms are used such as Logistic Regression (LR), Support Vector Machine (SVM), Linear Regression, and Random Forest (RF) to estimate the Barthel index (BI). The results showed the RF and LR were higher for the Area Under the Curve (AUC) with 0.79 rather than the SVM algorithm with AUC of 0.77. Moreover, the Mean Absolute Error (MAE) of both linear regression and SVM was 9.95 and 9.86, respectively.

Chu Pai et al [4] proposed a study of testing and predicting the time change of Activities Daily Living (ADLs) of post-stroke patients within 24 hours and the whole months of the year. The ADLs were estimated by the Barthel Index (BI). A latent growth curve model was utilized to analyze the dynamic variations in ADLs. The results showed that the time following a stroke increases, survivors attend to continuously upgrade with interest to ADLs. Moreover, the below levels of the primary ADLs were related to the larger growth of ADLs over time.

Douiri et al [5] developed a prediction monitoring model for the post-stroke patient's recovery and estimated its clinical use. Activities of Daily Living (ADLs) were evaluated utilizing the Barthel Index (BI) for many weeks after the stroke. Penalized linear blended models were generated to predict the functional recovery of patients. The results of the prediction recovery curve showed good accuracy for Root Mean Squared Deviation (RMSD) of 3 BI points to one year. The passive predictive values of risk poor recovery with BI less than 8 at three and 12 months.

Bertolin et al [6] presented a study of predicting acute stroke patients after mild to moderate. Many predictor variables were measured for predicting the ADLs. The results showed physical variables described more variety in ADLs than the communication, memory, and thinking results. The Short-Blessed Test (SBT) was the unique meaningful independent foreteller of communication, memory, and thinking, while the National Institute of Health Stroke Scale (NIHSS) was the only one that measures safely predicted ADLs.

Glanella et al [7] presented a study on investigating the Functional Independence Measure (FIM) of ADLs as potential predictors of the outcomes after stroke through inpatient rehabilitation. Regression analysis was used through Two backward stepwise to estimate the most independent variables. The results showed that social interaction, grooming, upper body dressing, and bowel control are the most independent and important variables for reviewing the FIM of ADLs.

Lee et al [8] proposed a study for developing a computational method to distinguish the potential of predicting the Quality of Life (QOL) after post-stroke rehabilitation. Five classifiers were used by using personal factors and nine functional issues to estimate the QOL. The results showed that the Particle Swarm-Optimized Support Vector Machine (PSO-SVM) gives the most extraordinary accuracy (58.33%), the highest cross-

validated accuracy (74.29%) which results as the best classifier in predicting the QOL of stroke patients.

### 3 Materials and methods

In this section, we will explain the dataset used and our recommended methodology, then making a comparison with the various classifier's performances.

#### 3.1 Materials

In this paper, the data proposed in paper [3] has been used. The dataset contains 776 patient's data and 15 features. The features contain the following gender, age, acute ward stay, LOS, BI admission, MRS, FOIS, MNA, IADL, BBT, 6MWT, FugIUE, MMSE, MAL Quality, BI discharge. These values are collected from patients and based on them the output (Class) is determined. Table 1 shows the features and class values for the used dataset. collected from the patient based on the following criteria:

- 1 Stroke onset time must be within the last month.
- 2 Hemodynamic parameters should be stable within the last 72 hours.
- 3 No neurological deterioration within the last 72 hours.
- 4 Sufficient cognition function and ability to learn rehabilitation exercise (Especially MRS between 2 and 4).

While they mentioned that the exclusion criteria for the patient include:

The dataset authors mentioned that the features are

1. Stroke onset time > 1 month.
2. Patients with end-stage renal disease or those receiving dialysis therapy.

After applying these requirements above, every three weeks before discharge all qualifying patients receive a detailed post-stroke admission status test. The ratings contain the following characteristics (MRS, BI, FOIS, MNA, QoL, IADL, BBT, gait speed, 6MWT, FugIUE, FugISEN, MMSE, MAL, and CCAT). During the recovery ward, both tests were used to illustrate the gravity of the BI status or the BI score forecasts for the rehabilitation unit discharge. Besides, an explanatory declaration is made of patients' age and longevity in the acute stroke ward before referral to PAC-CVD features [3]. The summary of the dataset is shown in Table 2.

#### 3.2 Methodology

The suggested study as shown in Figure 1 below is by firstly solving the problem of multiclass of BI score, then applying nine classification algorithms and make a comparison between them to predict the BI score which represents the ADL of stroke patients.

#### 3.3 Data preparation

Based on the BI at discharge from the PAC-CVD ward, it shows that the BI score is categorized with different values. The BI score with a value of 20 represents as independent ADL, the BI score with a value range of 15 - 19 represent mild dependent ADL, and the BI score with a value range of 10 - 15 represent as moderate dependent

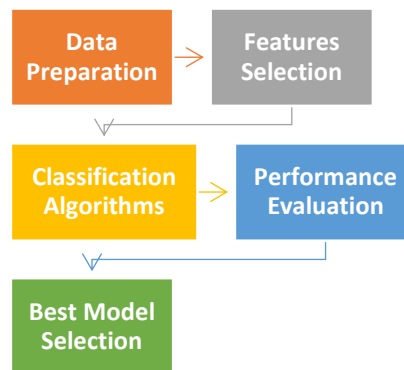


Figure 1: The block diagram of the proposed study.

ADL. Therefore, the multiclass classification problem of BI score can be solved by discretizing the BI into 3 ordinal classes.

##### 3.3.1 Features selections

In this paper for results enhancement and time reduction in future patient BI score prediction, we will perform a Chi-squared test for features selections. The Chi-squared test is well known statistical test of independence that is usually used to test and determine whether two categorical variables are independent or not. Features selection concept using this test is very simple where we find if the feature is independent of the target or not. If the feature is dependent, then the feature will likely not be useful in predicting the target and not selected otherwise if they are not independent the feature will most likely have predictive power on the target and selected [9].

##### 3.3.2 Classification algorithms

Classification in general is a plan that consists of orderly grouping something according to some features or standards. it is a type of supervised machine learning in which an algorithm "learns" to classify new observations from examples of labeled data and is used to develop models that predict what group or class that something belongs to [10-13]. Classification is a two-step process. During the first step, the model is created by applying a classification algorithm on a training data set then in the second step, the extracted model is tested against a predefined test data set to measure the model's trained accuracy. So, classification is the process to assign a class label from a dataset whose class label is unknown [13-16]. The following subsections will discuss the classifiers used in our paper.

##### Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised learning algorithm under machine learning, and it is used for both classification and regression tasks. But most often the supporting vector machine is used in the classification. The Support Vector Machine works by looking at a set of training examples, and then marking each of them as belonging to one or another of two classes, the SVM training algorithm builds a model that maps new examples to one class or another, making it a non-probability binary linear classifier [10].

Table 1: Dataset Features, Range, and Description.

Feature	Range	Description
Gender	Male: 1, Female: 2	Gender
Age	35 -85 Years	Age
Acute Ward Stay	7-20 Days	length of stay in acute stroke ward
LOS	8-25days	length of stay in rehabilitation ward
BI Admission	20 mean independent ADL, 15-19 mean mild dependent, 0-14 mean moderate dependent, and 5–9 mean severe dependent.	Barthel index on admission to PAC-CVD ward
MRS	0- 2 mean mild disability, and 3 - 5 mean moderate to severe disability.	Modified Rankin Scale
FOIS	Range from 1 to 7 where Level 1 mean nothing by mouth to Level 7 which mean total oral diet with no restrictions.	functional oral intake scale
MNA	<8 mean malnutrition, 8–11 mean risk of malnutrition, and >11 mean no malnutrition.	mini nutrition assessment
IADL	0 mean low function (dependent), and 8 mean high function (independent).	instrumental activities of daily living scale
BBT	Range from 0 to 4 determining sitting and standing balance)	Berg balance test
Gait	Range from 0.3 m/s to 1 m/s.	gait speed
6MWT	Range from 210 m to 300 m.	6-min walk test
FuglIUE	0 mean cDNNot perform, 1 mean partially, and 2 mean fully.	Fugl–Meyer upper extremity assessment
FuglISEN	0 mean cDNNot perform, 1 mean partially, and 2 mean fully.	Motor recovery assess the sensorimotor impairment in individuals who have had stroke
MMSE	MMSE $\geq 24$ mean normal, MMSE =19-23 mean mild, MMSE = 10-18 mean moderate, and MMSE $\leq 9$ mean severe.	mini-mental state examination
MAL Quality	0 mean never, 1 mean poor, MAL =2 mean fair, and 3 mean normal.	Motor activity log, quality of use arm function
BI Discharge	20 mean independent ADL, 5-19 mean mild dependent, 10-14 mean moderate dependent, and BI =5–9 mean severe dependent.	Barthel index at discharge from PAC-CVD ward
Class		Patient Classification Value

### Naïve Bayes (NB)

Naive Bayes Classifiers are based on Bayesian classification techniques. This is based on Bayles's theorem, which is an equation defining the relationship between the conditional probability of statistical quantities. In the case of the Bayesian classification, we are interested in finding the likelihood of a category considering some of the characteristics that have been observed. In general, Naive Bayes is a simple, but powerful and widely used machine learning classifier. It is a probabilistic classifier that allows classifications in a Bayesian setting using the Maximum A Posteriori

decision law. It can also be interpreted by a very simple Bayesian network. Naive Bayes Classifiers are highly common for text classification and are a standard approach to problems such as spam detection. The naive Bayes classifier does not need any parameters setup [11].

### Decision Tree (DT)

The Decision Tree algorithm is part of the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can also be used to solve regression and classification problems [12]. The purpose of the Decision Tree is to create a training model that can be used to predict the class or value

Table 2: Dataset Features Statistical Description.

Statistical Value	Standard Deviation	Mean	Minimum	Maximum	25% Percentile	50% Percentile	75% Percentile
Age	4.104 472	13.840 21	7	20	10	14	18
Acute Ward Stay	5.297 994	16.559 28	8	25	12	17	21
LOS	3.502 869	14.592 78	9	20	11	15	18
BI Admission	1.703 339	2.5979 38	0	5	1	3	4
MRS	2.016 674	4.0631 44	1	7	2	4	6
FOIS	4.036 867	8.2719 07	2	15	5	8	12
MNA	2.616 618	3.8402 06	0	8	1	4	6
IADL	1.409 733	1.9677 84	0	4	1	2	3
BBT	26.66 156	254.35 7	21 0	30 0	231	253	277
6MWT	0.809 418	1.0180 41	0	2	0	1	2
FugIUE	7.027 365	18.752 58	7	30	13	19	25
MMSE	1.104 851	1.4497 42	0	3	0	1	2
MAL Quality	3.092 184	16.430 41	11	20	14	17	20
BI Discharge	14.79 479	60.333 76	35	85	48	61	73

of the target variable by learning basic decision rules derived from previous results (training data). In Decision Trees, to predict the class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. Based on a comparison, we follow the branch corresponding to the value and leap to the next node. Decision trees classify the examples by sorting them down from the root tree to some leaf/terminal node, with the leaf/terminal node giving the example classification. Each node in the tree serves as a test case for some attribute, and each edge coming down from the node corresponds to the possible answers to the test case. This process is recursive and is repeated for every subtree rooted in a new node [12].

#### Random Forests (RF)

The Random Forest (RF) Classifier, first proposed by Breiman [13], is one of the most common classification tools and an excellent set of machine learning techniques. The key principle of the RF classifier is to construct a classification tree based on a few randomly selected features of randomly selected samples with a bagging technique. Designed trees are used to vote for an input vector to get a class name. RF classifiers are built by several simple learners, where each basic learner is an

individual binary tree following recursive partitioning. RF has many advantages; it has better precision than other classifiers, allows large-scale data efficiency, does not bypass, and can be conveniently extended to multi-class inputs. The RF classifier has demonstrated superior classification efficiency over the other suggested methods since it was proposed [10, 11].

#### eXtreme Gradient Boosting (XGBoost)

XGBoost is a Machine Learning algorithm based on the decision-tree, using a gradient enhancement method. Artificial neural networks tend to outdo all other algorithms or systems in prediction problems involving unstructured data (images, messages, etc.). In small to medium structured/tabular data, however, decision-tab based algorithms are now regarded as best-in-class. For advances in tree-based algorithms over the years please see the map below. At its inception, this algorithm has not only been regarded as the guiding force behind the hood but many leading-edge applications in the industry [14].

#### Adaptive Boosting (AdaBoost)

AdaBoost is an ensemble learning system that was originally designed to improve the performance of binary classifiers (also known as "Meta-Learning"). AdaBoost uses an iterative approach to learn from and improve the

errors of weak classifiers. AdaBoost is a common stimulation strategy, attempting to combine several weak categorizers to create a powerful category. A single classifier cannot reliably predict an object's class, but we can build up such a strong model by grouping several weak classifications with each of them eventually learning from the incorrectly categorized objects of the others. The above classifier may be any of your basic classifiers, from Decision Trees (often the default) to Logistic Regression, etc [15].

**Deep Neural Network (DNN)**

A deep neural network (DNN) is an artificial neural network (DNN) with several layers between the input and output layers. There are various types of neural networks, but they are all made up of the same components: neurons, synapses, weights, beliefs, and functions. These elements are identical to the human brain and can be learned like any other ML algorithm [16, 17]. DNNs can model complex non-linear relationships. DNN architectures create compositional models in which the object is expressed as a layered composition of primitives. Extra layers permit the composition of features from lower layers, theoretically modeling complex data with fewer units than an equally powerful shallow network. The deep architecture contains a variety of versions of a few simple approaches. Different architectures have achieved success in particular fields. It is not always possible to assess the performance of different architectures unless they are tested on the same data sets [18]. DNNs are usually feed-forward networks in which data flows from the input layer to the output layer without looping back. Next, the DNN maps simulated neurons and assigns arbitrary numerical values, or "weights," to the relations between them. The weights and inputs are multiplied, and the product is returned between 0 and 1 [19]. If a certain pattern was not correctly understood by the network, an algorithm would change the weights. This way, the algorithm will make those parameters more influential before it decides the right mathematical manipulation of the data to be completely processed [20].

**Performance evaluation**

Evaluating the performance of machine learning algorithms is an essential part of any project. The model leads to giving you pleasing results when estimated using metric indices such as accuracy, specificity, sensitivity, error, and AUC [10-20]. Calculating these metrics by using True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) can the outputs of the proposed system compared to reference data. and consequently, the accuracy, sensitivity, precision, specificity, error, and AUC were evaluated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Error = 100 - Accuracy$$

**Best model selection**

Table 3: Training and testing dataset division.

Post Stroke Dataset	Training set	Testing set
	620	156

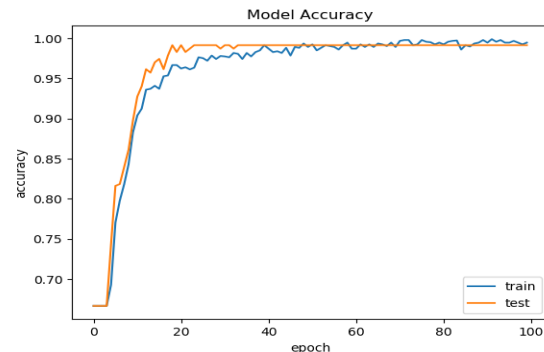


Figure 2: Training and Validation Accuracy.

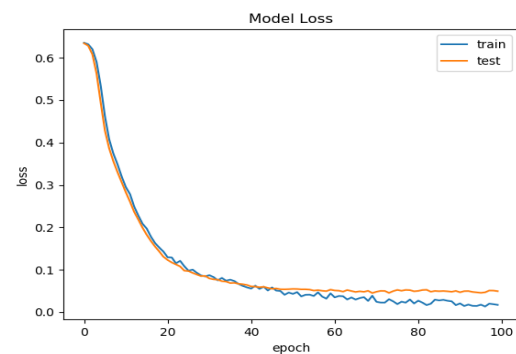


Figure 3: Training and Validation Loss.

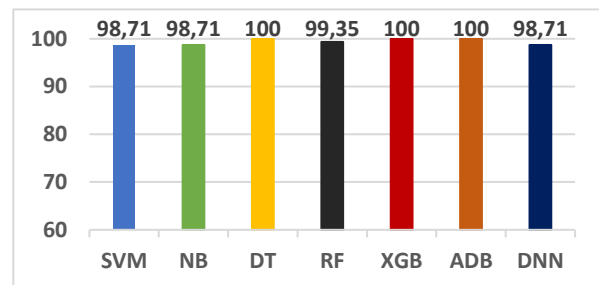


Figure 4: Accuracy performance evaluation.

For each ML algorithm that trains to the training dataset, the advantage of the learning algorithm here comes through the observed patterns of the training data that can do such as mapping to the input data attaches to the target, which is the required answer to be predicted, then creating ML model which makes capturing for these patterns. Moreover, the testing dataset is next implemented to the ML models to check and test their performances, but after comparing the results from the nine different algorithms that used, the most proper and appropriate model with the best accuracy, sensitivity, specificity, error, and AUC result is chosen to estimate and predict the BI as was discretized into 3 ordinal classes [10].

## 4 Results

The proposed methodology was employed with 80% - 20% training and testing division on the post-stroke dataset as shown in Table 3. Nine different ML classifiers are used for classifying the digitizing BI score that represents the ADL. The multi-classification process was successful in predicting the BI score with a high level of performance.

### 4.1 Features selection results

Before starting training and testing the used classifiers with and without features selection method first we applied the features selection methodology and select the top five features and the worst five features. The top five features selected by Chi-squared test methodology were ordered as follows: BI Admission, BI Discharge, 6MWT, Age, and FOIS. While the worst five were ordered as follows: Acute Ward Stay, FugIUE, MRS, Gender, and MAL Quality. The top five features are only used when we used the features selected classifiers.

### 4.2 Classifiers results without features selection

The nine classifiers were implemented for predicting as follows; the SVM classifier was performed with a linear kernel function. Moreover, the RF classifier was implemented too with 5 trees in the forest. Finally, the DNN classifier was trained by utilizing the Adaptive Moment Learning Rate (ADAM) solver with an initial learning rate value of 0.001. Figure 2 shows the training and validation process while Figure 3 shows the training and validation loss.

The hyperparameters of SVM, RF, DNN classifiers were chosen based on the principle of GridSearchCV which is a library function that helps to loop through predefined hyperparameters and fit the model on the training set. So, in the end, it can select the best parameters from the listed hyperparameters. The other classifiers that were also used were NB, DT, XGB, and ADB. The results of all algorithms employed show an extremely high level of results based on all the dataset features used. Figure 4 shows the results of accuracy, while Figure 5 shows the results for the error, where Figure 6 shows the results for the AUC, and finally, Figure 7 shows the results for the sensitivity and specificity of the three discretized classes that all obtained from the nine different ML algorithms.

### 4.3 Classifiers results with features selection

The same classifiers settings are performed on the selected features using the Chi-squared test where only five features are fed to the classifiers on the following results. Figure 8 shows the training and validation process while Figure 9 shows the training and validation loss.

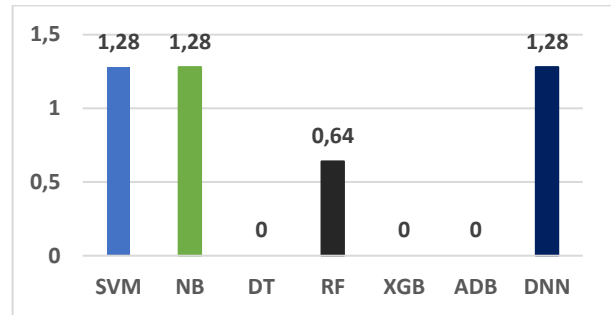


Figure 5: Error performance evaluation.

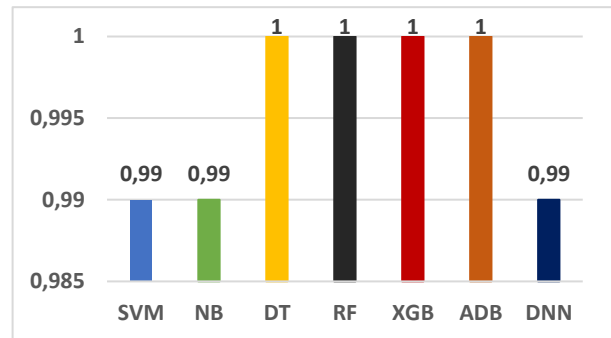


Figure 6: AUC performance evaluation.



Figure 7: Sensitivity and specificity performance evaluation of the three classes.

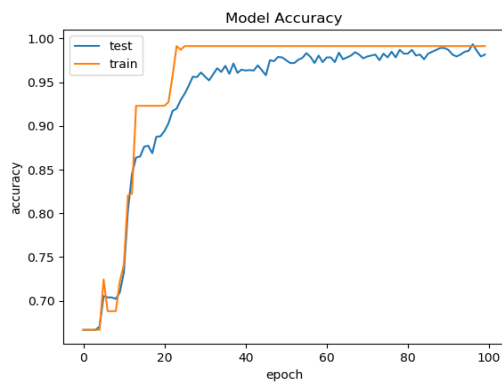


Figure 8: Training and Validation Accuracy.

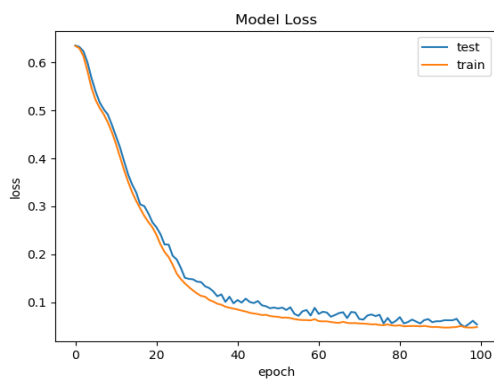


Figure 9: Training and Validation Loss.

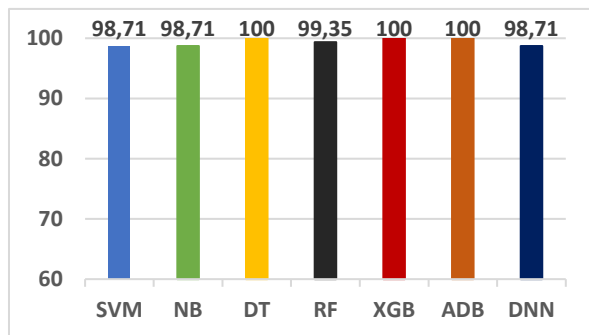


Figure 10: Accuracy performance evaluation.

The hyperparameters of SVM, RF, DNN classifiers were also chosen based on the principle of GridSearchCV which is a library function that helps to loop through predefined hyperparameters and fit the model on the training set. So, in the end, it can select the best parameters from the listed hyperparameters. The other classifiers that were also used were NB, DT, XGB, and ADB. The results of all algorithms employed show an extremely high level of results based on all the dataset features used. Figure 10 shows the results of accuracy, while Figure 11 shows the results for the error, where Figure 12 shows the results for the AUC, and finally, Figure 13 shows the results for the sensitivity and specificity of the three discretized classes that all obtained from the nine different ML algorithms.

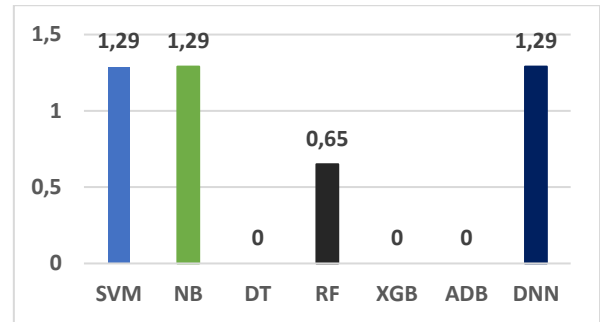


Figure 11: Error performance evaluation.

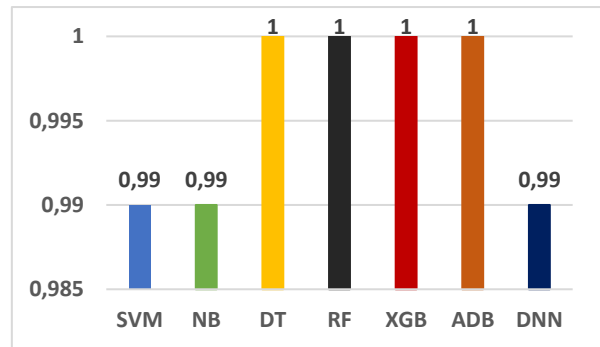


Figure 12: AUC performance evaluation.

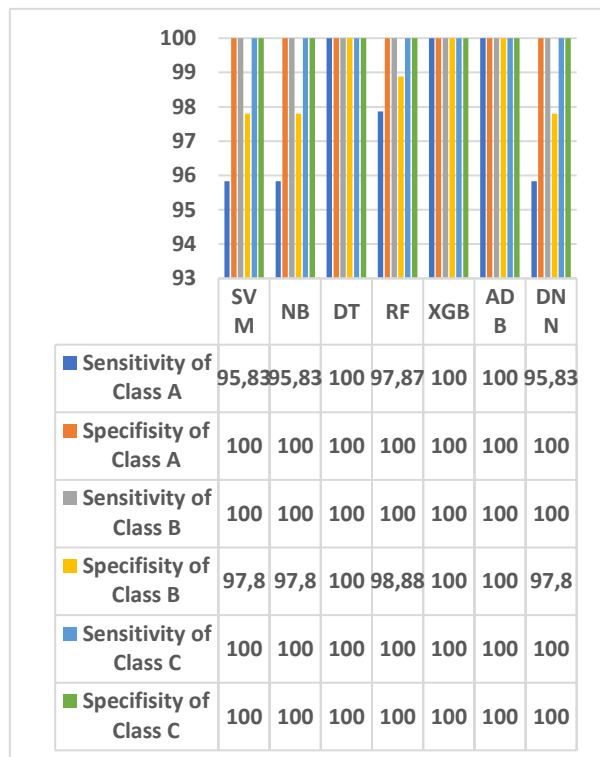


Figure 13: Sensitivity and specificity performance evaluation of the three classes.



Table 4: Comparing Proposed Methods with literature.

Reference	Number of Patients	Number of Features	Methods	Results (%)
[2]	48 of 8425 identified citations were included		Median Risk Bias	7 out of 27 (range, 6–22) points
[3]	313 individuals	15	LR SVM RF	The AUC is 0.79 for LR and RF, while SVM is 0.77
[4]	1,021 stroke survivors.	13-items	A latent growth curve model	The time following a stroke increases, survivors attend to continuously upgrade with interest to ADLs.
[5]	495, 1049 Patients	11	Penalized Linear Blended Models	RMSD for 3 BI points
[6]	498	7	6 Methods	
[7]	241	19	Two backward stepwise regression	
			Back Propagation Artificial Neural Network (BP-ANN)	Accuracy (38.33%) and cross-validated (48.51%)
			Learning Vector Quantization (LVQ)	Accuracy (50.00%) and cross validated (58.96%)
[8]	130	14	Self-Organizing Mapping (SOM)	Accuracy (53.33%) and cross validated (66.57%)
			Support Vector Machine (SVM)	Accuracy (53.33%) and cross validated (71.47%)
			Particle Swarm-Optimized SVM (PSO-SVM).	Accuracy (58.33%) and the highest cross-validated 74.29%
Proposed Method No Features Selection	313 with 776 Records	15	SVM	Accuracy 98.71
			NB	Accuracy 98.71
			DT	Accuracy 100
			RF	Accuracy 99.35
			XGB	Accuracy 100
			ADB	Accuracy 100
			DNN	Accuracy 98.71
Proposed Method with Features Selection	313 with 776 Records	15	SVM	Accuracy 98.71
			NB	Accuracy 98.71
			DT	Accuracy 100
			RF	Accuracy 100
			XGB	Accuracy 100
			ADB	Accuracy 100

## 5 Discussions

The present study aimed to investigate the influence of using many several ML algorithms for the classification of the BI score that represents the life living of patients after a stroke based on many features. Throughout the training step, all of the classifiers used were achieved an extraordinary scale of performances. While the testing results show that the used classifiers still show high accuracy. The results of this research based on the figures exhibited in the results showed that DT, XGB, and ADB classifiers achieved the most eminent performance reached 100% correctness in terms of accuracy, sensitivity, specificity, error, and AUC, for multi classify the digitized BI score, while the SVM and DNN classifiers are the worst. Moreover, using the features selection technique decreases the number of collected features in any future data collection from 15 to 5 only with a reduction ratio of 60% with the same results in classifiers except the RF enhanced from 98.71% to 100% in terms of accuracy. The results of feature selection mean that we can reduce the number of collected data and reducing the round time which makes patients more comfortable during data collection. Comparing the results of the proposed methods with other methods in the literature are shown in Table 4 The listed research studies in Table 4 have used the different datasets either collected by authors themselves or by others. It is noted that they used a different number of classes, patients, and features set. These factors can affect the performance of the used classification methods significantly. However, most of the listed methods in the literature have achieved accepted recognition methods have high classification rates compared to other methods, the system is tested for time consumption in intel core i5-6700 /3.4 GHz and 12 GB of RAM desktop computer using Python 3.9 on Spyder IDE.

## 6 Conclusion

In this research, a study on applying nine different machine learning algorithms for the prediction and multi-classification of Barthel index which represents the activities of daily living of post-stroke patients in clinical practice. The research focused on finding the best classifier(s) for diagnosing the dependency of life living of post-stroke patients. Also, we have provided a features reduction methodology using the Chi-squared test to reduce the number of features in the datasets and during the round where they were collected from the patients. Experimental results show that the proposed method achieves very high accuracy when the BI score of three classes is classified even in the full or reduced features dataset. Therefore, the proposed method may be used effectively in hospitals with a lower number of features collected from patients for predicting the status of the life living of patients after a stroke. By comparing the proposed method with other methods in the literature, the present method is proven to be more effective and can provide a powerful tool for automatic stroke patient evaluation using the mentioned features.

## Conflict of interest

The authors declare that they have no conflict of interest. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## References

- [1] B. R. Wittenauer and L. Smith, "Priority Medicines for Europe and the World " A Public Health Approach to Innovation " Update on 2004 Background Paper Written by Eduardo Sabaté and Sunil Wimalaratna Background Paper 6. 6 Ischaemic and Haemorrhagic Stroke," Who, no. December, 2012.
- [2] Veerbeek JM, Kwakkel G, van Wegen EE, Ket JC, Heymans MW. Early prediction of outcome of activities of daily living after stroke: a systematic review. *Stroke*. 42(5):1482-8,2011. <http://doi.org/10.1161/STROKEAHA.110.604090>
- [3] Lin WY, Chen CH, Tseng YJ, Tsai YT, Chang CY, Wang HY, Chen CK. Predicting post-stroke activities of daily living through a machine learning-based approach on initiating rehabilitation. *International journal of medical informatics*. 111(1):159-64, 2018. <http://doi.org/10.1016/j.ijmedinf.2018.01.002>.
- [4] Pai HC, Lai MY, Chen AC, Lin PS. Change in activities of daily living in the year following a stroke: a latent growth curve analysis. *Nursing research*. 67(4):286-93, 2018. <http://doi.org/10.1097/NNR.0000000000000280>.
- [5] Douiri A, Grace J, Sarker SJ, Tilling K, McKevitt C, Wolfe CD, Rudd AG. Patient-specific prediction of functional recovery after stroke. *International Journal of Stroke*. 12(5):539-48, 2017. <http://doi.org/10.1177/1747493017706241>.
- [6] Bertolin M, Van Patten R, Greif T, Fucetola R. Predicting cognitive functioning, activities of daily living, and participation 6 months after mild to moderate stroke. *Archives of Clinical Neuropsychology*. 33(5):562-76, 2018. <http://doi.org/10.1093/arclin/acx096>.
- [7] Gialanella B. Predicting outcome after stroke: the role of basic activities of daily living. *Eur J Phys Rehabil Med*. 49:629-37, 2013.
- [8] Lee JD, Chang TC, Yang ST, Huang CH, Hsieh FH, Wu CY. Prediction of quality of life after stroke rehabilitation. *Neuropsychiatry*. 6(6):369-75, 2016. <http://doi.org/10.4172/Neuropsychiatry.1000163>.
- [9] Jin X, Xu A, Bie R, Guo P. Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. In *International workshop on data mining for biomedical applications*, 9: 106-115, 2006. Springer, Berlin, Heidelberg. [http://doi.org/10.1007/11691730\\_11](http://doi.org/10.1007/11691730_11).
- [10] Alqudah AM. Ovarian cancer classification using serum proteomic profiling and wavelet features a comparison of machine learning and features

- selection algorithms. *Journal of Clinical Engineering*. 44(4):165-73, 2019.  
<http://doi.org/10.1097/JCE.0000000000000359>.
- [11] Ampomah EK, Nyame G, Qin Z, Addo PC, Gyamfi EO, Gyan M. Stock Market Prediction with Gaussian Naive Bayes Machine Learning Algorithm. *Informatica*. 15;45(2), 2021.  
<http://doi.org/10.31449/inf.v45i2.3407>
- [12] Tiwari P, Dao H, Nguyen GN. Performance evaluation of lazy, decision tree classifier and multilayer perceptron on traffic accident analysis. *Informatica*. 13;41(1), 2017.
- [13] Alqudah AM. Towards classifying non-segmented heart sound records using instantaneous frequency based features. *Journal of medical engineering & technology*. 3;43(7):418-30, 2019.  
<http://doi.org/10.1080/03091902.2019.1688408>.
- [14] Babajide Mustapha I, Saeed F. Bioactive molecule prediction using extreme gradient boosting. *Molecules*. 21(8):983, 2016.  
<http://doi.org/10.3390/molecules21080983>.
- [15] Sun Y, Liu Z, Todorovic S, Li J. Adaptive boosting for SAR automatic target recognition. *IEEE Transactions on Aerospace and Electronic Systems*. 7;43(1):112-25, 2007.  
<http://doi.org/10.1109/TAES.2007.357120>.
- [16] Alqudah AM, Alquran H, Qasmieh IA. Classification of heart sound short records using bispectrum analysis approach images and deep learning. *Network Modeling Analysis in Health Informatics and Bioinformatics*. 9(1):1-6, 2020.  
<http://doi.org/10.1007/s13721-020-00272-5>.
- [17] Alqudah AM, Alquraan H, Qasmieh IA. Segmented and non-segmented skin lesions classification using transfer learning and adaptive moment learning rate technique using pretrained convolutional neural network. In *Journal of Biomimetics, Biomaterials and Biomedical Engineering*, 42 :67-78, 2019. Trans Tech Publications Ltd.  
<http://doi.org/10.4028/www.scientific.net/JBBBE.42.67>.
- [18] Malkawi A, Al-Assi R, Salameh T, Alquran H, Alqudah AM. White blood cells classification using convolutional neural network hybrid system. In *2020 IEEE 5th middle east and Africa conference on biomedical engineering (MECBME) 27: 1-5, 2020. IEEE*.  
<http://doi.org/10.1109/MECBME47393.2020.9265154>.
- [19] Alqudah A, Alqudah AM, Qazan S. Lightweight Deep Learning for Malaria Parasite Detection Using Cell-Image of Blood Smear Images. *Rev. d'Intelligence Artif.*. 34(5):571-6, 2020.  
<http://doi.org/10.18280/ria.340506>
- [20] Alquran H, Alqudah AM, Abu-Qasmieh I, Al-Badarneh A, Almashaqbeh S. ECG classification using higher order spectral estimation and deep learning techniques. *Neural Network World*. 1;29(4):207-19, 2019.  
<http://doi.org/10.14311/NNW.2019.29.014>.



# Formal Verification of Pipelined Cryptographic Circuits: A Functional Approach

Abir Bitat and Salah Merniz

MISC Laboratory, NTIC faculty, Abdelhamid Mehri University, Constantine, Algeria

E-mail: abir.bitat@univ-constantine2.dz

**Keywords:** formal design, formal verification, cryptographic circuits, pipelined circuits, functional language, hardware description language

**Received:** May 28, 2020

*Cryptographic circuits are essential in systems where security is the main criteria. Therefore, it is crucial to ensure the correctness of not only the cryptographic algorithms, but also their hardware implementations. Formal methods, unlike the other validation techniques, guarantee the absence of errors. The problem is that designers still use conventional imperative Hardware Description Languages (HDLs), which are poorly suited for formal verification.*

*This paper presents an automatic verification methodology for the pipelined cryptographic circuits using formal methods. It consists of using the functional HDL Lava to describe and verify the equivalence between the behavioural specification and structural implementation of a circuit. To the best of our knowledge, we are the first to use this functional HDL for that purpose.*

*To show the features of the proposed approach, it was applied to verify the pipelined implementation of the cryptographic circuit AES (Advanced Encryption Standard).*

*Povzetek: Za namene preverjanja formalne pravilnosti delovanja vezij je opisan funkcionalni pristop.*

## 1 Introduction

Cryptography plays a major role in modern applications, as the present networks are trusted with highly sensitive information; hence, cryptographic circuits have become indispensable in these systems. To ensure the security of information, not only the cryptographic algorithms have to be verified but also their hardware implementations.

The first step of the design process consists in the conversion of the informal description of the design to a formal *Behavioural* (or *Algorithmic*) specification. From this latter, a *Structural* (or *Micro-architectural*) implementation is derived through refinement, followed by a sequence of design steps that reduce the abstraction levels until a realizable description is obtained. The verification of the circuit is carried out through all these stages; the most critical one is the *Functional verification*, which consists of confirming that the structural implementation provides the same behaviour mentioned in the behavioural specification. It is possible to verify the correctness of the cryptographic circuits, in quite a few ways, such as simulation, formal proof, and semi-formal verification in which formal techniques and simulation are strongly combined.

We focus on our work on functional verification, and we use formal methods to do so. In the following subsection, we present a summary of the related literature and their shortcomings to highlight the problematic; then we present the features of the proposed approach and the contribution of this paper. In section two, we explain the proposed verification methodology. In section three, we

demonstrate how we applied our approach for verifying the pipelined cryptographic circuit Advanced Encryption Standard (AES). And finally, conclusions are drawn in section four.

### 1.1 Related works

The vast majority of the existing literature related to the description and verification of pipelined cryptographic circuits are based on using imperative HDLs such as VHDL and Verilog, [1], [2], [3], [4], [5]. These languages are made for description, simulation, and synthesis of hardware; however, they are poorly suited for formal verification, because of their lack of formal semantics; besides, they do not allow descriptions of the highest design levels [6],[7]. In order to be able to use formal methods for verification, we need formal descriptions that we can reason about; however, imperative HDLs provide descriptions that are hard to express in any formal logic; [8],[6],[9],[10],[11], which requires either a translation of those descriptions to some formal logic or rewriting new "equivalent" descriptions, which eliminates the need for the imperative ones in the first place, because there is no relation between the two [6],[12],[13],[14]. This makes the formal verification of devices expressed in imperative HDLs a quite hard process.

Consequently, simulation is the technique that has been used in most of these approaches [2], [3], [4]. The problem with simulation is that it can not be sufficient as a verification technique for systems as critical as the cryptographic circuits. Formal verification can still be done but with

quite a few challenges when using imperative HDLs; therefore, the abstract behavioural specification needs powerful mechanisms of structuring and translating [5]. Mostly, deductive methods are applied for verification of such complex circuits, like the case of [1]; this kind of methods is quite difficult because it usually requires user interference through all the verification process in almost a manual way.

Some algebraic approaches [15], [16] used formal methods for the verification of these circuits; similar to the work presented in this paper, the hierarchy technique was used to reduce the complexity of design and therefore simplify the verification task. However, many details differ in our approach from this work, in particular, the verification time is significantly less than the one presented in [15], which took 13 minutes to formally verify the same 128bit AES circuit. Another approach that was slightly faster than the work mentioned before, with a difference of 5 minutes, is presented in [17]. It consists of a language that supports automated verification of cryptographic assembly code.

Several functional approaches were applied for the formal verification of hardware designs; however, to the best of our knowledge, the only one beside ours that was applied to implement and verify the pipelined cryptographic circuits is the work presented in [18]. This approach uses the functional language only to describe the behavioural specification; but not for the structural implementation; which makes translation difficulties reappear.

Another formal approach was proposed in [19]; it uses the equivalence checking technique. The specification is described at the RTL level using VHDL, and the verification process of some pipelined implementations of the KASUMI cipher took from 3 to 9 minutes, depending on the number of stages. Another work that targeted pipelined implementations is [20]; it uses VHDL to describe private and public key crypto-processor; the verification was done using simulation, formal verification, and static timing analysis.

## 1.2 Contribution of this work

- The most important feature of our approach, is that it consists of using a functional HDL, which is very suitable for hardware description and for formal verification as well. A comparison study of these HDLs against other types showed that they give the best results [21].
- Secondly, our approach performs the functional verification using formal methods, as this latter consists of proving mathematically the correctness of a design, which is crucial for security systems.
- In addition, our approach uses two techniques: which are hierarchy and modularity, in order to reduce the complexity of the design, which makes the verification much easier.
- Unlike the other conventional HDLs, it is possible

to represent the most abstract descriptions with functional HDLs.

- Descriptions of functional languages and HDLs can be executed, which allow the verification through the simulation technique as well as the formal verification.
- The functional HDL used in our approach has some built-in tools that allow automatic formal verification of circuits.
- The proposed approach presents a verification methodology that is easier and faster than the previous related works.
- Lastly, even though our approach was proposed for pipelined cryptographic circuits, it is not exclusive to them.

## 2 The proposed approach

The proposed approach consists of a formal design and verification methodology for the pipelined cryptographic circuits using a functional HDL. This choice is motivated by the interesting characteristics of these HDLs; such as the composition of functions in the same way that complex circuits are developed, using function composition, renaming, and abstraction; the other major advantage of introducing the functional style to hardware design is having much more concise descriptions, and the ability to provide reusable functions that are abstractions of common patterns. Moreover, functional languages usually have an extremely expressive type system, which allows being more strict on defining the limitations on values. This makes finding errors and violations easier. Several functional HDLs have been created over the decades; their high diversity is due to the complexity of hardware design. A historical survey that discusses these languages can be found in [22]. The language that we chose for our approach is Lava [23], which consists of a simple HDL embedded in the functional programming language Haskell. To the best of our knowledge, no functional HDLs (including Lava) has been used before for both description and formal verification of the cryptographic circuits.

The design flow of hardware devices is depicted in Fig. 1. It starts with an algorithmic description which will be considered as the initial specification of the design; then other descriptions are derived from it. Each implementation resulting in a certain abstraction level will be used as the specification for the next one.

Since the algorithmic level is the most abstract, the architectural details do not appear at it; thus, both sequential and pipelined architectures have the same algorithmic description. Accordingly, we use the same principle of the design flow to verify a pipelined implementation of a cryptographic circuit; we start first by verifying the correctness of a sequential structural implementation against its

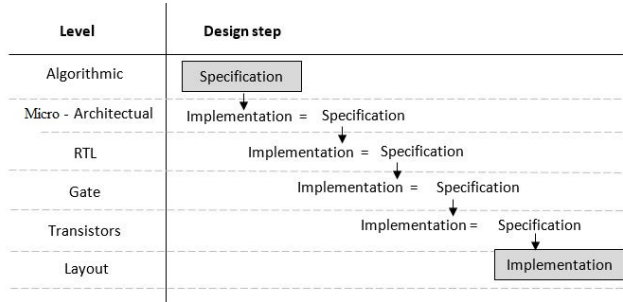


Figure 1: The design and verification flow of a hardware device.

behavioural specification; once it is verified, we check the equivalence of the pipelined structural implementation to it, as demonstrated in Fig. 2.

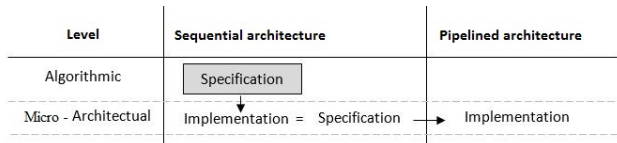


Figure 2: A verification approach of a pipelined implementation of a circuit.

The behavioural specification at the algorithmic level deals with a different type than the structural implementations at the micro-architectural level; thus, we need a mapping function between the two descriptions, as shown in Fig. 3.

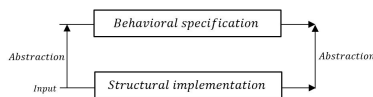


Figure 3: The abstraction function needed between the behavioural specification and structural implementation of a circuit.

The shared behavioural description is referred to by *Spec*; the sequential architecture by *imp1*, and the pipelined one by *imp2*. So, the correctness rule of this latter would be described by the following theorem:

$$\forall x, spec (abs x) = abs (imp2 x) \quad (1)$$

Theorem (1) has to be decomposed to be proven; thus, we must prove the following couple of theorems:

$$\forall x, spec (abs x) = abs (imp1 x) \quad (2)$$

$$\wedge \quad \forall x, imp1 x = imp2 x \quad (3)$$

The behavioural specification and both structural implementations are described as functions, the former using the

functional language Haskell, and the latter using the functional HDL Lava; which consists of Haskell modules that give the user various facilities to work on circuits.

For formal verification; we use one of Lava tools, which is a SAT solver, that verifies automatically the equivalence between descriptions. Fig. 4 shows an approach of performing equivalence checking using SAT solvers. If both descriptions are equivalents, the output of the XOR gate should be always *False*; if it becomes *True* for any input, it means that the two descriptions are producing different outputs for the same input, which negates their equivalence.

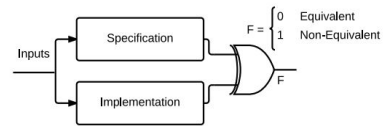


Figure 4: Performing equivalence checking using SAT solvers [24].

We must import three modules to be able to shift from the general-purpose language Haskell to the HDL Lava: *Lava* [25], which defines several operations that we can use to build circuits; *Patterns*, to access wiring circuits and connection patterns; and *Arithmetic*, to access the arithmetical circuits:

```
import Lava
import Lava.Patterns
import Lava.Arithmetic
```

Circuits in Lava are described by functions, and their inputs and outputs can either be of type *Signal Bool* which may take one of two values *low* or *high*; or *Signal Int*.

To verify that a circuit's structural implementation meets its behavioural specification, we must define a safety property that expresses their equivalence. Thus, for the verification of the pipelined implementation, we need two safety properties corresponding to the two theorems (2) and (3) mentioned before. These properties are also defined in Lava by functions in the following forms:

```
propertyEquiv1 in = ok
where
    out1 = spec (abs in)
    out2 = abs (imp1 in)
    ok = out1 <==> out2
```

```
propertyEquiv2 in = ok
where
    out1 = imp1 in
    out2 = imp2 in
    ok = out1 <==> out2
```

To verify these properties, we use the Lava function *satzoo*, which is a call to the satisfiability solver:

```
satzoo propertyEquiv1
satzoo propertyEquiv2
```

This operation generates a logical formula that expresses the equivalence property; this formula is then sent to an external theorem prover, which will prove (or disprove) its validity, and the result is taken back into Lava.

The input *in* must be of a finite form; this is not possible in cryptographic circuits, where both data and key are of an important size that can only be represented by lists. But since the inputs of block cipher cryptographic circuits are of a fixed size, we will only verify the properties for that size. Thus, we define new equivalence properties, which are explicit about what size of input we want to verify them.

```
propertyEquiv1ForSize n =
  forAll (list n) $ \ in →
    propertyEquiv1 in
```

```
propertyEquiv2ForSize n =
  forAll (list n) $ \ in →
    propertyEquiv2 in
```

Then we call the function *satzoo* for both properties with a specific size *n*:

```
satzoo (propertyEquiv1ForSize n)
satzoo (propertyEquiv2ForSize n)
```

This operation will verify that both descriptions give the same output, for all inputs of that size. When it finds that the output of one description always equals the other, it returns *Valid*. The SAT-solver actually checks that the negation of the formula is unsatisfiable, leading to the *Valid* answer inside Lava [25]. Our proposed approach explained above is depicted in Fig. 5.

### 3 Application and results

In this section, we demonstrate how we applied the proposed approach to verify the AES sequential circuit, illustrated in Fig. 6; and the pipelined circuit, illustrated in Fig. 7. AES [26] is a symmetric block cipher, constructed based on the Rijndael system. The plain and ciphertexts are taken as blocks of 128 bits. The key, on the other hand, varies depending on the system version, between 128, 192, and 256 bits. We only focus here on the 128-bit key size AES.

The encryption consists of ten identical rounds of processing. Except for the last one, each round includes four steps; the order in which these steps are executed is different in encryption from decryption. The 128-bit input

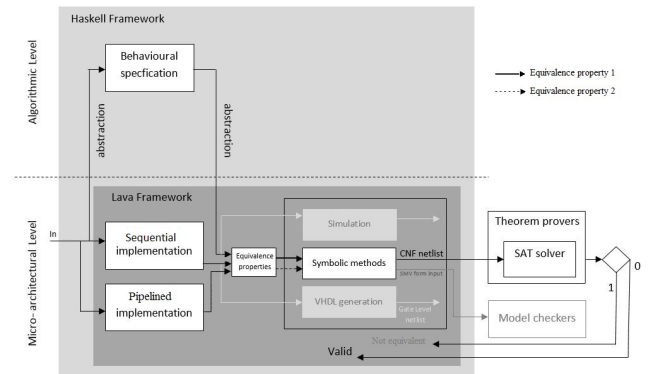


Figure 5: The proposed verification methodology of pipelined cryptographic circuits in a functional framework.

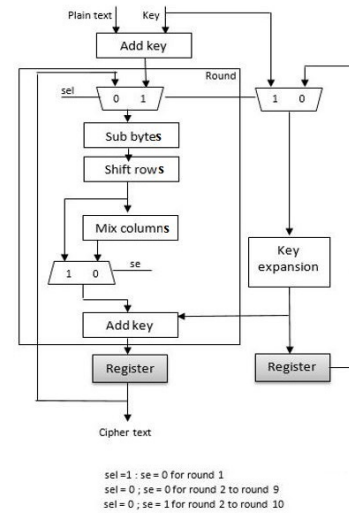


Figure 6: Sequential architecture of AES128 circuit.

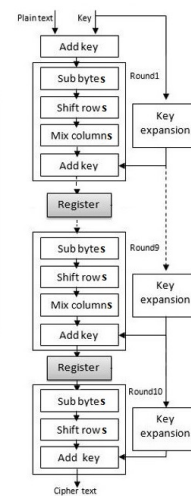


Figure 7: Pipelined architecture of AES128 circuit.



block is organized in a four by four-byte matrix, column by column. The matrix will be added to a sub-key using the *Xor* operation and the result obtained will be transmitted as input for the first round. At each round, the following operations are performed:

- The *subBytes* function is performed using S-boxes. These boxes in AES are based on a mathematical model, which is the modular arithmetic using polynomials.
- The *shiftRows* function is a simple-circular-left shift of bytes: the first row doesn't change; the second is shifted by one position; the third is shifted by two positions; and the last one, is shifted by three positions.
- The *mixColumns* function consists on taking each column of the matrix and multiplying it by a fix matrix, then reducing the answers modulo the polynomial  $x^8 + x^4 + x^3 + x^1 + x^0$ .
- The *addKey* function adds the round sub-key to the block. The sub-key used here is calculated by the *keyExpansion* function.

The application of these four operations takes place at each round except for the last one, where the *mixColumns* function is not applied. The *KeyExpansion* is a basic operation in the encryption process; it uses the cipher key to produce sub-keys in the same number of rounds. The first sub-key is just the original cipher key; then, to get the next sub-key, the previous one is passed through a function that involves a rotation P-box, a set of identical S-boxes, and addition modulo 2 to a round-constant.

### 3.1 Behavioural specification of the AES circuit

The function *aesSpec* represents the behavioural description of AES; it takes one input (*m*: plain text, *k*: cipher key); and one output (*c*: cipher text).

$$\begin{aligned} \text{aesSpec } (m, k) &= c \\ \text{where} \\ \text{subKeys} &= \text{keyExpSpec } 1 \ k \\ n &= \text{addKeySpec } (m, (\text{subkeys } !! 0)) \\ c &= \text{roundsSpec } 1 \ (n, \text{subKeys}) \end{aligned}$$

The function *roundsSpec* is recursive and defined in the following way:

$$\begin{aligned} \text{roundsSpec } n \ (m1, \text{subKeys}) &= m2 \\ \text{where} \\ n1 &= \text{subBytesSpec } m1 \\ n2 &= \text{shiftRowsSpec } n1 \\ n3 &= \text{mixColumnsSpec } n2 \\ n4 &= \text{addKeySpec } (n3, (\text{subKeys } !! (n - 1))) \\ m2 &= \text{roundsSpec } (n + 1) \ (n4, \text{subKeys}) \end{aligned}$$

$$\begin{aligned} \text{roundsSpec } 10 \ (m9, \text{subKeys}) &= m10 \\ \text{where} \\ n1 &= \text{subBytesSpec } m9 \\ n2 &= \text{shiftRowsSpec } n1 \\ m10 &= \text{addKeySpec } (n2, (\text{subKeys } !! 9)) \end{aligned}$$

The *keyExpSpec* function is recursive as well, and is defined as follows:

$$\begin{aligned} \text{keyExpSpec } n \ ki &= kj : (\text{keyExpSpec } (n + 1) \ kj) \\ \text{where} \\ w0 &= \text{wXorSpec } (sBox(\text{shiftSpec } (ki!!3)), \\ &\quad ki!!0, \text{rconst}!!(n - 1)) \\ w1 &= \text{wXorSpec } (ki!!1, w0) \\ w2 &= \text{wXorSpec } (ki!!2, w1) \\ w3 &= \text{wXorSpec } (ki!!3, w2) \\ kj &= [w0, w1, w2, w3] \end{aligned}$$

$$\begin{aligned} \text{keyExpSpec } 10 \ k9 &= k10 : [] \\ \text{where} \\ w0 &= \text{wXorSpec } (sBox(\text{shiftSpec } (k9!!3)), \\ &\quad k9!!0, \text{rconst}!!9) \\ w1 &= \text{wXorSpec } (k9!!1, w0) \\ w2 &= \text{wXorSpec } (k9!!2, w1) \\ w3 &= \text{wXorSpec } (k9!!3, w2) \\ k10 &= [w0, w1, w2, w3] \end{aligned}$$

## 3.2 Structural implementation of the AES circuit

### 3.2.1 AES sequential architecture

The function *aesImpl* represents the structural description of AES sequential architecture. It takes (*m, k*) as input, and outputs *c*. The definition of this function is quite similar to *aesSpec*, with the difference in their signature, and their internal functions, because they work with two different types. The *aesSpec* function works with data as hexadecimal, which are taken in Lava as *Signal Int*. However, *aesImpl* works with bits, which are represented by the type *Signal Bool*.

$$\begin{aligned} \text{aesSpec} &:: ([\text{Signal Int}], [\text{Signal Int}]) \rightarrow [\text{Signal Int}] \\ \text{aesImpl} &:: ([\text{Signal Bool}], [\text{Signal Bool}]) \rightarrow [\text{Signal Bool}] \end{aligned}$$

The function *aesImpl* is defined as it follows:

$$\begin{aligned} \text{aesImpl } (m, k) &= c \\ \text{where} \\ \text{subKeys} &= \text{keyExpImpl } 1 \ k \\ n &= \text{addKeyImpl } (m, (\text{subkeys } !! 0)) \\ c &= \text{roundsImpl } 1 \ (n, \text{subKeys}) \end{aligned}$$

The functions that *aesImp1* calls represent the inner components of the AES circuit; they differ from their corresponding functions in *aesSpec*, even though they use the same hierarchical way of description.

### 3.2.2 AES pipelined architecture

The function *aesImp2* represents the structural description of a pipelined architecture of AES. Its definition is different than *aesImp1*; to allow the hardware parallelism, we need multiple functional components, one for each round; instead of using the same one for all of them; this will allow us to encrypt multiple blocks at the same time. For instance, when the first data block is on the second round of encryption, another block can start its first round; therefore when the first block is at its last round, nine other blocks can be calculated simultaneously. *aesImp2* has the same signature as *aesImp1* and it is defined in the following way:

$$aesImp2 :: ([Signal Bool], [Signal Bool]) \rightarrow [Signal Bool]$$

$$\begin{aligned}
 & aesImp2(m, k) = c \\
 & \text{where} \\
 & \quad s0 = addKeyImp2(m, k) \\
 & \quad k1 = getK 1 k \\
 & \quad s1 = roundImp2(s0, k1) \\
 & \quad k2 = getK 2 k1 \\
 & \quad s2 = roundImp2(s1, k2) \\
 & \quad k3 = getK 3 k2 \\
 & \quad s3 = roundImp2(s2, k3) \\
 & \quad k4 = getK 4 k3 \\
 & \quad s4 = roundImp2(s3, k4) \\
 & \quad k5 = getK 5 k4 \\
 & \quad s5 = roundImp2(s4, k5) \\
 & \quad k6 = getK 6 k5 \\
 & \quad s6 = roundImp2(s5, k6) \\
 & \quad k7 = getK 7 k6 \\
 & \quad s7 = roundImp2(s6, k7) \\
 & \quad k8 = getK 8 k7 \\
 & \quad s8 = roundImp2(s7, k8) \\
 & \quad k9 = getK 9 k8 \\
 & \quad s9 = roundImp2(s8, k9) \\
 & \quad k10 = getK 10 k9 \\
 & \quad s10 = round10Imp2(s9, k10) \\
 & \quad c = s10
 \end{aligned}$$

The functions *roundImp2* is recursive as well, and it is

defined in the following way:

$$\begin{aligned}
 & roundImp2(s, k) = nextS \\
 & \text{where} \\
 & \quad s1 = subBytesImp2 s \\
 & \quad s2 = shiftRowsImp2 s1 \\
 & \quad s3 = mixColumnsImp2 s2 \\
 & \quad nextS = addKeyImp2(s3, k)
 \end{aligned}$$

$$\begin{aligned}
 & round10Imp2(s9, k10) = s10 \\
 & \text{where} \\
 & \quad s1 = subBytesImp2 s9 \\
 & \quad s2 = shiftRowsImp2 s1 \\
 & \quad s10 = addKeyImp2(s2, k10)
 \end{aligned}$$

The function *getK* is the corresponding of *keyExpImp1* on the sequential implementation, it calculates only one key, and it is defined in the following way:

$$\begin{aligned}
 & getK n ki = kj \\
 & \text{where} \\
 & \quad w0 = wXorImp2(sBox(shiftImp2(ki !! 3)), ki !! 0, rconst !! (n - 1)) \\
 & \quad w1 = wXorImp2(ki !! 1, w0) \\
 & \quad w2 = wXorImp2(ki !! 2, w1) \\
 & \quad w3 = wXorImp2(ki !! 3, w2) \\
 & \quad kj = [w0, w1, w2, w3]
 \end{aligned}$$

### 3.3 Formal verification of the AES circuit

To verify the correctness of the AES pipelined circuit, we need to prove theorem (4) that expresses the equivalence between its behavioural specification and sequential implementation; and then theorem (5) that expresses the equivalence between the sequential implementation and the pipelined one.

$$\forall m, k, m \in [SignalBool], k \in [SignalBool],$$

$$aesSpec(abs1(m, k)) = abs2(aesImp1(m, k)) \quad (4)$$

$$\forall m, k, m \in [SignalBool], k \in [SignalBool],$$

$$aesImp1(m, k) = aesImp2(m, k) \quad (5)$$

The equivalence properties are described by functions with one input  $(m, k)$  of type *Signal Bool*, which will be passed to both descriptions, in order to verify that they always produce the same output. As a result, an abstraction function called *fromSbToSi* is introduced, it converts from the type *Signal Bool* to *Signal Int*. Both functions *abs1* and

*abs2* are defined using *fromSbToSi*.

```

propertyEquivSeqAES in = ok
  where
    out1 = aesSpec (abs1 in)
    out2 = abs2 (aesImp1 in)
    ok = out1 <==> out2

propertyEquivPipAES in = ok
  where
    out1 = aesImp1 in
    out2 = aesImp2 in
    ok = out1 <==> out2
    
```

Since we use the infinite structure *list*, we also need to define equivalence properties that are explicit about the size of inputs:

```

propEquivSeqAES_forSize n =
  forAll (list n) $ \ m →
    forAll (list n) $ \ k →
      propertyEquivSeqAES (m, k)
propEquivPipAES_forSize n =
  forAll (list n) $ \ m →
    forAll (list n) $ \ k →
      propertyEquivPipAES (m, k)
    
```

To verify the AES pipelined circuit, we call the *satzoo* function for both implementations:

```

verificationSeqAES =
  satzoo (propEquivSeqAES_forSize 128)
verificationPipAES =
  satzoo (propEquivPipAES_forSize 128)
    
```

The *satzoo* function generates an output of the type *IO proofResult*. The execution of this function outputs the value **Valid** for both properties, which means that the sequential implementation *aesImp1* gives the same output as the behavioural specification *aesSpec*, and the pipelined implementation *aesImp2* gives the same output as the sequential one *aesImp1*, for every possible combination of plain test and key of 128 bits size. Thus, we conclude that the pipelined implementation *aesImp2* is equivalent to its behavioural specification *aesSpec*.

A comparison between the proposed approach and all the similar works mentioned here is summarized in Table 1. As we can see the majority of the previous works are based on using imperative HDLs [1],[2],[3],[4],[5],[17],[19],[20]. Unlike functional HDLs, the imperative ones used in these approaches do not permit abstract descriptions of the high levels of design, which means that their descriptions are more detailed and therefore require longer code-lines; which makes them of higher complexity. Besides, finding

Work	Approach	Method	Circuit	Time (s)
[1]	Imperative	Formal methods	SHA1	/
[2]	Imperative	Simulation	TDES	/
[3]	Imperative	Simulation	AES	/
[4]	Imperative	Simulation	Kasumi	180
[5]	Imperative	Formal methods	DES	59
[15]	Algebraic	Formal methods	AES	800
[16]	Algebraic	Formal methods	AES	844
[17]	Imperative	Formal methods	SHA AES	2100
[18]	Functional	Formal methods	AES	/
[19]	Imperative	Formal methods	Kasumi	180
[20]	Imperative	Formal methods	AES	/
Ours	Functional	Formal methods	AES	2.23

Table 1: Comparative table to the similar works’ verification methods and time.

errors and correcting them becomes a harder and more tedious process compared to functional HDLs descriptions. Although there is no actual comparison of the different approaches’ code-lines (due to the lack of data), we can say with no hesitation that the functional HDLs provide more concise descriptions than the imperative ones for the same circuit. [2],[3],[4] use the simulation technique for the functional verification, which is not sufficient for critical systems such as the cryptographic circuits, even if they provide fast and automatic verification. As we can see, it is still possible to use formal verification with imperative HDLs, but as we established since they lack formal semantics, this makes the verification process very hard, as it requires translation of descriptions into a formal logic to be able to reason about them. [2] used deductive methods, which means that the verification process was not automatic. In the proposed approach we were able to verify the AES sequential data-path automatically in 1.18s, and the AES pipelined one in 1.05s, which makes the total 2.23s, which is faster than all of the other works that we know of their verification time [5], [15],[16],[17],[19].

Since the behavioural specification and both structural implementations are specified by functions, they are executable; therefore, we can simulate them and examine the results. This is interesting because it allows us to verify that not only they are equivalents to each other, but that in fact, they give the expected results. All three functions (*aesSpec*,*aesImp1*,*aesImp2*) were simulated and they give the expected output of encryption.

To verify the other versions of AES, with the appropriate changes in the behavioural and structural descriptions, the equivalence properties need to be explicit about two different sizes.

$$\begin{aligned} \text{propEquivSeqAES\_forSizes } n \ l = & \\ & \text{forAll (list } n) \$ \setminus m \rightarrow \\ & \quad \text{forAll (list } l) \$ \setminus k \rightarrow \\ & \quad \quad \text{propertyEquivSeqAES } (m, k) \\ \text{propEquivPipAES\_forSizes } n \ l = & \\ & \text{forAll (list } n) \$ \setminus m \rightarrow \\ & \quad \text{forAll (list } l) \$ \setminus k \rightarrow \\ & \quad \quad \text{propertyEquivPipAES } (m, k) \end{aligned}$$

For instance, to verify the AES circuit of 192-bit key size, we call the *satzo* function in the following way:

$$\begin{aligned} \text{verificationSeqAES192} = & \\ & \text{satzo}(\text{propEquivSeqAES\_forSizes } 128 \ 192) \\ \text{verificationPipAES192} = & \\ & \text{satzo}(\text{propEquivPipAES\_forSizes } 128 \ 192) \end{aligned}$$

When automatic verification is not possible, we can use the bottom-up proof method proposed in [27], where they started by verifying the functions (components) at the lowest level of the hierarchy, once they are verified, they replaced the implementation with its equivalent specification at the upper level, and so on until verifying the whole circuit.

## 4 Conclusion

In this paper, we presented an automatic formal verification methodology for the pipelined cryptographic circuits. It is the first application of the functional HDLs for the design and verification of such complex circuits. The proposed approach was demonstrated and applied to the 128-bit AES pipelined circuit. As prospects, we aim to verify the super-scalar designs as well, on which the adopted scalable methodology should be able to prove.

## References

- [1] Toma, D. (2006) *Vérification Formelle des systèmes numériques par démonstration de théorèmes: application aux composants cryptographiques* (Doctoral dissertation).
- [2] Singh, Kirat Pal, and Shivani Parmar. (2015) "Design of high performance MIPS cryptography processor based on T-DES algorithm."
- [3] Ali, Imran, Gulistan Raja, and Ahmad Khalil Khan. (2014) "A 16-Bit Architecture of Advanced Encryption Standard for Embedded Applications." *12th International Conference on Frontiers of Information Technology*. IEEE, Pakistan, pp 220-225. <https://www.doi.org/10.1109/FIT.2014.49>
- [4] Lam, Chiu Hong. (2009) *Verification of pipelined ciphers*. MS thesis. University of Waterloo.
- [5] Clarke, E., Kroening, D. (2003) "Hardware verification using ANSI-C programs as a reference". *The ASP-DAC Asia and South Pacific Design Automation Conference*, IEEE, Japan, pp. 308-311. <https://www.doi.org/10.1109/ASPDAC.2003.1195033>
- [6] Camilleri, A., Gordon, M., Melham, T. (1986). *Hardware verification using higher-order logic* University of Cambridge, Computer Laboratory.No. UCAM-CL-TR-91.
- [7] Damaj, I. W. (2007). *Parallel algorithms development for programmable devices with application from cryptography*. *International Journal of Parallel Programming*, 35(6), 529-572. <https://doi.org/10.1007/s10766-007-0046-1>
- [8] Salah, M. (2008) *Méthodologie de Vérification Formelle Pour les Microarchitectures RISC: Approche Fonctionnelle* (Doctoral dissertation).
- [9] Walker, R. A., Camposano, R. (2012). *A survey of high-level synthesis systems*. Springer Science and Business Media. Vol. 135.
- [10] Seger, C. J. (1992). *An introduction to formal hardware verification*. University of British Columbia, Department of Computer Science.
- [11] Salem, A. M. E. F. (1992). *Vérification formelle des circuits digitaux décrits en VHDL* (Doctoral dissertation, Université Joseph-Fourier-Grenoble I).
- [12] Guo, X., Dutta, R. G., Jin, Y., Farahmandi, F., Mishra, P. (2015). Pre-silicon security verification and validation: A formal perspective. *In Proceedings of the 52nd Annual Design Automation Conference* ACM. United States. (pp. 1-6). <https://doi.org/10.1145/2744769.2747939>
- [13] Araiza-Illan, D., Eder, K. Richards, A. (2014). Formal verification of control systems' properties with theorem proving. *UKACC International Conference on Control (CONTROL)*. United Kingdom. IEEE. (pp. 244-249). <https://doi.org/10.1109/CONTROL.2014.6915147>
- [14] Singh, K. P., Parmar, S. (2015). *Design of high performance MIPS cryptography processor based on T-DES algorithm*. arXiv preprint arXiv:1503.03166.

- [15] Homma, Naofumi, Kazuya Saito, and Takafumi Aoki. (2011) "A Formal Approach to Designing Cryptographic Processors Based on  $GF(2^m)$  Arithmetic Circuits." *IEEE Transactions on Information Forensics and Security* vol. 7, no 1, p. 3-13.  
<https://doi.org/10.1109/TIFS.2011.2157687>
- [16] Homma, Naofumi, Kazuya Saito, and Takafumi Aoki. (2013) "Toward formal design of practical cryptographic hardware based on Galois field arithmetic." *IEEE Transactions on Computers*. vol. 63, no 10, p. 2604-2613.  
<https://doi.org/10.1109/TC.2013.131>
- [17] Bond, Barry, et al. (2017) "Vale: Verifying high-performance cryptographic assembly code." *26th USENIX Security Symposium (USENIX Security 17)*. USENIX, Canada, p. 917-934.
- [18] Lewis, Jeff. (2007) "Cryptol: specification, implementation and verification of high-grade cryptographic applications." *The 2007 ACM workshop on Formal methods in security engineering.*, ACM, United States, p. 41-41.  
<https://doi.org/10.1145/1314436.1314442>
- [19] Lam, Chiu Hong, and Mark D. Aagaard. (2007) "Formal Verification of a Pipelined Cryptographic Circuit Using Equivalence Checking and Completion Functions." *2007 Canadian Conference on Electrical and Computer Engineering*. IEEE, Canada, p. 1401-1404.  
<https://doi.org/10.1109/CCECE.2007.352>
- [20] Kim, Ho Won, and Sunggu Lee. (2004) "Design and implementation of a private and public key crypto processor and its application to a security system." *IEEE Transactions on Consumer Electronics*.vol. 50, no 1, p. 214-224.  
<https://doi.org/10.1109/TCE.2004.1277865>
- [21] Wolfs, Davy, et al. (2011) "Design automation for cryptographic hardware using functional languages." *Proceedings of the 32nd WIC Symposium on Information Theory in the Benelux. Werkgemeenschap voor Informatie-en Communicatietheorie.*; Netherlands, p. 194-201.
- [22] Chen, Gang. (2012) "A short historical survey of functional hardware languages." *ISRN Electronics* vol 2012.  
<https://doi.org/10.5402/2012/271836>
- [23] Bjesse, Per, et al. (1998) "Lava: hardware design in Haskell." *ACM SIGPLAN Notices* vol. 34, no 1, p. 174-184.  
<https://doi.org/10.1145/291251.289440>
- [24] Guo, Xiaolong, et al. (2015) "Pre-silicon security verification and validation: A formal perspective." *The 52nd Annual Design Automation Conference.*, Association for Computing Machinery United states, p. 1-6.  
<https://doi.org/10.1145/2744769.2747939>
- [25] Claessen, Koen, and Mary Sheeran. (2007) *A slightly revised tutorial on lava: A hardware description and verification system.*
- [26] Daemen, Joan, and Vincent Rijmen. (2013) "The design of Rijndael: AES-the advanced encryption standard". *Springer Science and Business Media.*
- [27] Abir, Bitat., and merniz. Salah. (2018) "Towards formal verification of cryptographic circuits: A functional approach." *The 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*. IEEE, Algeria, p. 1-6.  
<https://doi.org/10.1109/PAIS.2018.8598527>



# Skeleton-aware Multi-scale Heatmap Regression for 2D Hand Pose Estimation

Ikram Kourbane and Yakup Genc

Department of Computer Engineering, Faculty of Engineering, Gebze Technical University, Kocaeli, Turkey

E-mail: ikourbane@gtu.edu.tr; yakup.genc@gtu.edu.tr

**Keywords:** hand pose estimation, hand detection, hand dataset, convolutional neural networks, heatmaps

**Received:** March 14, 2021

*Hand pose estimation plays an essential role in sign language understanding and human-computer interaction. Existing RGB-based 2D hand pose estimation methods learn the joint locations from a single resolution, which is not suitable for different hand sizes. To tackle this problem, we propose a new deep learning-based framework that consists of two main modules. The first one presents a segmentation-based approach to detect the hand skeleton and localize the hand bounding box. The second module regresses the 2D joint locations through a multi-scale heatmap regression approach that exploits the predicted hand skeleton as a constraint to guide the model. Moreover, we construct a new dataset that is suitable for both hand detection and pose estimation tasks. It includes the hand bounding boxes, the 2D keypoints, the 3D poses and their corresponding RGB images. We conduct extensive experiments on two datasets to validate our method. Qualitative and quantitative results demonstrate that the proposed method outperforms the state-of-the-art and recovers the pose even in cluttered images and complex poses.*

*Povzetek: V prispevku je predstavljena učna metoda za nalogo 2D ocenjevanja položaja roke z uporabo monokularne RGB kamere.*

## 1 Introduction

The hands are one of the most important and intuitive interaction tools for humans. Solving the hand pose estimation problem is crucial for many applications, including human-computer interaction, virtual reality, augmented reality and sign language recognition.

The earlier works in hand tracking use special hardware to track the hand, such as gloves and visual markers [1]. But, these types of solutions are expensive and restrict the applications to limited scenarios. Tracking hands without any device or markers is desirable. To this end, several works have been proposed in the literature to tackle this problem [2]. However, markerless hand pose estimation is very challenging due to strong articulations and self-occlusions. Furthermore, the hands have a huge variation in shape, size, skin texture and color.

The rapid development of deep learning techniques revolutionizes complex computer vision problems [3, 4] and outperforms conventional methods in many challenging tasks, including object classification [5], object segmentation [6, 7] and object detection [8, 9]. Hand pose estimation is not an exception and deep convolution neural networks (CNNs) [10] have been applied successfully in [11, 12, 13]. These studies address the scenarios where the hand is tracked via an RGB-D camera. However, depth-enhanced data is not available everywhere, and they need an overhead setup to utilize. Thus, estimating the hand pose from a single RGB image has been an active and challenging area of research, as they are cheaper and easier to use than depth sensors [14, 15, 16, 17].

We can classify RGB-based hand pose estimation methods into two broad categories as regression-based and detection-based. The former approach uses CNNs as an automatic feature extractor to directly estimate the joint locations [14, 18, 19]. Although the regression-based approach is fast at inference time, it remains a difficult optimization problem due to its non-linear nature requiring many iterations and a lot of data for convergence [20].

To overcome these limitations, recent solutions to human and hand pose estimation problems use probability density maps such as the heatmap [16, 21, 22]. They divide the pose estimation problem into two steps. The first one finds a dense pixel-wise prediction for each joint while the second step infers the joint locations by finding the maximum pixel in each heatmap. The heatmap representation helps the neural network to estimate the joint locations robustly and has a fast convergence property.

In this work, we focus on the 2D hand pose estimation from a single RGB image. This task is also challenging due to the many degrees of freedom (DOF) and the self-similarity of the hand. The proposed approach has two principal components; The first one estimates the hand skeleton using the UNet-based architecture [23]. The hand bounding boxes are extracted in a post-processing step from the predicted skeleton. The second part presents a new multi-scale heatmap regression approach to estimate joint locations from multiple resolutions. Specifically, the network output is supervised on different scales to ensure accurate poses for different hand image sizes. This strategy helps the model for better learning of the contextual and the location information. Besides, our method uses the

predicted hand skeleton as additional information to guide the network to predict the 2D hand pose.

We validate the proposed method on a common existing Large-Scale Multi-View hand pose dataset (LSMV) [18]. Furthermore, we create a new dataset suitable for hand detection and 2D pose estimation tasks using leap motion sensors. This dataset includes 60 thousand samples, such that each one contains the hand bounding box, the 2D keypoint, 3D pose and the corresponding RGB image. We extended our experiments to our newly created dataset (GTHD). Results demonstrate that our method generates accurate poses and outperforms three state-of-the-arts [18, 24, 25]. In summary, our contributions are the following:

- We propose a segmentation-based approach for skeleton detection and hand bounding box localization.
- We propose a multi-scale heatmap regression architecture that uses the hand skeleton as additional information to constrain the 2D hand pose estimation task. The reported qualitative and quantitative results demonstrate the competitiveness of the proposed method.
- We introduce a new dataset to validate the hand detection and the 2D pose estimation methods.

We organize the rest of the paper as the following. Section 2 gives the problem definition as well as the related work. Section 3 describes in detail our hand detection and pose estimation approaches and defines the required steps to build our hand pose dataset. Section 4 discusses the conducted experiments and the obtained qualitative and quantitative results. Finally, Section 5 provides the main conclusion of this work and a direction for further research.

## 2 Related work

### 2.1 Hand detection

The hand detection task identifies the hand region and distinguishes it from the background. It has many applications including, gesture recognition [26] hand segmentation and hand tracking [27]. Traditional computer vision methods follow a feature extraction and classification scheme for hand detection. They extract skin color features, shape features or combine the two types of features to represent the image [28]. Following, they utilize a classifier to check each pixel, whether it belongs to the hand or not [29].

Deep learning-based methods circumvent such bottlenecks by unifying feature extraction and classification phases. This combined strategy has been outperforming conventional methods for the last five years. For instance, [30] employs two streams Faster RCNN [8] for hand detection. The first stream extracts feature maps from depth video while the second one extracts it from RGB video. After that, they use an alignment stage to connect the two features and they run a region proposal network to classify

the pixels. Another method [31] applies multi-scale Faster-RCNN to avoid missing the small hands.

### 2.2 Hand pose estimation

Estimating the 2D hand pose has been an active and challenging area of research in computer vision. Recently deep learning-based methods achieve competitive performance in this task as well. We can classify these based on the input modality into two broad categories as depth-based and RGB-based. In the former class, several studies achieve accurate 2D pose estimation results for images containing single hand [11, 12, 13, 32]. Also, [33] handles multi-hands using pictorial structure models and Mask-RCNN.

RGB-based methods are more challenging and less studied in the literature. Early studies give the cropped hand image as input to the ResNet-based model to directly regress the 2D joint by minimizing the mean-squared error (MSE) between the predicted 2D joint annotations and their ground truth [18]. Recently, [25] employs a graph-based framework to allow features at each node to be represented by 2D spatial confidence maps. Also, [24] propose an adaptive graphical model network that includes two branches of CNNs computing unary and pairwise potential functions and a graphical model to integrate the calculated information. [34] employs a cascaded CNN to predict the silhouette information (mask) and the 2D key-points in an end-to-end manner after localizing the hand region. To perform efficient small hand 2D pose estimation, [35] simultaneously regresses the hand region of interests and hand key-points. Subsequently, it iteratively uses the hand ROIs as feedback information for boosting the hand keypoints estimation performance. [8] proposes the Limb Probabilistic Mask, which uses a Gaussian distribution mask rather than the one-hot mask. To address the self-occlusion issue, it splits the whole hand mask into five fingers and the palm. The 2D pose regression task employs the synthesized hand mask to model the structural relationship between the 2D keypoints. All the aforementioned state-of-the-art methods results are presented in Table 1 that summarized the hand detection and pose estimation techniques. Besides, it shows the used datasets, including LSMV [18], OneHand10K [34], CMU and MPII+NZSL [37]. The results are reported using the mean PCK metric [37], which is widely used to evaluate human and hand pose estimation methods. It considers the predicted joints as correct if the distance to the ground truth joint is within a certain threshold  $\gamma$ . Some approaches use a normalized threshold by dividing all the joints values by the size of the hand bounding box. In this work, we propose a new multi-scale heatmap regression architecture that uses the 2D skeleton as a constraint to accurately estimate the 2D hand pose for small and big hands.



	Hand detection	Model	Estimation method	$meanPCK_{a,b}^\uparrow$	Threshold $\gamma$
Gomez et al [18]	Faster R-CNN for bounding box detection	ResNet-50	Direct regression	80.74 on LSMV (Self-dataset)	0.01-0.06 (N)
Kong et al [24]	Cropping square image patches of annotated hands	Adaptive graphical model	Heatmaps detection	70.34 on CMU and 85.63 on LSMV	0.01-0.06 (N)
Kong et al [25]	Cropping square image patches of annotated hands	Spatial information aware graph convolutional network	Heatmaps detection	81.72 on MPII+NZSL, 71.65 on CMU and 85.56 on LSMV	0.01-0.06 (N)
Wang et al [34]	Semantic segmentation using CNN	Mask-pose cascaded CNN	Heatmaps detection	90.27 on OneHand10K (Self-dataset) and 74.82 on MPII+NZSL	0.2
Wang et al [35]	Hand region localization using CNN-based bounding box regression	Simultaneously regress the hand region of interests and hand key-points	Heatmaps detection	0.94 on OneHand10K	0.2
Chen et al [8]	Limb probabilistic mask with splitting the hand into fingers and palm	Nonparametric structure regularization machine	Direct regression	88.46 on OneHand 10k and 80.03 on CMU	0.1-0.3 and 0.04-0.012 (N)

Table 1: Summary of related 2D hand pose estimation approaches and their obtained results. We show the  $meanPCK$  metric for defined thresholds on a specific dataset.  $\uparrow$ : higher is better,  $a, b$ : begin and the end of the experimented interval of thresholds  $\gamma$  and N refers to a normalized threshold.

### 3 Proposed method

Our proposed approach for 2D hand pose estimation uses a skeleton-based approach to detect the hand and extract the bounding boxes. The second part uses the predicted skeleton as a constraint to guide the proposed multi-scale heatmap regression approach to predict the 2D joint locations of the cropped hand.

#### 3.1 Skeleton detection and hand bounding box localization

We represent the detected hand location in an image by a rectangular region with four corners. Faster-RCNN [8] type of deep network models directly regress the four corner coordinates from the given hand image.

Alternatively, we can predict the 2D hand skeleton and extract the bounding box in a post-processing step (Figure 1). Direct regression of the bounding box is useful for hand cropping but cannot be further exploited for other tasks. In contrast, estimating the hand skeleton includes useful information that guides the 2D pose estimation. Also, the segmentation task is less challenging than predicting the bounding box.

Of course, one needs to have the training data with corresponding skeletons. We can obtain this type of data using a 3D hand tracker and an RGB camera to provide the 2D key-points (see Section 3.3). We create the ground truth data for the skeleton by connecting the joints in each finger

and attaching the palm to the ends of each finger. Also, we represent each joint location by the standardized Gaussian blob.

We can treat hand skeleton data as a segmentation mask. Thus, we use the well-known UNet architecture [23] since it is one of the best encoder-decoder architectures for semantic segmentation. It has two major properties. The first one is the skip connections between the encoder and the decoder layers that enable the network to learn the location and the contextual information. The second property is its symmetry, leading to better information transfer and performances.

The model outputs single feature maps on which we apply a *sigmoid* activation function to bound the prediction values between 0 (background) and 1 (hand). We localize the bounding box using a post-processing step, in which we identify the foreground pixels, and then we apply a region growing algorithm. In our case, the horizontal and vertical boundaries of the recovered regions are reported as the location of the detected hand. Our model robustly differentiates between the skin of the hand and that of the face. Also, it can detect the hand even in cluttered images or different lightning conditions (see Section 4.2).

Concerning the loss function, we did two experiments. In the first one, we only used the  $L_1$  loss function, which can not robustly localize the skeleton and adversely affecting the bounding box localization results. In contrast, using the combination of  $L_1$  loss (Equation 1) and a *SoftDice*

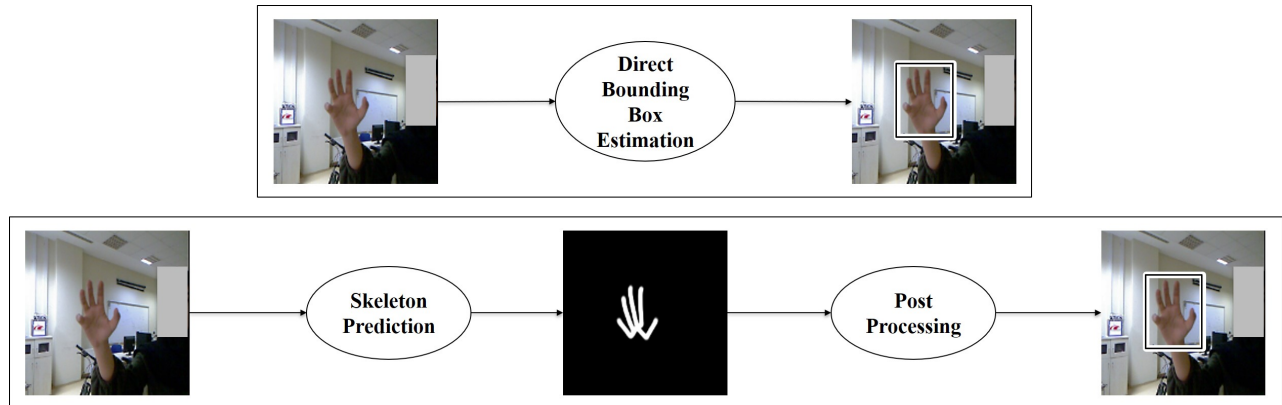


Figure 1: The proposed method for hand bounding box detection. Unlike many deep learning approaches that use Faster R-CNN [8] model to directly estimate the bounding box (top), we predict the skeleton image and infer the bounding box in a post-processing step (bottom).

(Equation 2) loss with their empirical weights can robustly localize the hand (Equation 3).

$$L_1(x, \hat{x}) = \|x - \hat{x}\|_1 \quad (1)$$

$$SoftDice(x, \hat{x}) = 1 - \frac{2\hat{x}^T x}{\|\hat{x}\|_2^2 + \|x\|_2^2} \quad (2)$$

$$Total(x, \hat{x}) = \lambda_1 L_1(x, \hat{x}) + \lambda_2 SoftDice(x, \hat{x}) \quad (3)$$

Where:  $x$ ,  $\hat{x}$ ,  $\lambda_1$  and  $\lambda_2$  represent the ground truth skeleton, the predicted skeleton and the two hyperparameters of the loss function (set to 0.4 and 0.6, respectively). We trained the model for 20 epochs using a batch size of 8.

### 3.2 Multi-scale heatmaps regression for 2D hand pose estimation

Most of the existing hand pose estimation methods predict the heatmaps at a single-scale. However, the hand in the original image can have several sizes (close/far hands). Hence, when we use a single scale image, the cropped hand image size cannot be suitable for all the dataset samples.

To address this limitation, we propose a multi-scale heatmaps regression architecture that performs the back-propagation process for many resolutions providing better joint learning for both large and small hands. Moreover, the cropped hand image would include some parts of the background. To overcome this problem, we employ the predicted hand skeleton to act as an attention mechanism for the network to focus on hand pixels. This makes the 2D pose regression task less challenging to optimize.

Figure 2 shows our skeleton-aware multi-scale heatmaps network approach for 2D hand pose estimation. We feed the concatenation of the cropped hand image and the predicted skeleton to the first convolution layer. The latter is followed by two downsampling ResNet blocks, two upsampling ResNet blocks, and a final transposed convolution

layer that recovers the input resolution. After each downsampling (similarly upsampling), we apply a  $1 \times 1$  convolution layer followed by a *sigmoid* activation function to output 21 or 20 feature maps representing the heatmaps in GTHD or LSMV datasets, respectively. The heatmaps resolution is divided/multiplied by two after each downsampling/upsampling.

In test time, we calculate a weighted average of the predicted five heatmaps to find the coordinate of the 2D key-points. We formulate the loss function in (Equation 4) as:

$$L(x, \hat{x}) = \sum_{i=1}^k \delta_i \|x_i - \hat{x}_i\|_2^2 \quad (4)$$

Where:  $k$  is the number of scales including the full resolution output, and  $\delta_i$  is the weight given for each scale. In our experiments we choose  $k = 5$  and  $\delta_i$  is set be 1, 1/2 and 1/4 for scales 128, 64 and 32 respectively.

### 3.3 Datasets

Deep learning methods require a large number of labeled data for training. There is a lack of datasets that has RGB hand images with their 2D annotations that we can use to train our proposed approaches. For example, [38] has RGB images with their 2D annotations, but they are both small scale and do not describe the hand by joint annotations.

Our method has been implemented and tested on two different datasets. The first one is LSMV [18], which is one of the large-scale datasets that provide the hand bounding boxes, the 2D key-points as well as the 3D pose. We split the data into 60000, 15000, and 12760 samples for the training set, validation set, and test set, respectively. While LSMV [18] can be used to train and validate the 2D hand pose estimators, it can not be used for hand detection since it does not have images without hands.

To overcome this limitation, and train both the hand detector and the hand pose estimator, we have built our own

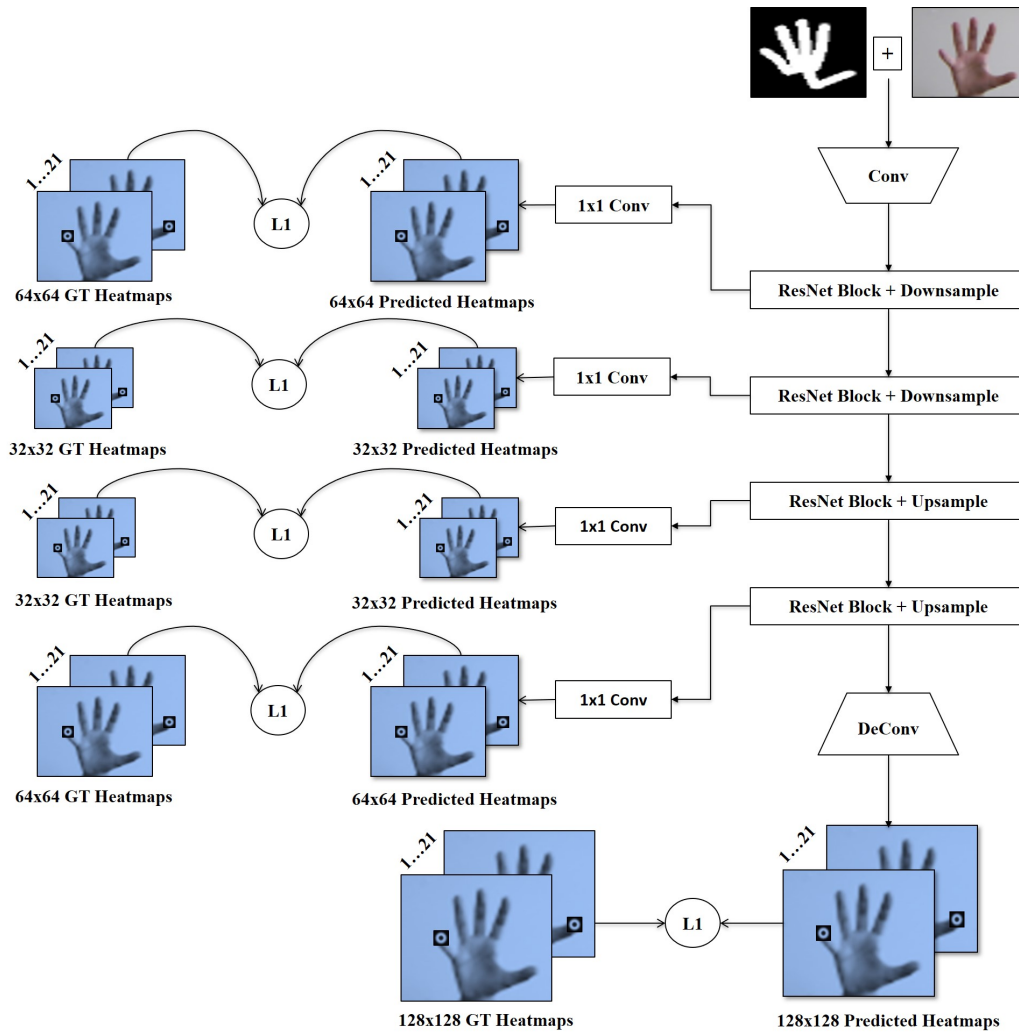


Figure 2: The overall architecture of the proposed 2D hand pose estimation approach uses the hand skeleton as a constraint and estimates the joint heatmaps from multiple scales.

dataset (GTHD) using an RGB camera and a Leap Motion sensor [39]. It is composed of two subsets; The first one has 60 thousand RGB images with their corresponding hand bounding boxes, 2D keypoints, and 3D pose. The second set has 15 thousand RGB images that present either the background or people who do not show their hands. The new dataset has a large variation in hand poses, backgrounds, skin color and texture

The RGB camera provides an image with a resolution of  $640 \times 480$  pixels. The leap motion controller is a combination of hardware and software that senses the fingers of the hand to provide the 3D joint locations. Hence, a projection process from 3D space to the 2D image plane is necessary. We achieve this goal in two steps. In the first one, we use OpenCV to estimate specific intrinsic parameters of the camera. In the second step, we estimate the extrinsic parameters between the leap motion controller and the camera. To get the correct pose with its corresponding image, we synchronize the two sensors in time.

Finally, to find the rotation and translation matrices, we

manually mark one key-point in a set of hand images and solve the *PnP* problem by computing the 3D-2D correspondences [40]. Figure 3 illustrates the results of the calibration process. We randomly split the GTHD dataset into a training set (75%), a validation set (10%) and a test set (15%).

### 3.4 Evaluation metrics

We report the performance of the hand skeleton detection module using famous classification metrics, such as *Accuracy*, *Precision*, *Recall* and *F1*. Furthermore, we calculate the Area Under ROC Curve (*AUC*) for GTHD datasets since it measures how well the two classes (Hand and No-Hand) are separable. It calculates the trade-off between the true positive rate and the false positive rate. Also, we report the Intersection over Union (*IOU*) metric to quantify our model performance in the hand bounding box detection task. It evaluates the predicted bounding boxes by comparing them against the ground truth.

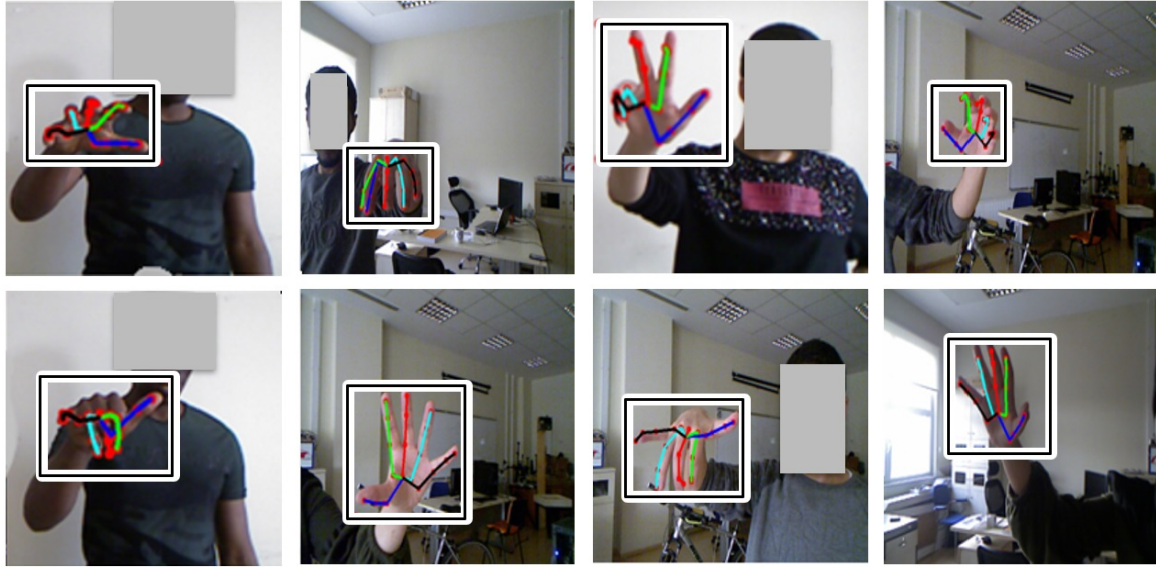


Figure 3: Examples of our hand dataset images having the bounding boxes and 21 joints annotations taken from four subjects and covering many pose and backgrounds.

To quantitatively evaluate the performance of the proposed 2D hand pose regression methods, we use the Probability of Correct Keypoint (PCK) metric [37] as it is used frequently in human and hand pose estimation tasks. We use a normalized threshold by dividing all the joints values by the size of the hand bounding box. Also, for additional quantification of the performance of the proposed method, we report the mean joint pixel error (MJPE) over the input hand image with  $128 \times 128$  resolution.

## 4 Experiments

### 4.1 Implementation details

We train the models for 30 epochs using a batch size of 8 and Adam as an optimizer. We initialize the learning rate to 0.01, and we decrease it after every eight epochs by 10%. We conduct all experiments on NVIDIA GTX 1080 GPU using PyTorch v1.6.0.

Before giving the RGB image as input to the model, we resize and normalize the datasets by subtracting the mean from all the images. The number of heatmaps is the number of joints where we represent each one with a Gaussian blob in a map of the same size of the image. The coordinate of the joint is the location of the highest value in the heatmap. We find them by applying the *argmax* function.

To validate our hand detection approach, we use different degrees of skeleton thickness (1, 3 and 6). In the first case, the skeleton is simply composed of lines. In other cases, it has thicker connections and regions around the joints. Furthermore, we select a threshold that represents the the number of foreground pixels to be the criterion to separate the two classes (Hand and NoHand).

Dataset	Faster-RCNN [8]	Ours
GTHD	0.912	0.923
LSMV	0.895	0.917

Table 2: Bounding box evaluation on LSMV and our GTHD dataset with IOU.

### 4.2 Hand detection and bounding-box estimation results

Our approach for hand bounding box localization can robustly estimate the hand skeleton and localize the hand bounding box for the two datasets. It does not produce any false positives for background images or images with people who do not show their hands (see Figure 4 and Figure 5).

The correct threshold for selecting *Hand* from *NoHand* depends on the data. A robust threshold should eliminate the noise and be in an interval that does not miss samples from the dataset distribution. In other words, the selected threshold should decrease both the false-negative rate (adding samples from the *NoHand* class) and false positive rate (missing samples from the *Hand* class) to achieve high performance and robustly detect the hand. Figure 7. shows that selecting a threshold from the interval [200, 400] is the best choice for our dataset. Also, the thickest skeleton representation seems to be more robust to the noise. It outperforms the other representations and achieves a higher performance ( $AUC = 0.99$ ). Finally, our approach records high scores of  $Accuracy = 0.99$ ,  $Precision = 0.97$ ,  $Recall = 0.99$  and  $F1 = 0.98$ .

We do not report the AUC for the LSMV dataset because it does not have images without hands. Nevertheless, we predict the skeleton representation to extract the

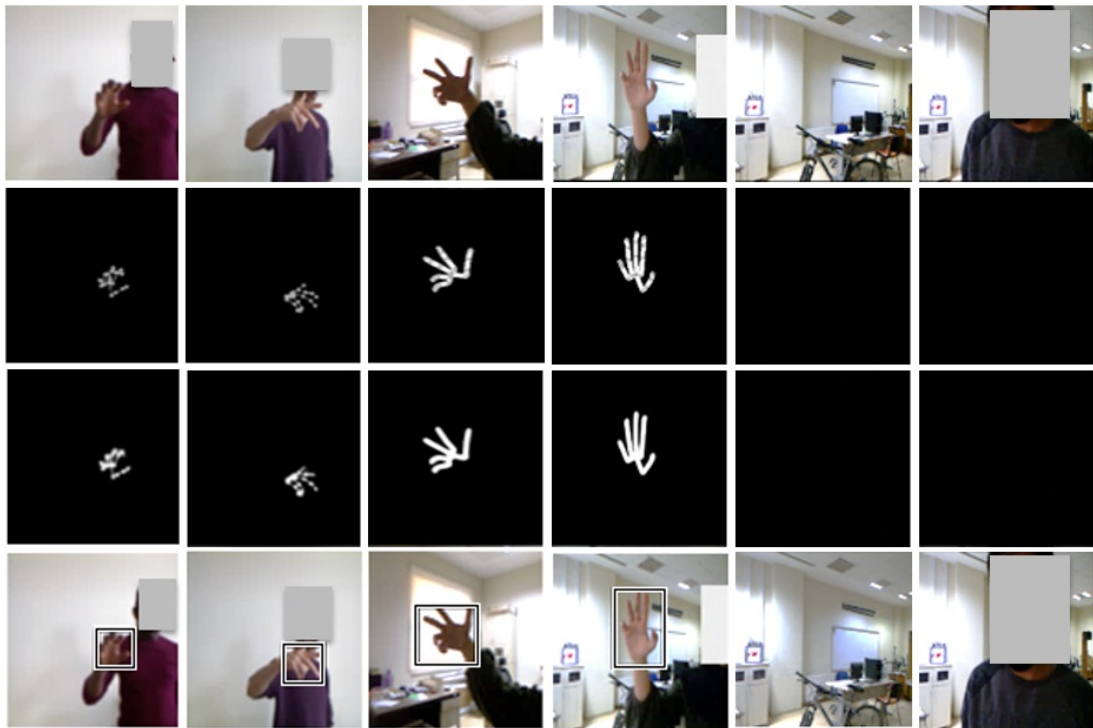


Figure 4: The results of the skeleton estimation and the bounding box localization on the GTHD dataset using thick and thin skeleton representations. The rows from top to down show: the input image, the ground truth skeleton, the predicted skeleton, and the obtained bounding boxes.

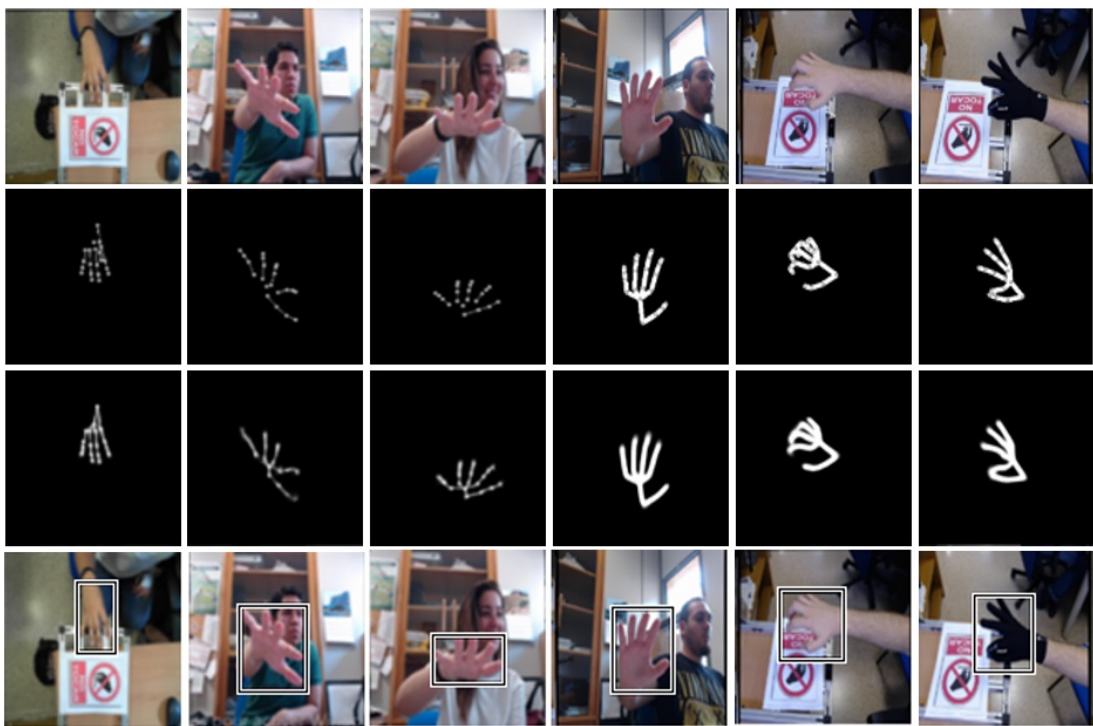


Figure 5: The results of the skeleton estimation and the bounding box localization on the LSMV dataset [18] using thin and thick skeleton representations. The rows from top to down show: the input image, the ground truth skeleton, the predicted skeleton, and the obtained bounding boxes.



Figure 6: Qualitative results for 2D hand pose estimation on GTHD and LSMV datasets. The columns from left to right in each image show: the direct regression proposed in [18], our proposed skeleton aware multi-scale heatmaps regression and the ground truth joints.

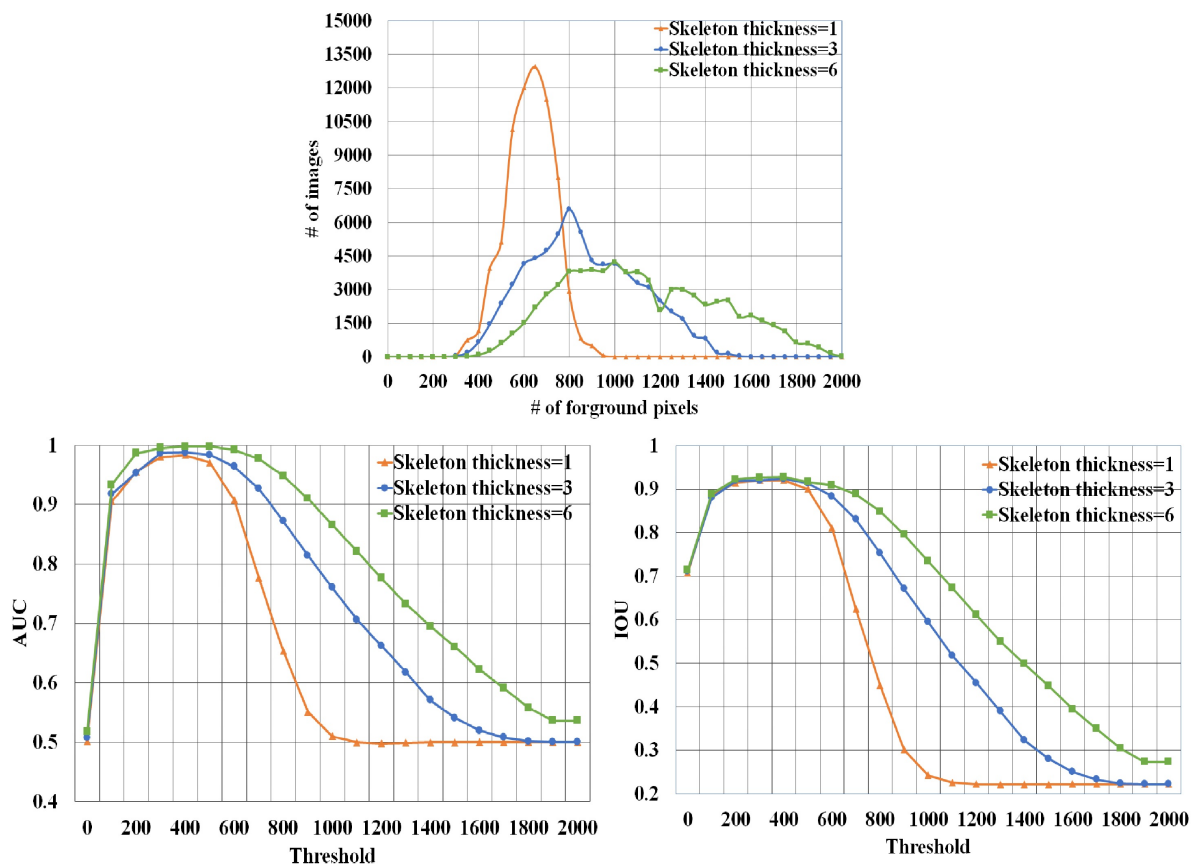


Figure 7: The impact of the threshold selection on the performances.

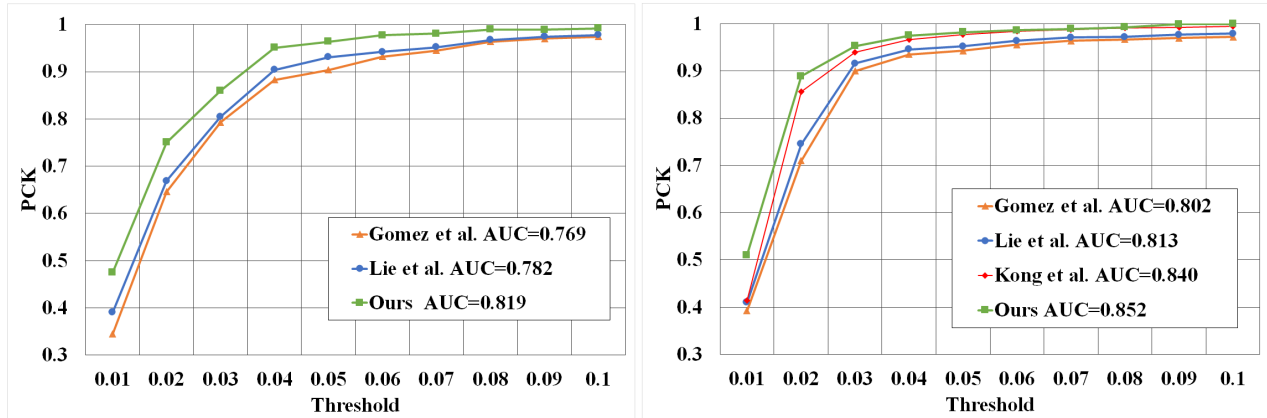


Figure 8: Quantitative comparison of the proposed 2D hand pose estimation with the other methods [18, 41, 24] using PCK metric. Left for GTHD and right for LSMV.

hand bounding boxes and perform our proposed 2D hand pose estimation method (Figure 5). Also, we report *IOU* in Table 2 showing that the proposed method outperforms Faster-RCNN [8].

To show the influence of the hand detection step on the bounding box localization performance, we record AUC and IOU metrics for different thresholds on the GTHD dataset. Figure 7 shows that the performance of the bounding box localization is strongly related to skeleton detection. Also, the thickest skeleton representation seems to be more robust to the noise. It outperforms the other representations and achieves a higher performance (0.99 in AUC and 0.92 in IOU).

### 4.3 Pose estimation results

The proposed method can robustly estimate the 2D hand pose even in the cases of complex poses and cluttered images. Figure 6 shows some randomly selected test images on LSMV and GTHD datasets.

We compare the proposed pose estimation approach against three-deep learning-based methods [18, 24, 25] on the LSMV dataset. Our baseline is [18] that uses ResNet-50 architecture [5] to directly regress the 2D joints from RGB images. The other deep-based methods [24, 25] are two of the existing state-of-the-art in 2D hand pose estimation. The proposed skeleton-aware multi-scale heatmaps regression method outperforms [18, 24, 25] since it learns the joint location from many resolutions. It reports the highest PCK across all the thresholds (Table 3).

To further demonstrate the effectiveness of the proposed approach, we conduct additional experiments on the GTHD dataset. In the first one, we perform two state-of-the-art methods [18, 41]. The second experiment applies single-scale heatmap regression using UNet architecture [23] on  $128 \times 128$  resolution images. The third experiment performs our multi-scale heatmaps regression without the skeleton information. In the last experiment, we perform our skeleton aware multi-stage heatmap regression archi-

tecture shown in Figure 2. We can see from Figure 8 that our method achieves a high PCK score (0.98) with a small threshold in LSMV and GTHD datasets. Furthermore, the hand skeleton representation improves the proposed multi-scale heatmaps regression method since it constrains the 2D pose estimation task (Table 4).

Estimating the 2D hand pose using the single-scale heatmaps regression outperforms the direct regression since the detected heatmaps help CNNs to learn better the joint locations and converge faster (Figure 8). Finally, our proposed method for 2D hand pose estimation provides more improvement for our dataset since it has more complex poses, face occlusion cases, and lighting condition variations (Figure 8 and Table 4).

## 5 Conclusion

In this work, we propose a new learning-based method for 2D hand pose estimation. It performs multi-scale heatmaps regression and uses the hand skeleton as additional information to constrain the regression problem. It provides better results compared with the direct regression and single-scale heatmaps regression. Also, we present a new method for hand bounding box localization that first estimates the hand skeleton and then extracts the bounding box. This approach provides accurate results since it learns more information from the skeleton. Furthermore, we introduce a new RGB hand pose dataset that can use both for hand detection and 2D pose estimation tasks.

For future work, we plan to exploit our 2D hand pose estimation method to improve the 3D hand pose estimation from an RGB image. Also, we plan to incorporate other constraints that can restrict the hand pose estimation problem.

Threshold of PCK	0.01	0.02	0.03	0.04	0.05	0.06	meanPCK
Gomez et al [18]	39.27	71.12	90.43	93.56	94.38	95.69	80.74
Kong et al [24]	41.38	85.67	93.96	96.61	97.77	98.42	85.63
Kong et al [25]	41.27	85.89	93.82	96.43	97.61	98.29	85.56
Ours	<b>51.02</b>	<b>88.91</b>	<b>95.30</b>	<b>97.54</b>	<b>98.27</b>	<b>98.63</b>	<b>88.27</b>

Table 3: Comparison with the state-of-the-art methods on the LSMV datasets with the PCK metric.

Methods	GTHD	LSMV
Gomez et al [18]	13.20	10.00
Lie et al. [41]	6.25	8.05
Single-scale [23]	7.33	5.87
Ours w/o skeleton	5.89	4.95
Ours	<b>5.51</b>	<b>4.67</b>

Table 4: Comparison with the state-of-the-art methods on GTHD and LSMV datasets with Mean pixel errors.

## References

- [1] El-Sawah, A., Georganas, N. D., & Petriu, E. M. (2008). A prototype for 3-D hand tracking and posture estimation. *IEEE Transactions on Instrumentation and Measurement*, 57(8), 1627-1636. <https://doi.org/10.1109/TIM.2008.925725>
- [2] Chen, T. Y., Wu, M. Y., Hsieh, Y. H., & Fu, L. C. (2016, December). Deep learning for integrated hand detection and pose estimation. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (pp. 615-620). IEEE. <https://doi.org/10.1109/icpr.2016.7899702>
- [3] Samed, S., Ferhat, C., Kevser, S. (2021, Vol 45, No 1). A Generative Model based Adversarial Security of Deep Learning and Linear Classifier Models. *Informatica* (pp.33-64). <https://doi.org/10.31449/inf.v45i1.3234>
- [4] Biserka, P., Tatjana, P., Natasa, S., Aleksandra, S., Mirjana, K. (2021, Vol 45, No 3). Machine Learning with Remote Sensing Image Data Sets. *Informatica* (pp.347-344). <https://doi.org/10.31449/inf.v45i3.3296>
- [5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). <https://doi.org/10.1109/cvpr.2016.90>
- [6] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440). <https://doi.org/10.1109/cvpr.2015.7298965>
- [7] Sina, S., Sara, K. (2020, Vol 44, No 4). Teeth Segmentation of Bitewing X-Ray Images Using Wavelet Transform. *Informatica* (pp.421-426). <https://doi.org/10.31449/inf.v44i4.2774>
- [8] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 91-99. <https://doi.org/10.1109/tpami.2016.2577031>
- [9] Stefan, K., Martin, G., Hristijan, G., Matjaz, G. (2021, Vol 45, No 2). Analysis of Deep Transfer Learning Using DeepConvLSTM for Human Activity Recognition from Wearable Sensors. *Informatica* (pp.289-296). <https://doi.org/10.31449/inf.v45i2.3648>
- [10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105. <https://doi.org/10.1145/3065386>
- [11] Tompson, J., Stein, M., Lecun, Y., & Perlin, K. (2014). Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5), 1-10. <https://doi.org/10.1145/2629500>
- [12] Spurr, A., Song, J., Park, S., & Hilliges, O. (2018). Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 89-98). <https://doi.org/10.1109/cvpr.2018.00017>
- [13] Wan, C., Probst, T., Van Gool, L., & Yao, A. (2018). Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5147-5156). <https://doi.org/10.1109/cvpr.2018.00540>
- [14] Zimmermann, C., & Brox, T. (2017). Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision* (pp. 4903-4911). <https://doi.org/10.1109/iccv.2017.525>
- [15] Spurr, A., Song, J., Park, S., & Hilliges, O. (2018). Cross-modal deep variational hand pose estimation.



- In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 89-98). <https://doi.org/10.1109/cvpr.2018.00017>
- [16] Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., & Theobalt, C. (2018). Generated hands for real-time 3d hand tracking from monocular rgb. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 49-59). <https://doi.org/10.1109/cvpr.2018.00013>
- [17] Santavas, N., Kansizoglou, I., Bampis, L., Karakasis, E., & Gasteratos, A. (2020). Attention! a lightweight 2d hand pose estimation approach. *IEEE Sensors Journal*, 21(10), 11488-11496. <https://doi.org/10.1109/jsen.2020.3018172>
- [18] Gomez-Donoso, F., Orts-Escolano, S., & Cazorla, M. (2019). Large-scale multiview 3d hand pose dataset. *Image and Vision Computing*, 81, 25-33. <https://doi.org/10.1016/j.imavis.2018.12.001>
- [19] Carreira, J., Agrawal, P., Fragkiadaki, K., & Malik, J. (2016). Human pose estimation with iterative error feedback. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4733-4742). <https://doi.org/10.1109/cvpr.2016.512>
- [20] Bulat, A., & Tzimiropoulos, G. (2016, October). Human pose estimation via convolutional part heatmap regression. In European Conference on Computer Vision (pp. 717-732). Springer, Cham. [https://doi.org/10.1007/978-3-319-46478-7\\_44](https://doi.org/10.1007/978-3-319-46478-7_44)
- [21] Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., & Murphy, K. (2017). Towards accurate multi-person pose estimation in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4903-4911). <https://doi.org/10.1109/cvpr.2017.395>
- [22] Iqbal, U., Molchanov, P., Gall, T. B. J., & Kautz, J. (2018). Hand pose estimation via latent 2.5 d heatmap regression. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 118-134). [https://doi.org/10.1007/978-3-030-01252-6\\_8](https://doi.org/10.1007/978-3-030-01252-6_8)
- [23] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
- [24] Kong, D., Chen, Y., Ma, H., Yan, X., & Xie, X. (2019). Adaptive graphical model network for 2d handpose estimation. arXiv preprint arXiv:1909.08205.
- [25] Kong, D., Ma, H., & Xie, X. (2020). Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation. arXiv preprint arXiv:2009.12473.
- [26] Ren, Z., Meng, J., Yuan, J., & Zhang, Z. (2011, November). Robust hand gesture recognition with kinect sensor. In Proceedings of the 19th ACM international conference on Multimedia (pp. 759-760). <https://doi.org/10.1145/2072298.2072443>
- [27] Hammer, J. H., Voit, M., & Beyerer, J. (2016, July). Motion segmentation and appearance change detection based 2D hand tracking. In 2016 19th International Conference on Information Fusion (FUSION) (pp. 1743-1750). IEEE.
- [28] Kumar, A., & Zhang, D. (2006). Personal recognition using hand shape and texture. *IEEE Transactions on image processing*, 15(8), 2454-2461. <https://doi.org/10.1109/tip.2006.875214>
- [29] Ong, E. J., & Bowden, R. (2004, May). A boosted classifier tree for hand shape detection. In Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings. (pp. 889-894). IEEE. <https://doi.org/10.1109/afgr.2004.1301646>
- [30] Liu, Z., Chai, X., Liu, Z., & Chen, X. (2017). Continuous gesture recognition with hand-oriented spatiotemporal feature. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 3056-3064). <https://doi.org/10.1109/iccvw.2017.361>
- [31] Hoang Ngan Le, T., Zheng, Y., Zhu, C., Luu, K., & Savvides, M. (2016). Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 46-53). <https://doi.org/10.1109/cvprw.2016.13>
- [32] Garcia-Hernando, G., Yuan, S., Baek, S., & Kim, T. K. (2018). First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 409-419). <https://doi.org/10.1109/cvpr.2018.00050>
- [33] Duan, L., Shen, M., Cui, S., Guo, Z., & Deussen, O. (2018). Estimating 2d multi-hand poses from

- single depth images. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops (pp. 0-0). [https://doi.org/10.1007/978-3-030-11024-6\\_17](https://doi.org/10.1007/978-3-030-11024-6_17)
- [34] Wang, Y., Peng, C., & Liu, Y. (2018). Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11), 3258-3268. <https://doi.org/10.1109/tcsvt.2018.2879980>
- [35] Wang, Y., Zhang, B., & Peng, C. (2019). Srhandnet: Real-time 2d hand pose estimation with simultaneous region localization. *IEEE transactions on image processing*, 29, 2977-2986. <https://doi.org/10.1109/tip.2019.2955280>
- [36] Chen, Y., Ma, H., Kong, D., Yan, X., Wu, J., Fan, W., & Xie, X. (2020). Nonparametric structure regularization machine for 2d hand pose estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 381-390). <https://doi.org/10.1109/wacv45572.2020.9093271>
- [37] Simon, T., Joo, H., Matthews, I., & Sheikh, Y. (2017). Hand keypoint detection in single images using multi-view bootstrapping. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1145-1153). <https://doi.org/10.1109/cvpr.2017.494>
- [38] Pisharady, P. K., Vadakkepat, P., & Poh, L. A. (2014). Hand posture and face recognition using fuzzy-rough approach. In *Computational Intelligence in Multi-Feature Visual Pattern Recognition* (pp. 63-80). Springer, Singapore. [https://doi.org/10.1007/978-981-287-056-8\\_5](https://doi.org/10.1007/978-981-287-056-8_5)
- [39] Potter, L. E., Araullo, J., & Carter, L. (2013, November). The leap motion controller: a view on sign language. In Proceedings of the 25th Australian computer-human interaction conference: augmentation, application, innovation, collaboration (pp. 175-178). <https://doi.org/10.1145/2541016.2541072>
- [40] Beardsley, P., Murray, D., & Zisserman, A. (1992, May). Camera calibration using multiple images. In *European Conference on Computer Vision* (pp. 312-320). Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-55426-2\\_36](https://doi.org/10.1007/3-540-55426-2_36)
- [41] Li, S., & Chan, A. B. (2014, November). 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision* (pp. 332-347). Springer, Cham. [https://doi.org/10.1007/978-3-319-16808-1\\_23](https://doi.org/10.1007/978-3-319-16808-1_23)

# A Blockchain and NLP Based Electronic Health Record System: Indian Subcontinent Context

Pranab Kumar Bharimalla  
School of Computer Engineering, KIIT University, India  
E-mail: pranab.bharimalla@gmail.com

Hammad Choudhury  
Infosys Ltd., USA  
E-mail: hchoudhury99@gmail.com

Shantipriya Parida  
Idiap Research Institute, Martigny, Switzerland  
E-mail: shantipriya.parida@idiap.ch

Debasish Kumar Mallick and Satya Ranjan Dash (Corresponding Author)  
School of Computer Applications, KIIT University, India  
E-mail: mdebasishkumar@gmail.com, sdashfca@kiit.ac.in

**Keywords:** blockchain, electronic health record, ResNet, CNN, natural language processing

**Received:** April 8, 2021

*The healthcare system in the Indian subcontinent is plagued with numerous issues related to the access, transfer, and storage of patient's medical records. The lack of infrastructure to properly communicate and track records between all key participants has allowed the distribution of counterfeit drugs, dependency on unsafe methods of communication, and lack of trust between patients and providers. During the global COVID-19 pandemic, the need for a robust communication and record tracking system has been further emphasized. To facilitate efficient communication and mitigate the mentioned issues, a nationwide EHR (electronic health record) system must be introduced to bring the healthcare system into digital space. To further enhance security, efficiency, and cost, the innovation of Blockchain is introduced. Blockchain is a decentralized data structure that allows secure transactions between untrusted parties without needing a central authority. In this paper, a Hyperledger fabric-based Blockchain Electronic Healthcare Record (EHR) system is proposed. The system is integrated with technologies such as NLP (Natural Language Processing), and Machine Learning to provide users with practical features.*

*Povzetek: Predstavljen je elektronski zdravstveni zapis na osnovi bločnih in NLP tehnologij v kontekstu Indije.*

## 1 Introduction

The SARS-CoV-2 or commonly known as the Coronavirus pandemic, has challenged healthcare systems worldwide and has exposed vulnerabilities even among the best prepared due to its uncertainty of transmission, the unavailability of a patient's proper medical history, and lack of adequate contact tracing, to name a few examples. Tracking a threat such as this pandemic requires dynamic adaptation of resource deployment to manage rapidly evolving care demands, ideally based on real-time data from a large population sample. Healthcare issues plague all nations regardless of development status due to environmental, economic, or societal conditions; according to The Institute of Medicine (IoM), over one hundred thousand people die each year from preventable medical errors in the US [2]. While healthcare systems in developed nations are in no way perfect as stated by IoM and proven by the pandemic,

developing or underdeveloped regions such as the Indian sub-continent remain more vulnerable. Some common issues that have arisen are due to the lack of communication and tracking infrastructures, such as the influx of counterfeit drugs in the market, dependability on handwritten prescriptions, especially in remote areas lacking any computer systems, and almost nonexistent integration between healthcare and insurance systems. The inadequate infrastructure facilitates the lack of accountability for healthcare providers and further damages relations with patients. Resolving the many health care issues faced in the Indian subcontinent is a formidable challenge but can be significantly improved with the implementation of an end-to-end integrated Electronic Health Record (EHR). In comparison to paper-based record keeping, a practice still utilized in the Indian subcontinent, EHR has clear and distinct advantages [7]. It is fair to say that EHR holds a lot of promise: lower morbidity and mortality rates, better continuity of

care, increased efficiencies, fewer adverse drug reactions, and, most importantly, lower healthcare costs. Paper-based records are more susceptible to human error due to fundamental factors such as legibility or loss of the physical item, causing a delay in treatment and possible fatality, which could have been prevented [18]. Also, the impact of EHR will undoubtedly be felt during the coming months as the global effort to distribute and administer the Coronavirus vaccine intensifies.

An ideal healthcare system should be open, affordable, innovative, and secure. Care costs should not hamper patients from receiving required treatments or buying medicines, and healthcare should be affordable to all regardless of wealth [19]. In the current scenario, an ideal healthcare system might seem far-fetched, but there is always room for improvement using the latest innovations [13]. As our digital infrastructure evolves, the need for robust privacy is increasing. Because of the sensitivity of an individual's healthcare information, a breach in the system could jeopardize the identity of a patient and the reputation of providers. According to [10], the patient's data contains information that is highly prized by cybercriminals. Few noticeable hacks of medical information such as AMCA Data Security Incident (approx. 25M records) and Anthem data breaches (approx. 80M records) caused enormous damage to the medical system [20]. Blockchain technology could help secure and protect sensitive patient information and is emerging as an alternative to the conventional way to log transactions and transfer data through a trusted intermediary to provide validity to the transaction [3].

So far, we have discussed various healthcare issues prevailing in the Indian subcontinent, such as the lack of complete end-to-end EHR interlinking between individuals, stand-alone hospital recording systems, extensive use of handwritten prescriptions, unsafe hospital databases vulnerable to data manipulation by hospital authorities without patient permission and lack of accommodation for caregivers. We also understood how a Blockchain-based EHR system can be a perfect solution to these problems. This article proposes a patient-centered blockchain-based EHR system that identifies patients using their national ID, grants individual control over their health records, protects against unauthorized data manipulation, and allows scalability. Cloud integration stores large CT scans and X-ray reports. An EHR patient-centered system must accurately identify individuals using unique identifiers. There is no end-to-end EHR system with national ID cards like Aadhar or PAN (Permanent Account Number). Integrating unique identifiers like national IDs is essential to successfully identify patients. The proposed scalable system will enable existing EHR and healthcare databases to be integrated along with other useful features, including a mobile app-based interface to convert paper prescriptions to text using Natural Language Processing (NLP) algorithms to bring old paper-based medical records into the new system.

The paper is organized as follows: Section 2 briefly de-

scribes some related work and their major contributions as well as research gaps. The proposed system architecture is presented in Section 3, followed by the proposed algorithms. Section 4 discusses the implementation strategy as well as the analysis of the results. Finally, Section 5 provides the conclusion and suggestions for future research.

## 2 Related work

With the advent of different Blockchain platforms like Hyperledger Fabric, Ethereum, and Azure Blockchain Workbench, many patients centric, permission-based Blockchain EHR schemes have been proposed in the literature. [17] proposed a Permission-based EHR sharing system intending to enhance security and privacy. They also proposed a design access control policy algorithm with a smart contract and formulated a performance optimization mechanism of the system. However, this work ignores the reusability of existing healthcare records sitting in individual hospital databases. [8] proposed a Blockchain-based permissioned EHR system with the capability of data integration of local standalone EHR systems that house in different hospitals or clinics. The framework proposed to store metadata and access only in Blockchain whereas actual health records in the cloud. This is a novel concept, but actual patient-sensitive EHR data resides in the cloud and does not enjoy the immutability that Blockchain offers. [15] tried to enhance the framework proposed in [17]; and added few new modules like a chemist, insurance, and doctor's appointment. Even though the authors formulated a comprehensive approach, the work ignores the scalability of data and interoperability of existing EHR and healthcare databases. [6] proposed a similar Blockchain-based searchable encryption scheme for EHR that not only brings convenience to patients, healthcare providers but also to researchers. In the proposed system only, indexes are added to the Blockchain, whereas actual patient information is stored in an encrypted format in the cloud. There are different ways of encrypting data in the cloud and achieving privacy preservations. [9] defined a novel way of splitting EHR records into sub-messages and finally construct shares of EHR to store in different computer nodes locally and upload the indexes in healthcare Blockchain. [3, 11] evaluated the performance for common public and private/consortium Blockchain-based healthcare systems using metrics such as memory consumption, disk write and read performance, network data utilization, transaction execution per unit time, and CPU usage with consortium-based systems yielding the best performance results. [4] proposed a framework called Blockchain-Based Deep-Learning as a-Service. The framework shares EHR records among multiple healthcare users and operates in two phases to prevent collusion attacks through authentication and predicts possible future conditions for patients through deep learning. [5] proposed a Blockchain-based architecture that allows access to the database based on user

roles and enhances the traditional encryption system employing the Quantum blind signature to protect the system from quantum attacks using hyperledger fabric.

On the other hand, [14] proposed a system that uses AES cryptography to perform the cryptic operation and block chaining it through the hash keys. In addition to that, their proposed healthcare ecosystem includes a prediction model to diagnose the disease of the patient with the deep learning algorithm. In addition to the related works mentioned above, we also performed literature surveys on the ANN-based text extraction model, which could contribute to our work. [1] have implemented the Artificial Neural Network (ANN) approach for text extraction from 64 different types of prescriptions with 98% accuracy. [12] have proposed a CNN common approach for extracting numeric text using handwritten numbers commonly found in India, including Odiya, Telugu, Devnagari, Bangla, and English. The Bangla characters were 95% accurate, Devanagari characters were 98.54% accurate, Odia characters were 97.2% accurate, Telugu characters were 96.5% accurate, and English characters were 99.10% accurate.

### 3 Proposed model

The proposed system prototype is based on Hyperledger Fabric, an open-source distributed ledger technology built to meet enterprise requirements. It is a widely used private Blockchain option. Being a permissioned platform with improved configurability and modularity using pluggable consensus protocols, it is ideal for a range of industries requiring a level of trust between known participants in a governance model. For a transaction to occur, the user must be admitted with the organization's certificate authority, received the means for network authentication, the chain code must be deployed to the channel and installed on the peers, and both parties must have agreed to the endorsement policy. A transaction proposal using an SDK is constructed along with a signature to call a chain code function with parameters to update the ledger. After approvals from peers are received, the chain code is executed against the current database, and the response, read set, and write set are received; these values and the signature are sent back to the SDK to be parsed and consumed by the application. Once the application has validated the responses, the proposal and response are bounded in a transaction message for the ordering service; the ordering service creates blocks of transactions per channel. The transactions are sent to all peers in the channel, the peers complete a final validation, the ledger is updated, and the peer reports to the client about transaction validation or invalidation.

#### 3.1 System architecture

In the proposed patient-centric, Blockchain-based healthcare system, there are seven key participants, the patients or the public, doctors or caregivers, pharmacies, labs, insurance company staffs, government institutions, and the

admin user. Assistance from the government is necessary to implement a functional Blockchain-based EHR system. The government provides credibility to insurance companies, labs, pharmacies, doctors, and even patients through national identification numbers. The proposed system provides patients substantial control over their medical records, including the right to read, write, authorize, and revoke records in the Hyperledger Fabric Blockchain network. Doctors work closely with patients to diagnose conditions, plan treatments, and prescribe medication. Pharmacies work in parallel with doctors to distribute medications to patients. Due to the many factors related to an accurate diagnosis, labs are specialized in detecting distinct conditions with the help of specific tools and trained professionals. Insurance companies help share risk between a large population, making the cost of healthcare affordable for the public, especially during unexpected events such as accidents. Doctors, pharmacies, labs, and insurance companies can read and update the patient's medical record in the Hyperledger Fabric Blockchain network if access has been provided. The admin is critical for system maintenance, and they have unrestricted access to the system, including the right to read, write, update, remove and grant access to participants in the Hyperledger Fabric Blockchain network. The admin's enrollment certificate is obtained from the certification authority. Implementing a national health portal or EHR system is not possible without the government's involvement in India due to the sizable population of 1.4 billion. In the proposed system, the government institution acts as a founder organization or a trusted anchor that can provide credibility to hospitals, pharmacies, insurance companies, labs, and other institutions and provide them with trusted roles to play. The participants with trusted roles can create and issue credential schema and definition to the public or patients

The participants could register through a client application or SDK and request an enrollment certificate from a Membership Service Provider (MSP) to the certificate authority. An MSP allows peers to validate incoming transactions and sign off endorsements. After receiving the enrollment request, the certificate authority issues the certificate and private key with a new ID to enroll the participant. The Hyperledger Fabric Blockchain network distributes all transactions. Participants such as doctors, pharmacies, labs, insurance companies, etc., have different roles in the system and are only granted access when authorized. In the proposed system, an individual's identity could be verified against a national identification number such as Aadhar and be structured to contain identity information like names, date of birth, gender, and other identifying information which could be fetched from the Aadhar database. Patients can use the client application to update details like blood group, allergies, medications, insurance details, etc. Once the transaction is submitted, it will be broadcasted to the network. Endorsing peers will verify the transaction and authenticate using their certification and private key. The transaction will next go to the Orderer through

the SDK client. The Orderer creates a block and sorts the blocks based on different ordering algorithms (viz crash fault-tolerant) and broadcasts to the network peers. All the committing peers validate the blocks once again and check if it is from the correct Orderer and validate conflicts before committing. MS is the body that manages the network identities of organizations and users; however, it does not have access to medical records on the Blockchain network. The MS verifies participants based on TIN/PAN before enrolling them within the network. CouchDB, a NoSQL database that stores data in JSON-based format, is a popular database option used alongside Hyperledger Fabric. A network is comprised of peer groups that hold ledgers and smart contracts used to encapsulate shared network processes and information. In Hyperledger Fabric, transactions produced by smart contracts are contained in a chain code. The key components and processes of the proposed system are highlighted in the below architecture Figure 1. The algorithm 1 explains enrollment of patients whereas algorithm 2 refers to hospital, pharmacy, etc. addition to the network. Table 1 details the abbreviations used in both the algorithms.

### 3.2 Data pulling and sharing

In a permissioned blockchain system, the patients have the authority to decide who can read or update their records; however, the admin reserves the right to grant access to an institution in case of an emergency. Individual hospitals will integrate their EHR system with a Blockchain node and a web API that has full access to their local EHR to convert any existing SQL records to No SQL format for storage within Blockchain. A hybrid data management approach is utilized to facilitate EHR data scalability where all key patient information, including demographics, allergies, medications, and access controls, are stored in Blockchain, and sensitive medical files such as x-ray and scanning reports are stored in private cloud storage using encryption in Figure2.

In the proposed system, a patient can provide access to institutions and participants, including hospitals, doctors, insurance companies, and pharmacies, through the user interface using the web or mobile app. The patient will need to identify the participant requiring access, the category of data to be shared, and the period until the data is accessible. If a patient has visited a particular institution in the past and there are medical records contained in the local EHR system not found on the Blockchain network, they can initiate a pull request using web apps. Once the pull request is approved by the institution's admin user, the web apps will connect to the local EHR system to fetch relevant data, insert patient information into the Blockchain network and upload large files to the private cloud in an encrypted format.

### 3.3 Patient data management at hospital

During a routine or emergency hospital visit, the patient provides the hospital access to their medical records on the Blockchain network to check and amend their records based on the latest assessment. The Hospital must have a valid node on the Blockchain network and request keys from the network admin to permit the login. The patient will select the category of data to be shared and how long the Hospital will have access to the records. Once the Hospital has been provided access, they can read and update the records for an individual using their Aadhar identification card described in Figure3.

### 3.4 Patient data management at pharmacy, patho lab and insurance firm

During a pharmacy visit to fulfill a prescription, the patient provides the pharmacy access to their medical records on the Blockchain network, assuming the pharmacy has a valid node on the network and has requested keys from the network admin to enable login. The patient can provide itemized permissions where the pharmacy will only have access to select prescriptions through private datasets and control how long the pharmacy has access to the records. Once the pharmacy has been provided access, they can read and update the records for an individual using their Aadhar identification card.

During a lab visit at a specialized facility to assess specific conditions, the patient provides the lab access to their medical records on the Blockchain network. The lab requires a valid node on the network and must have requested keys from the network admin to allow login. The patient controls what data is to be shared and how long the lab will have access. Once the lab has been provided access, they can read and update the records for an individual using their Aadhar identification card. Large lab files such as x-ray reports can be encrypted and uploaded to private cloud storage. Similarly, Insurance firms need to update a patient's insurance and policy information for policy procurement or medical expense claims. The patient provides the insurance firm access to their medical records on the Blockchain network if the institution has a valid node on the network and has requested keys from the network admin to authorize a login. The patient will select the type of data to be shared and how long the insurance firm will have access to the record. Once access has been provided, they can read and update the records for an individual using their Aadhar identification card.

### 3.5 Patient uploads old handwritten/printed prescriptions and bills

A core proposal of this paper is to convert paper prescriptions to text using Natural Language Processing (NLP) algorithms to bring old paper-based medical records into the new system through a mobile app-based interface. In the

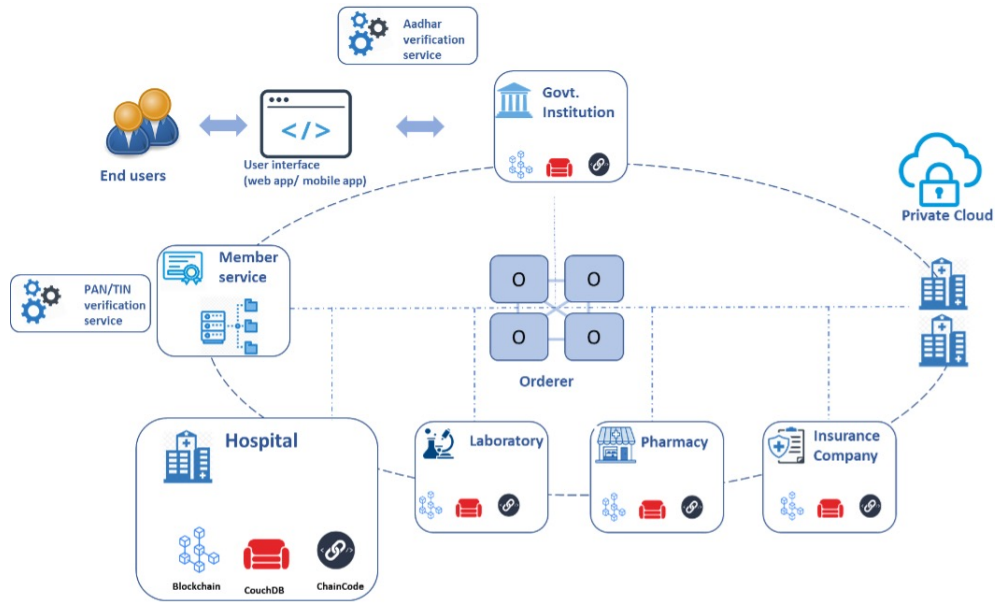


Figure 1: Proposed architectural Framework for Blockchain-based Healthcare System. It depicts key participants, components and transaction processes involved.

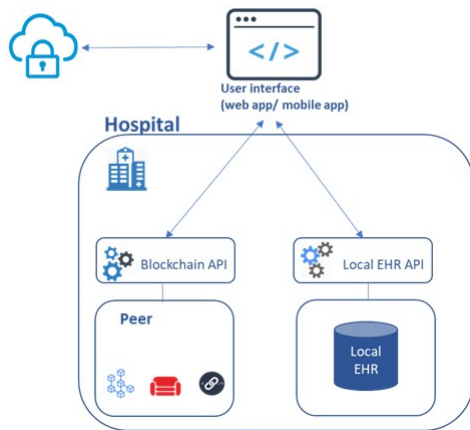


Figure 2: Secured Hospital data management and data pulling process from local EHR.

below module in Figure 4, the system flow for NLP-based data extraction is presented, highlighting the key components.

**3.5.1 Handwritten prescription data extraction**

**Convolutional Neural Network (CNN)** CNN is a Neural Network (NN) that performs convolutional operations instead of simple matrix multiplication operations; it is one of the layers. The structure of CNN consists of a Convolution layer, pooling layer, and fully connected layer. Feature extraction operations are done in the convolution layer, and the output of it is passed to the activation function. The size of output reduces by pooling layers and gives robust learn-

ing results for input data. By performing the convolution layer and pooling layer multiple times, global features can be obtained. In the end, extracted features are passed to the fully connected layer for regression and classification.

**Residual Network (ResNet)** According to the image processing research, the number of layers or depth of a network is crucial for the performance of a model, but the greater number of layers is responsible for the degradation. Much research shows those types of degradation are not caused by overfitting but due to the matter of optimization. The ResNet can solve the degradation problem by introducing a residual framework.

**Long Short-Term Memory (LSTM)** RNN is also a Neural Network that is specially designed for the processing of sequential data. In time-series data, the output  $t - 1$ -time step affects the decision of future time step  $t$ . So, RNN is not able to solve long sequence data. i.e., is called a vanishing or exploding problem. LSTM was designed to resolve this issue of vanishing or exploding problems. LSTM has an internal memory cell called a cell state. This gets the previous output and determines which element should be updated, erased, and maintained in the internal state vector. These processes are handled by four gates, forget gate  $f_t$ , output gate  $o_t$ , input gate  $g_t$ . which are shown in Figure 5.

The proposed model consists of Resnet-50 and three LSTM layers. To do non-linearity, Rectified Linear Unit (RELU) activation function is used in every convolutional layer. The images are divided into 28 sub-windows, so the image height is equal to the height of text-line images. The vector map will be produced by the last layer of convolution. The output of the last convolution fed into the first

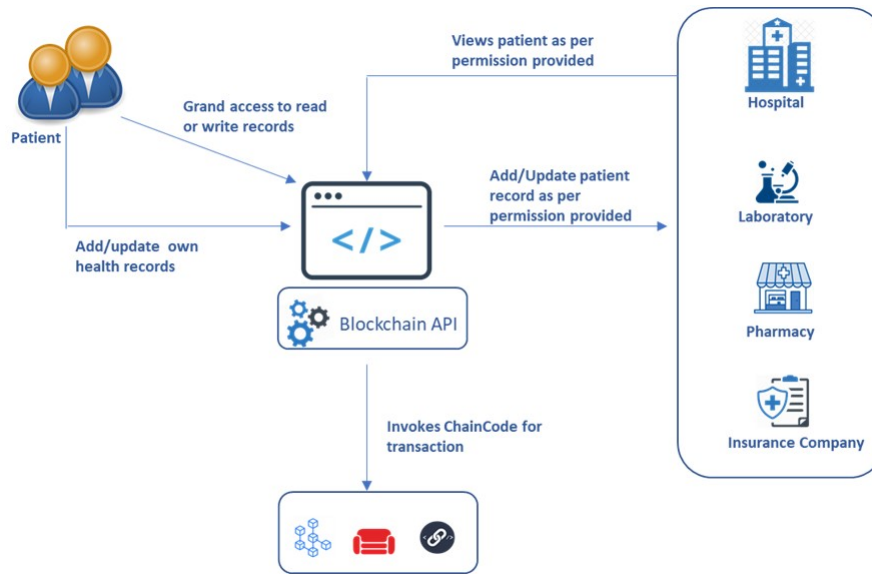


Figure 3: High-level access flow diagram: The patient can add own records to the Blockchain or grants access to other participants like Hospital, Laboratory, Pharmacy or Insurance company to add/update records.

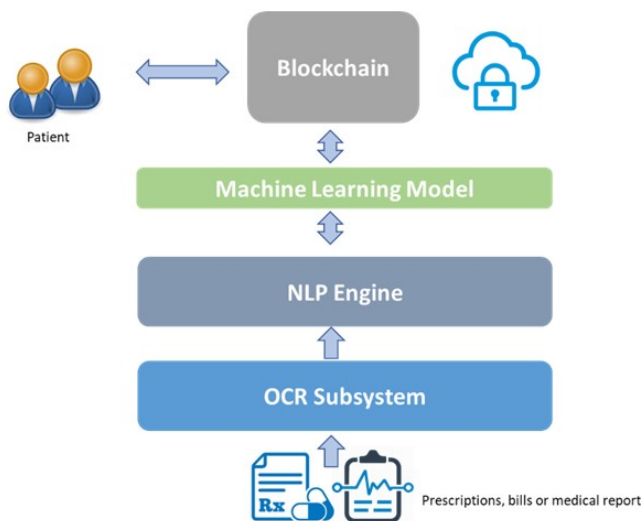


Figure 4: NLP based data extraction flow showing the components.

LSTM layer. By doing LSTM operations, the weights are optimized. Initially, 0.001 is used as the rate of learning.

### 3.5.2 Printed prescription data extraction

For extracting printed prescription data, we have used the Google tesseract model,[16] a pre-trained character made by Google. For the extraction process, we have done some preprocesses. For a more efficient model, first, we convert the image into grayscale. After the greyscale operation, the Otsu thresholding is next, where pixels are converted into zeros and ones. During thresholding, some of the pixels

#### Algorithm 1: Patient Enrollment.

---

**Input:** Enrollment Certificate ( $E_C$ ) from Certification Authority ( $C_A$ )  
**Output:** Successful registration of patient  
**Initialization:**  $N_{Admin}$  should be valid node.  
 $N_{Admin}$  can Write/Read/Remove/Update patients;  
**while** (*true*) **do**  
    **if**  $P_{AId}$  is valid **and**  $FetchAadharRecord(P_{AId})$   
        **not null then**  
             $P_{Rec} \leftarrow FetchAadharRecord(P_{AId})$   
            Add Patient ( $B_{NT}, P_{AId}$ )  
            Grant Access( $P_{AId}$ )  
            Create Record ( $P_{AId}, P_{Rec}, B_{NT}$ )  
        **else**  
            Invalid( $P_{AId}$ )  
        **end**  
    bool  $chk \leftarrow (0 : malicious, 1 : genuine)$   
    **if**  $!(behaviour(chk))$  **then**  
        | Remove Update ( $P_{AId}$ )  
    **else**  
        | Add Update ( $P_{AId}$ )  
    **end**  
**end**

---

may be lost. To restore those pixels, Erosion and Dilation operations are performed where Erosion expands some pixels, and Dilation shrinks some pixels as shown in Figure 6a, Figure 6b and Figure 7.

After these images complete preprocessing, the image will be sent to Google-Tesseract. The tesseract will extract all text from the image and send it to the network.



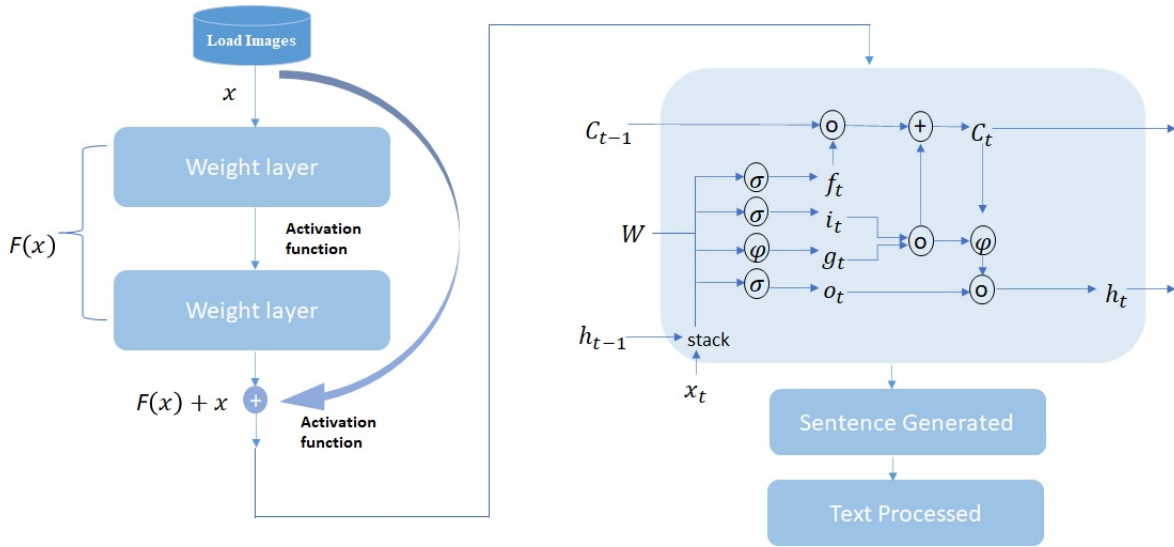


Figure 5: Resnet-LSTM approach for Text Extractions.

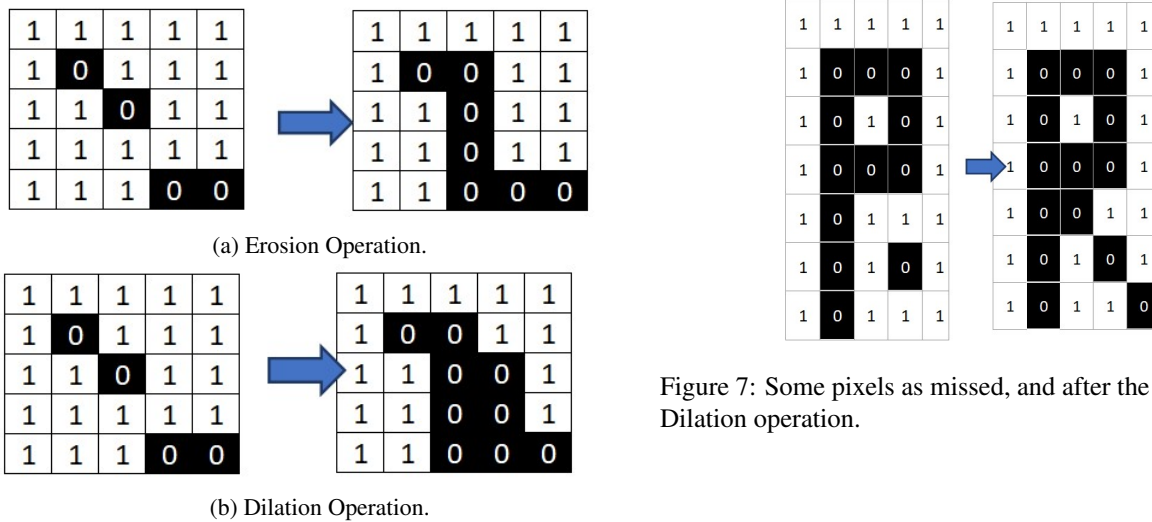


Figure 6: Pixel restoration using Erosion and Dilation operations.

## 4 Implementation and result analysis

### 4.1 Blockchain network setup

To realize the proposed architecture, hyperledger fabric and Sandbox are utilized. Hyperledger is authentication and distributed ledger-based platform. It is an open-source technology used to implement different smart contracts with constraints and logic over the network for applications. The smart contracts are implemented over the network using the sandbox module. In Sandbox, the participants are known, and the Blockchain is in the permissioned consortium mode, making it a secure and trusted Blockchain. The proposed architecture is not limited to

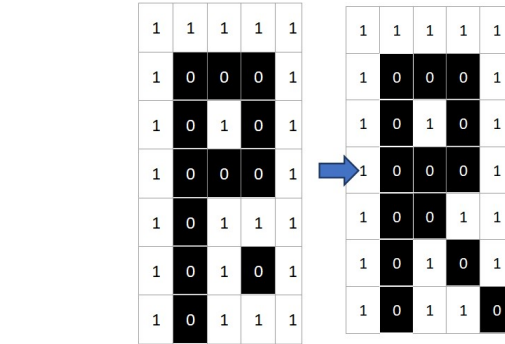


Figure 7: Some pixels as missed, and after the Erosion and Dilation operation.

the healthcare domain. Programming languages such as Node.js, Java, Go, etc., are used for contract and business network development. Docker is used for setting up and initialization when working with hyperledger fabric and composer. Docker is an operating system-level container used by developers, system administrators, etc., for creating, deploying, and running business networks or hyperledger-based applications in a container, enabling the dependencies and functionalities to be packaged together. The hyperledger fabric and composer network can run inside a container using Docker.

In our simulation phase, we used a network model of 3 organizations with 2 peers each and one Orderer. The experiment is carried out with basic writing transactions at various rates, with 1000 transactions per round at 50, 100, 150, 200, and 250 transactions per second. The experiment is done for 1 Org 1 Peer, 2 Org 2 Peer, and 3 Org 3 Peer with different performances of transactions. The results are calculated over five rounds, with each round consisting of 1000 transactions at various transaction rates per second

---

**Algorithm 2:** Hospital, Pharmacy, Patho Lab and Insurance firm Enrollment.
 

---

**Input:** Enrollment Certificate ( $E_C$ ) from Certification Authority ( $C_A$ )

**Output:** Access to all nodes  $H_N, P_N, L_N, I_N$

**Initialization:**  $N_{Admin}$  should be valid node.  
 $N_{Admin}$  can Read/Write/Update/Remove participants  $H_N, P_N, L_N, I_N$  ;

**while** (*true*) **do**

**if**  $H_N$  is valid **and**  $H_{TIN}$  is valid **then**

Add Node ( $B_{NT}, H_N$ )

Grant Access( $H_N$ )

**else**

Invalid( $H_N$ )

**end**

**if**  $P_N$  is valid **and**  $P_{TIN}$  is valid **then**

Add Node ( $B_{NT}, P_N$ )

Grant Access( $P_N$ )

**else**

Invalid( $H_N$ )

**end**

**if**  $L_N$  is valid **and**  $L_{TIN}$  is valid **then**

Add Node ( $B_{NT}, L_N$ )

Grant Access( $L_N$ )

**else**

Invalid( $L_N$ )

**end**

**if**  $I_N$  is valid **and**  $I_{TIN}$  is valid **then**

Add Node ( $B_{NT}, I_N$ )

Grant Access( $I_N$ )

**else**

Invalid( $I_N$ )

**end**

**end**

bool  $chk \leftarrow (0 : \text{malicious}, 1 : \text{genuine})$

**if**  $!(\text{behaviour}(chk))$  **then**

Remove Update ( $H_N, P_N, L_N, I_N$ )

**else**

Add Update ( $H_N, P_N, L_N, I_N$ )

**end**

---

(tps).

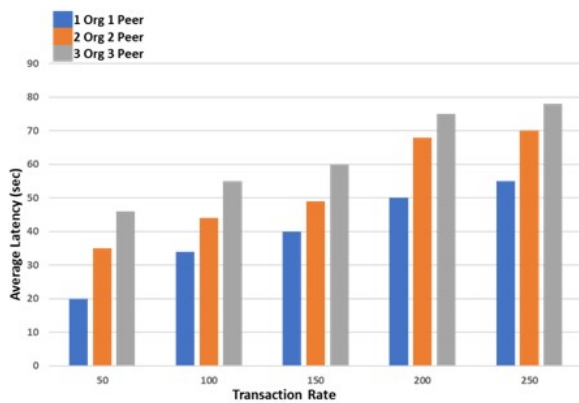
The graphs in figure 8a and 8b highlight the average latency and throughput for varying transaction rates along with the number of transactions completed per minute per three network models. The network model using 1 Org and 1 Peer had the lowest average latency and the highest throughput per transaction rate while completing the highest number of tasks per minute. In contrast, the network model using 3 Org and 3 Peer had the highest average latency and the lowest throughput per transaction rate while completing the lowest number of tasks per minute. The network model using 2 Org and 2 Peer fell in between the above results. (i.e  $3org3peer > 2org2peer > 1org1peer$ ). So, we can conclude from figure 8a, that latency increased as the system scaled up with more organizations and more peers. Throughput of 1 org 1 peer is measured to be highest of 190 whereas it keeps decreasing with the number of organization and peer increases. For 2 org 2 peer, it was found to be 182, and the same for 3 org 3 peer was 180 in Figure 8b. Figure 8c shows successfully completed transactions per minute. 1 org 1 peer completed 5000 transactions in around 4 minutes whereas the 2 org 2 peers completed 4500 transactions and 3 org 3 peers completed 4000 transactions at the same time. As a result, transaction time has been observed to increase in perfect sync with the organization's and peers' growth. Figure 8d on the other hand, highlights the CPU consumption per network model. For different rates of transactions, the network models resulted in varying average CPU usage. We observed that among all peers, peer1.org1.example.com touched the highest CPU utilization at a transaction rate of 200 per sec. whereas peer0.org1.example.com recorded the lowest CPU utilization at a transaction rate of 100 per sec. Table 2, details other resource consumption parameters. With these experiment results, we move forward to our next set of experiments related to data extraction from prescriptions.

## 4.2 Handwritten prescription data extraction- training and validation

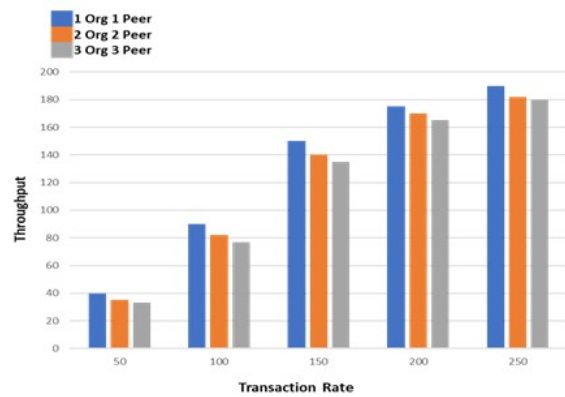
In regions dominated by paper prescription usage, providing the ability to transfer vast amounts of existing data into the digital space is essential. Allowing users to upload prescriptions simplifies the transition to an electronic system and populates the system with useful patient data given in Figure 9a, 9b and 10. The following experiments are related to training and validation for data extraction from handwritten prescriptions. The first step of training requires the number of inputs, hidden layers, and output layers. Twenty handwritten prescriptions of different classes are taken, including numeric characters, alphabetical characters, spaces, and punctuation. To improve image accuracy or legibility, the prescription image may be taken by section by users, causing the model to be confused and less accurate. To overcome this issue, images are converted into small segments. For analysis of images 64x64 pixels in size, 16 feature vectors are extracted from the feature

Abbreviation	Explanation
$P_{Rec}$	Patient’s Record
$B_{NT}$	Blockchain Network
$N_{Admin}$	Admin Node
$P_N, H_N, L_N, I_N$	Pharmacy, Hospital, Patho lab, Insurance firm nodes respectively
$P_{TIN}, H_{TIN}, L_{TIN}, I_{TIN}$	Pharmacy, Hospital, Patho lab, Insurance firm tax Id’s respectively

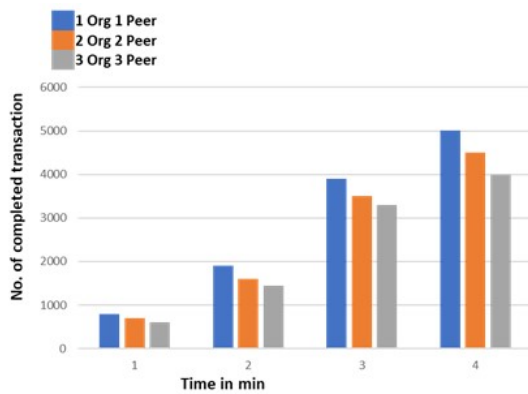
Table 1: Abbreviation used in the algorithm and its explanations.



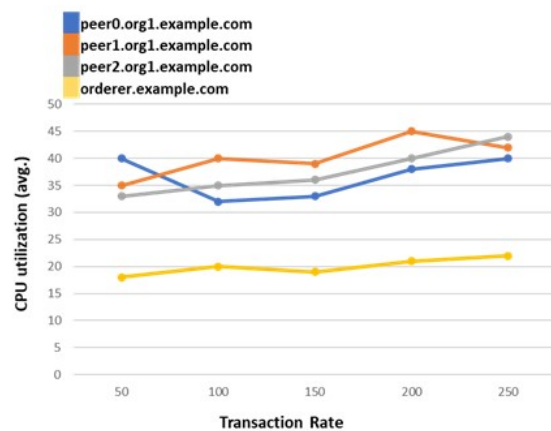
(a) Average Latency with Varying Transaction Rate.



(b) Throughput with Varying Transaction Rate.



(c) No. of Completed Transactions with Time.



(d) Resource consumption.

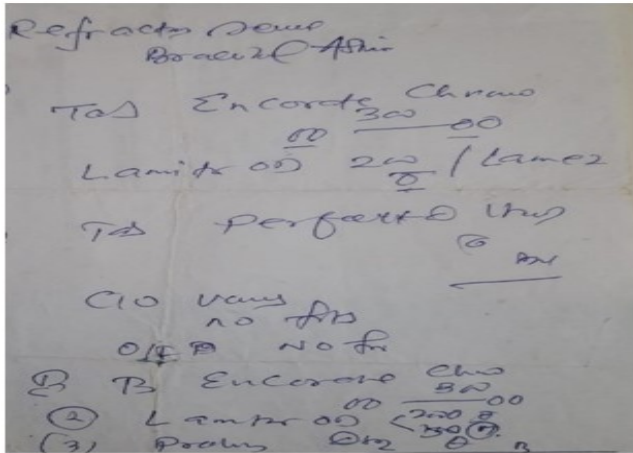
Figure 8: Measurement of different performance parameters. The results are calculated over five rounds, with each round consisting of 1000 transactions at various transaction rates per second (tps).

Type	Name	CPU(avg)	Memory(avg)	Traffic-In	Traffic-out	disc Write
Docker	peer0.org1.example.com	36.6	284.5MB	10.4MB	4.5MB	4.2MB
Docker	peer0.org2.example.com	28.4	280.0MB	10.5MB	5.6MB	4.2MB
Docker	peer0.org3.example.com	25.1	275.5MB	9.8MB	9.8MB	4.2MB
Docker	Orderer.example.com	2.34	50.0MB	2.5MB	1.2MB	1.2MB

Table 2: Resource consumption of various parameters.

map. The feature vectors are produced by the convolution layer and are extracted from each sliding. The model is

trained by 11,000 samples, with the epoch starting from 0 to 10000. The learning rate is 0.0005 when the size of the batch is 10. For this experiment, 2000 images are validated.



(a) Sample handwritten Prescription.

```
(base) debasish@ThinkPad-Yoga-460:~/NLP/Dustbin$ python Code.py
Refraction new
Br
Tas Encords Chrjkl
Lamitor on 200
Tas perfect
bjhvsdfgvhnfvjhj
O/F No For
B Encorete
Lamtor on 200 sjfgh
300
Prosdfbngngjf jhf df
(base) debasish@ThinkPad-Yoga-460:~/NLP/Dustbin$
```

(b) Sample output.

Figure 9: Handwritten prescription and sample output.

	CNN-LSTM	ResNet-LSTM
Train	89.1	90.3
Test	81.6	88.3

Table 3: Model Accuracy for Handwritten Data extraction approach.

According to the above results, the training accuracy was improved after the 30th epoch. Finally, an 88.3% accuracy result was accomplished using Test Dataset given in table 3. The generate model will take an input and generate a text output which is pre-processed by string operations. After the string operations are completed, the produced output will be sent to the network.

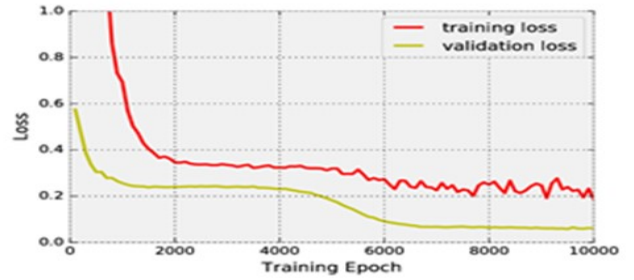


Figure 10: Training graph sample of Resnet-LSTM with output.

### 4.3 Printed prescription data extraction-training and validation

The Tesseract-OCR is a pre-trained model created by Google.

For improved image processing dilation, erosion methods are applied to Otsu’s thresholding. This approach also provides better accuracy as mentioned by the Google tesseract research. [16]. In our case, the test data is resulting in 99% accuracy. The illustration of the image is shown in 11.

*Dr Sanghamitra Kanungo*

Patient Name: SANBHAV DA Hospital : Kar Viaison Eye 2  
 Age : 9 Patient Id: KV20325  
 Gender : Male Regd. Date: 13-Jul-2019 1:40 PM

(a) Printed Prescription.

```
Dr Sanghamitra Kanungo
Patient Name SANBHAV DA Hospital 2 Kar Viaion Bye
age 9 Patient ID Kv20325
Gender Mate Regd. Date 13-gul-2019 1540 #4
```

(b) Sample output.

Figure 11: Output of Tesseract OCR text extraction.

## 5 Conclusion and future work

EHR systems will be of considerable importance to advance the digital medical space of developing regions such as the Indian subcontinent. With the advancement of Blockchain technology, its potential has been recognized to significantly impact the future of EHR systems due to the superiority of Blockchain-based systems over traditional systems and paper-based record keeping. Blockchain-based EHR systems improve security, efficiency, and cost, making it an excellent option for the Indian subcontinent. In this paper, we have highlighted some common

issues that have arisen due to the lack of communication and tracking infrastructure, such as the influx of counterfeit drugs, dependability on handwritten prescriptions, and lack of integration between healthcare and insurance systems. We have discussed solutions to these issues using Blockchain, NLP, Hyperledger Fabric and Docker Containers, etc. The proposed scalable system will allow integration of existing EHR and healthcare databases, national identification, cloud technology to store large files with encryption, and a mobile app-based interface to convert paper prescriptions to text using OCR and deep learning techniques, then to bring old paper-based medical records into the new system. Our future research will include addressing implementation challenges at the grassroots level and also collect more samples for training, to increase the accuracy during converting the handwritten prescriptions into text.

## References

- [1] R. Achkar, K. Ghayad, R. Haidar, S. Saleh, and R. Al Hajj. Medical handwritten prescription recognition using crnn. In *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5. IEEE, 2019. <https://doi.org/10.1109/CITS.2019.8862004>.
- [2] A. Baker. *Crossing the quality chasm: a new health system for the 21st century*, volume 323. British Medical Journal Publishing Group, 2001. <https://doi.org/10.1136/bmj.323.7322.1192>.
- [3] P. K. Bharimalla, S. Praharaj, and S. R. Dash. Ann based block chain security threat mechanism. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8 (10), 2019. <https://doi.org/10.35940/ijitee.J9442.0881019>.
- [4] P. Bhattacharya, S. Tanwar, U. Bodke, S. Tyagi, and N. Kumar. Bindaas: Blockchain-based deep-learning as-a-service in healthcare 4.0 applications. *IEEE Transactions on Network Science and Engineering*, 2019. <https://doi.org/10.1109/TNSE.2019.2961932>.
- [5] M. Bhavin, S. Tanwar, N. Sharma, S. Tyagi, and N. Kumar. Blockchain and quantum blind signature-based hybrid scheme for healthcare 5.0 applications. *Journal of Information Security and Applications*, 56: 102673, 2021. <https://doi.org/10.1016/j.jisa.2020.102673>.
- [6] L. Chen, W.-K. Lee, C.-C. Chang, K.-K. R. Choo, and N. Zhang. Blockchain based searchable encryption for electronic health record sharing. *Future Generation Computer Systems*, 95:420–429, 2019. <https://doi.org/10.1016/j.future.2019.01.018>.
- [7] B. Devkota and A. Devkota. Electronic health records: advantages of use and barriers to adoption. *Health Renaissance*, 11(3):181–184, 2013. <https://doi.org/10.3126/hren.v11i3.9629>.
- [8] A. Dubovitskaya, F. Baig, Z. Xu, R. Shukla, P. S. Zambani, A. Swaminathan, M. M. Jahangir, K. Chowdhry, R. Lachhani, and N. Idnani. Actionehr: Patient-centric blockchain-based electronic health record data management for cancer care. *Journal of medical Internet research*, 22(8):e13598, 2020. <https://www.jmir.org/2020/8/e13598/>.
- [9] J. Fu, N. Wang, and Y. Cai. Privacy-preserving in healthcare blockchain systems based on lightweight message sharing. *Sensors*, 20(7):1898, 2020. <https://doi.org/10.3390/s20071898>.
- [10] G. Gavrilov, O. Simov, and V. Trajkovik. Blockchain-based model for authentication, authorization, and immutability of healthcare data in the referrals process. 2020. <http://hdl.handle.net/20.500.12188/8179>.
- [11] C. Kombe, A. Sam, M. Ally, and A. Finne. Blockchain technology in sub-saharan africa: Where does it fit in healthcare systems: A case of tanzania. *Journal of Health Informatics in Developing Countries*, 13(2), 2019. <https://www.jhidc.org/index.php/jhidc/article/view/24>.
- [12] D. S. Maitra, U. Bhattacharya, and S. K. Parui. Cnn based common approach to handwritten character recognition of multiple scripts. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1021–1025. IEEE, 2015. <https://doi.org/10.1109/ICDAR.2015.7333916>.
- [13] B. Mounia and C. Habiba. Big data privacy in healthcare moroccan context. *Procedia Computer Science*, 63:575–580, 2015. <https://doi.org/10.1016/j.procs.2015.08.387>.
- [14] R. Shanthapriya and V. Vaithianathan. Block-healthnet: security based healthcare system using block-chain technology. *Security Journal*, pages 1–19, 2020. <https://doi.org/10.1057/s41284-020-00265-z>.
- [15] A. P. Singh, N. R. Pradhan, S. Agnihotri, N. Jhanjhi, S. Verma, U. Ghosh, D. Roy, et al. A novel patient-centric architectural framework for blockchain-enabled healthcare applications. *IEEE Transactions on Industrial Informatics*, 2020. <https://doi.org/10.1109/TII.2020.3037889>.
- [16] A. P. Tafti, A. Baghaie, M. Assefi, H. R. Arabnia, Z. Yu, and P. Peissig. Ocr as a service: an experimental evaluation of google docs ocr, tesseract, abbyy finereader, and transym. In *International*

- Symposium on Visual Computing*, pages 735–746. Springer, 2016. [https://doi.org/10.1007/978-3-319-50835-1\\_66](https://doi.org/10.1007/978-3-319-50835-1_66).
- [17] S. Tanwar, K. Parekh, and R. Evans. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications*, 50:102407, 2020. <https://doi.org/10.1016/j.jisa.2019.102407>.
- [18] M. Thakkar and D. C. Davis. Risks, barriers, and benefits of ehr systems: a comparative study based on size of hospital. *Perspectives in Health Information Management/AHIMA, American Health Information Management Association*, 3, 2006. <https://pubmed.ncbi.nlm.nih.gov/18066363/>.
- [19] D. Thompson, F. Velasco, D. Classen, and R. J. Raddemann. Reducing clinical costs with an ehr: investments in performance management are essential to realizing the full benefits of an ehr system—including reduced costs and improved quality of care. *Healthcare Financial Management*, 64(10):106–112, 2010. <https://link.gale.com/apps/doc/A243277528/AONE?u=anon~b7bb66fd&sid=googleScholar&xid=4b596be9>.
- [20] V. Varadharajan, D. Bansal, S. J. Nair, et al. Blockchain reinventing the healthcare industry: Use cases and applications. In *Industry Use Cases on Blockchain Technology Applications in IoT and the Financial Sector*, pages 309–328. IGI Global, 2021. <https://doi.org/10.4018/978-1-7998-6650-3.ch013>.

# An Intelligent Decision Support System For Recruitment: Resumes Screening and Applicants Ranking

Arwa Najjar

Information Technology College, Hebron University, Hebron, Palestine

E-mail: ar1993wa@gmail.com

Belal Amro

Information Technology College, Hebron University, Hebron, Palestine

E-mail: Bilala@hebron.edu

Mário Macedo

Sciences and Technologies of Information and Communication College, Atlântica University, Lisbon, Portugal

E-mail: mariojcmacedo@gmail.com

**Keywords:** recruitment, Decision Support System (DSS), Intelligent Decision Support System (IDSS), Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP).

**Received:** November 7, 2020

*The task of finding the best job candidates among a set of applicants is both time and resource-consuming, especially when there are lots of applications. In this concern, the development of a decision support system represents a promising solution to support recruiters and facilitate their job. In this paper, we present an intelligent decision support system named I-Recruiter, that ranks applicants according to the semantic similarity between their resumes and job descriptions; the ranking process is based on machine learning and natural language processing techniques. I-Recruiter is composed of three sequentially connected blocks namely 1) Training block: which is responsible for training the model from a set of resumes, 2) Matching block: that is responsible for matching the resumes to the corresponding job description, and 3) Extracting block: that is responsible for extracting the top n ranked candidates. Experimental results for accuracy and performance showed that I-recruiter is capable of doing the job with high confidence and excellent performance.*

*Povzetek: Predlagan je inteligentni sistem za podporo odločanju (IDSS) za pregledovanje in razvrščanje življenjepisov prosilcev na podlagi strojnega učenja in obdelave naravnih jezikov.*

## 1 Introduction

Organizations always seek to hire employees who perfectly suit the job. Improper selection decisions for a new employee often have costly impacts on the work. Hence, Persons who stand behind the selection decision face an arduous task of selecting the most appropriate person from several applicants.

Recruitment is the process of searching, attracting, and hiring qualified applicants for employment in an organization [1]. Figure 1 presents an overview of the key steps of the recruitment process.

A recruitment process starts with the advertising of an available job position. This is carried out using diverse advertising channels such as websites, newspapers, and others. Job seekers who are interested in that job will apply for the job by creating their profiles using a designated

online form or uploading their resumes through the organization's website. Received applications are then screened to find out the suitable candidates to interview.

Screeners firstly should understand the requirement for the job. After that, they look through each of the submitted applications and reject applicants who do not meet the requirements. Finally, they find the best applicant who matches the job by comparing resumes with the job profile. The top few candidates listed during the screening stage will go along advanced stages in the process of evaluation, like interviews, written tests, and group discussions. The feedback received from the evaluation processes is used to make the final hiring decision. The candidate who passes the interview stage will be offered the position [2].

Human resources (HR) staff need to spend a significant amount of time going through applications in order to identify the few candidates who are truly qualified for the position. Automated systems can scan resumes for job compatibility, reducing efficiently HR's time spent

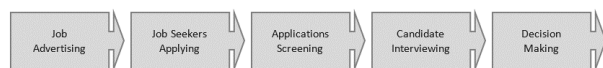


Figure 1: Overview of the recruitment process.

analyzing documents and performing the job with high accuracy as well [3].

A Decision Support System (DSS) is an interactive computer system that helps decision-makers to use data, models, and knowledge in solving structured, semi-structured, or unstructured problems. Any DSS employs Artificial Intelligence (AI) techniques to generate decision alternatives called Intelligent Decision Support System (IDSS) [4]. AI is the development of machines that work and react as though they were intelligent [5]. The development of Decision Support Systems (DSS) is a promising solution for the candidate selection process for a job position in terms of time and effort. As organizations today receive a large number of resumes each time they advertise a job. The needed time and effort for screening is directly proportional to the number of applicants.

This paper presents an IDSS called I-Recruiter for applicants resumes screening to find the best job match ones in the Information Technology sector (IT). I-Recruiter ranks applicants according to the semantic similarity between the resume and job description. Then, it presents the top-ranked candidates' details to go ahead with the recruitment stages. System functions build on the basis of machine learning and natural language processing.

The rest of the paper is organized as follows. In section 2, we will present the related works. The system overview is explained in section 3. I-Recruiter implementation and results are discussed in section 4. In the end, a conclusion and future work is provided in section 5.

## 2 Related works

This work is related to two disciplines of Artificial Intelligence (AI). First, Machine Learning (ML) which is the discipline of giving programs the ability to learn and adapt. Here data represent the experiences where ML models derived. These models help in capturing complicated hidden patterns of new data [6]. The second discipline in AI is Natural Language Processing (NLP) which is the discipline of processing spoken or written forms of free text used by humans with the use of computational methods [7]

For the ML discipline, there are different approaches used approaches namely 1) supervised learning, 2) unsupervised learning, 3) reinforcement learning, and 4) deep learning [8]. In supervised learning, patterns are found from data with labeled features that define their meaning. It is used for weather forecasting. Unsupervised learning is more suitable for unlabeled data. As the data that comes from social media applications. Unlike previous approaches, reinforcement learning depends on trial and error and not on a set of data for training. This kind of learning can be used for training robots. The deep learning main concept relies on the base of incorporating neural networks in consecutive layers for learning. This approach is most suitable for the training of unlabeled and unstructured data in the cases of image recognition, speech, and computer vision [8].

NLP tasks include 1) Sentence Boundary Detection, 2) Tokenization, 3) Part-Of-Speech Assignment To Individual Words (POS Tagging), 4) Morphological Decomposition Of Compound Words, 5) Shallow Parsing (Chunking), 6) Problem-Specific Segmentation, Spelling/Grammatical Error Identification And Recovery, 7) Named Entity Recognition (NER), 8) Word Sense Disambiguation (WSD), 9) Negation And Uncertainty Identification, 10) Relationship Extraction, 11) Temporal Inferences/Relationship Extraction, And 12) Information Extraction (IE) [9].

Over the last few years, deep learning-based NLP attained outstanding results on various NLP tasks. This was achieved via the success of word embeddings and deep learning methods [10]. Word Embedding (WE) is a numerical representation of words, usually as vectors [11]. WE training can be done in a variety of ways, including Word2vec, FastText, and BERT [12]. Word2Vec is the most used form of WEs. The Word2Vec takes text corpus as input and produces word vectors as output. The generation of WE with Word2Vec can be based on two types of models 1) the Continuous Bag Of Words (CBOW) model and 2) the Skip-Gram model [13], [14]. In the CBOW model, a word is predicted based on surrounding words. While in the Skip-Gram model, surrounding words are predicted based on a given word [13]. The architecture of these models is shown in Figure 2.

The main usefulness of WEs is detecting the similarity between words [10]. Measuring similarity between vectors is possible, using term-based similarity measures such as Block Distance, Cosine Similarity, Dice's Coefficient, Euclidean Distance, Jaccard Similarity, Matching Coefficient, and Overlap Coefficient [15].

[16] proposed a resume ranking and recommendation system named Smart Applicant Ranker, which is designed for IT companies to guide them in their recruiting process. This system used Ontology to find and classify the implicit and explicit linkages between the candidate models and the job requirement model. Smart Applicant Ranker architecture is composed of 3 modules: 1) information extraction, 2) candidate search, and 3) candidate ranking algorithms. The information extraction module is responsible for reading the resume information,

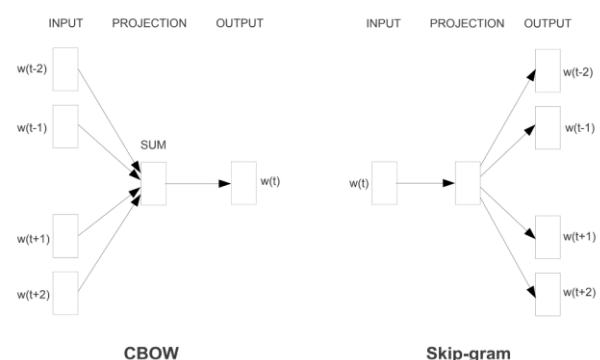


Figure 2: Word2Vec models architecture [13].



constructing an ontology model for a resume, and finally saving the information in the database. Then, the candidate search module provides the resumes ranked by their relative score after calculating the similarity value between the selected resume models and the supplied job requirements. In the last module, the nominated candidates will be evaluated using two fundamental algorithms that will assign rank points based on their 1) educational credentials and 2) skills and work experience.

In [17], the authors proposed a recommendation system that is based on the employer's inquiry to recommend relevant resumes. The system was mainly based on the Vector Space Model (VSM). Where two databases are used to store the terms to be retrieved from the documents (skill DB and candidate DB). The vocabulary is then constructed using the corpus's unique terms. The documents are then represented as vectors using the Term Frequency (TF) approach, and cosine similarity is used to calculate document similarity. Finally, the document that has the highest similarity value is recommended.

[18] proposed a resume classifier application in the IT sector. The application classifies a candidate profile to the best match domain based on the information included in the resume and allocation of a project for the candidate in a particular domain. This application employs the ensemble learning based voting classifier, which consists of five individual classifiers Naïve Bayes, Multinomial Naïve Bayes, Linear SVC (Support Vector Classifier), Bernoulli Naïve Bayes, and Logistic Regression. The architecture is composed of 2 modules: 1) natural language processing pipeline and 2) classification module. The natural language processing pipeline is responsible for removing extraneous information from resumes and providing only the relevant data in the form of tokens. While the classification module analyzes the list of tokens to classify the resume into the appropriate domain.

[19] proposed a resume matching system. The parser system is composed of 4 main phases: 1) text segmentation, 2) named entity recognition, 3) text normalization, and 4) Co-reference merging and conflict resolution. In the first phase, the extracted text is separated into segments of similar information based on attributes such as Name, Phone, and so on. Next, the texts are categorized into named entities. The text normalization process guarantees that specified entities are transmuted to make them consistent and reliable and that abbreviations are enlarged with the help of a reference library. In the last phase, Co-reference resolution, a sort of textual or syntactic-semantic connection in which two or more nominal groups name the same object, is applied to the parsed resume. The outputs of the parser system then passed through a weighting task and matching process based on the firefly algorithm.

[20] proposed a system for resumes classification and matching to a specific job position. As an output of this system, the top ten candidates are selected from a set of applicants. This system first classifies resumes into different categories using Random Forest (RF), Multinomial Naïve Bayes (NB), Logistic Regression (LR), and Linear Support Vector Classifier (Linear SVM)

Article	Basic Methods	Accuracy	Limitations
[16]	Ontology	83%	The proposed solution works only with structured data.
[17]	NLP Vector space model	Not mentioned	The model only considers the skills characteristic and neglects other factors such as education and work experience, besides, accuracy was not reported so we can not decide whether it is usable or not.
[18]	NLP Ensemble learning	91%	The model merely matches a candidate's profile to an appropriate domain, not to a specific employment position.
[19]	NLP Firefly algorithm	94% "only for a part of the whole system (parser)"	The model concentrates only on the job seeker's educational qualifications and skills.
[20]	NLP ML	79%	The main issue faced with this system is that some important data is lost because of text summarization, also it does not support some real-world resumes format such as PDF.
[21]	NLP Vector space model	Not mentioned	We can't say whether it's usable or not because the accuracy hasn't been reported.
[22]	Ontology	Not mentioned	The proposed system focused only on the CyberSecurity field. Also, other factors such as experience, education, and so on are not taken into account. As well accuracy hasn't been reported.

Table 1: Comparison of similar solutions.

models. Then, applies the content-based recommendation using cosine similarity and k-Nearest Neighbors (k-NN) algorithm for ranking resumes.

[21] proposed an automated resumes screening system, which works in two phases. First, NLP is used to extract all relevant candidate information such as skills, work experience, education, certifications, and so on from the unstructured text in resumes. Then, resumes were ranked according to how well their content matched the job description using the Vector Space Model, where the documents are represented as vectors, and then similarity measurements such as cosine similarity are used to determine which group of resumes is the best fit for the job. Finally, a ranked list of applicants is generated.

In [22], an ontology-based recommender system was presented for analyzing and assessing information while taking into consideration the changing demands of the firm and the talents of the job applicant. It is composed of two main parts including ontologies construction and matching process. Where the construction of ontologies is done through three phases 1) Job requirements are represented as ontologies, 2) the system collects all of the information from job seekers' profiles and creates ontology models for them, and 3) two different ontologies for skills, IT skills, and Cybersecurity skills, are created. The matching process then uses a matching engine to compute matching scores using pre-generated ontologies and a set of matching rules, taking as input a job description and a number of job seeker's profiles to be matched.

A summary of the related work and comparison among several strategies is provided in Table 1. As seen from the table, [16] [17], [19], [21], and [22] used ML in its solution architecture which is different from what was used by I-recruiter. Besides, I-recruiter intersects with [17], [18], [19] and [20] in its support of unstructured resumes. In general, I-Recruiter uses different techniques from other proposed works such as word embeddings and supports continual learning to adapt to data changing. It also enables users to specify the number of ranks which makes it more flexible as well as its supports to real-world resumes format such as PDF.

### 3 System overview

All the time, companies have been spent a lot of time and effort on traditional recruiting, especially in cases with a large number of received resumes. With the support of technology power, recruitment can be more efficient with fewer resources needed. As an attempt to support and facilitate the recruitment process, an IDSS called I-Recruiter was designed to speed up the screening step of recruitment. I-Recruiter automatically finds the top-ranked candidates based on the degree of semantic similarity between the job description and applicants' resumes. The architecture of the I-Recruiter is shown in Figure 2. The system consists of three main building blocks namely 1) training, 2) matching, 3) and extracting.

In the next sub-sections, we will explain how does each block of the I-Recruiter work.

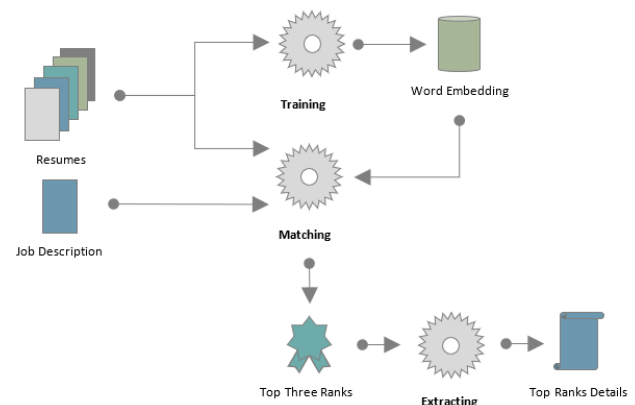


Figure 3: I-Recruiter architecture.



Figure 4: Levels of the training process.

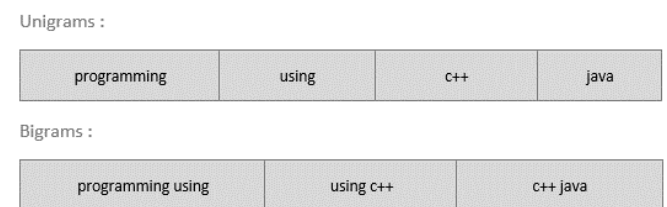


Figure 5: Unigram and bigrams of the cleaned version of the sentence "Programming using C++ and Java".

#### 3.1 Training block

This block is responsible for training the domain WE from a set of resumes. In this subsystem, the Skip-Gram model is used. According to [13] skip-gram model is more efficient than CBOW as rare words or phrases are well presented. Also, this subsystem is responsible for updating the basic WE with any new inputted resume, this assists the system to continuously develop knowledge. The generated WE models from this block will be used later in the matching block. Both training and updating processes are done on four levels as shown in Figure 4.

- Level 1 — text extraction: resume files are being read and text is extracted from them.
- Level 2 — unigrams generation: a list of one word generated from resumes text in two steps. First, the text is pre-processed with the removal of unnecessary parts of the text, such as links, symbols, and stop words. Also, by dividing it into tokens with NLP tokenization and returning words to their basic form with NLP lemmatization. Second, words for training are extracted from the text using POS tagging. In which unannotated words in natural language are labeled with Parts-Of-Speech labels such as verbs, nouns, adverbs, adjectives, etc. [7]. Figure 5 shows an example of unigrams.
- Level 3 — bigrams generation: a list of word pairs is generated from resumes text in two steps. First, bigrams

are generated and scored with the use of the Natural Language Toolkit (NLTK), which is a suite of libraries for language processing [23]. Second, top-scored bigrams are cleaned to remove any noisy pairs that do not harmonize together or are not related to the job. This cleaning process employs POS tagging and NER techniques for defining unwanted words. NER recognizes the named entities occurring in the text such as persons, organizations, locations, dates, etc. [7]. Cleaning is also based on some frequent words that appear in resumes and must be ignored like name, contact, and so on. Other IT-related words must not be ignored like C++, 3Ds, and so on. Figure 5 shows an example of bigrams.

- Level 4 — training word embeddings: models are trained at the last level from previously generated lists of bigrams and unigrams using the Word2Vec technique.

### 3.2 Matching block

This block is responsible for matching the text of the job description with the text of each applicant's resume. It depends on pre-trained WEs to create vectors for all inputted resumes and the job description. Figure 6 shows a sample of the vector representation of words. After that, the average of generated vectors is computed as the vector's average estimation leads to a meaningful representation of longer pieces of text [24]. Lastly, the semantic similarity degree is calculated with the cosine similarity method. Cosine similarity refers to the measure of similarity between two vectors, where vectors represent the compared documents and the cosine degree between these vectors represents similarity degree [25]. The similarity is calculated as shown in the following equation where A and B are vectors and  $\emptyset$  represent the angle between them:

$$\begin{aligned} \text{Similarity}(\vec{A}, \vec{B}) &= \cos \emptyset = \frac{A \cdot B}{\|A\| \|B\|} \\ &= \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \end{aligned} \quad (1) [26]$$

The top ranks of the resumes are for the top highest similarity degrees. These top-ranked resumes will be transferred to the next block for data extraction.

### 3.3 Extracting block

This block is responsible for extracting the top-ranked candidates' basic information that including candidate name, phone number, and email. NLP gazetteer and pattern matching approaches are used in this stage. For extracting emails and phone numbers, the pattern matching approach is used. Where extraction patterns are defined using Regular Expressions (RE). The result is then matched with a given input text and the matched text will be extracted [7]. I-Recruiter is designed to detect any text that matches the email pattern for finding candidates' email addresses. Concerning getting candidates' phone numbers, the text detected whatever matches the Palestine dialing code. Regard extracting names, the gazetteer approach was used. Gazetteer or gazette is a pre-defined list of all possible values of a named entity [7]. Here the

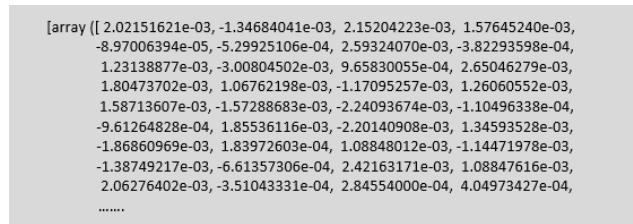


Figure 6: Sample of words vector representation.

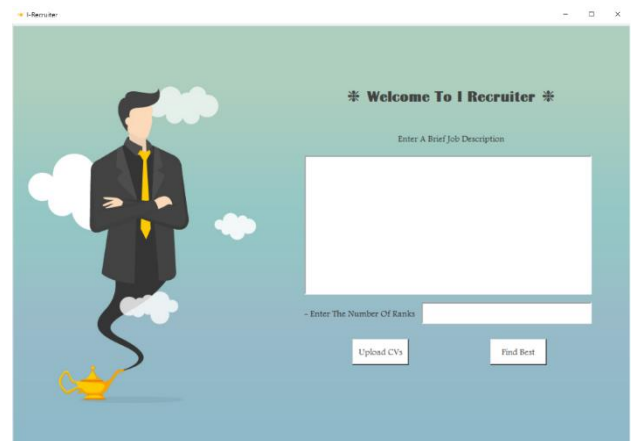


Figure 7: I-Recruiter interface.

system detects entities named 'PERSON' in the process of finding candidates' names.

## 4 Implementation and discussion

In this section, we will explain the implementation of the I-Recruiter prototype, data collection, and I-Recruiter testing as well as a discussion of the results.

### 4.1 Prototype development

A prototype of I-Recruiter has been developed as a desktop application with the use of Python programming language version 3.8 escorted by python artificial intelligence libraries such as Gensim, spaCy, and Natural Language Toolkit (NLTK). Figure 7 presents the I-Recruiter interface.

### 4.2 Data collection

A dataset of 101 unstructured resumes in the domain of information technology (IT) were collected from different resources. All resumes within this dataset are in PDF format.

### 4.3 Test and results

To test I-Recruiter, we designed two experiments, the first aimed to test the performance (elapsed time) where the second aimed to test the accuracy of I-Recruiter results.

#### 4.3.1 Performance of I-Recruiter.

The time required to train models using 101 resume files was calculated. The calculated time has been carried out in one trial. Also, the time required to match 101 resumes to a job position and the time required to extract

information from the top three ranked candidates' resumes were calculated. Four trials with different job descriptions for each trial were used for calculating the required time of both matching and extracting. Table 2 shows the results of these tests.

As shown in Table 1, I-Recruiter took approximately 327.23 seconds for train 101 resumes, an average of 36.34 seconds for matching, and an average of 0.73 seconds for extracting.

There were no reports about total execution time for other related work, however, an average of 37.07 seconds execution time is acceptable and effectively applicable in comparison to the lengthy manual method. According to this, I-Recruiter can work efficiently and will do save time for the human resources department.

### 4.3.2 Accuracy of I-Recruiter.

Following [16] and [18], accuracy was measured by comparing system results to individual ones. In our experiment, I-Recruiter was configured to find out the best 3 candidates among 10 applications for 4 job positions. The same was performed manually by an IT specialist and both results were compared together in Table 3.

I-Recruiter has proved that it can be reliable with an overall accuracy of 100% in candidate selection and 92% in rank order. These degrees of accuracy were calculated by finding the average of the results of all testing trials. The 8% error percentage appears because of the existence of some noisy bigrams that were not removed in the cleaning process. To overcome this issue of wrong ordering, users can specify more than needed ranks. Then, reorder them manually.

In comparison with other proposed solutions and models, I-Recruiter showed excellent performance in terms of execution as well as accuracy. The average accuracy of I-Recruiter is 96% while for [16], [18], and [20] it is 83%, 91%, and 79% respectively. Hence, it is obvious that I-Recruiter outperforms the other models in terms of accuracy and hence is reliable to be used for the purpose it was proposed.

## 5 Conclusion and future work

I-Recruiter is an intelligent decision support system for screening a set of applicants' resumes for a job position and find out the most appropriate candidates. This system is composed of three main building blocks training, matching, and extracting. Domain-trained word embeddings are generated from the training block. While the matching block finds the top candidates based on the semantic similarity between resumes and the job description. Basic information on top-ranked applicants was extracted in the last block. I-Recruiter showed very good results with a high degree of accuracy and a short time of work. We'll work on increasing the system's performance and accuracy in the future, as well as introducing more capabilities like personality analysis from personal pictures.

<b>Testing Device</b>	Processor: intel core i7-6700HQ RAM: 16 GB System Type: 64-Bit			
<b>System Block</b>	<i>Training</i>		<i>Matching</i>	<i>Extracting</i>
<b>Time / Seconds</b>	327.23	<i>Trial 1</i>	37.55	0.69
		<i>Trial 2</i>	35.74	0.86
		<i>Trial 3</i>	36.11	0.43
		<i>Trial 4</i>	35.95	0.94

Table 2: Performance of I-Recruiter.

Job Description	Top Ranks	I-Recruiter Result	Human Result	Selections Correctness	Order Correctness
JD 1	<i>Rank 1</i>	CV1	CV1	100%	67%
	<i>Rank 2</i>	CV8	CV5		
	<i>Rank 3</i>	CV5	CV8		
JD 2	<i>Rank 1</i>	CV2	CV2	100%	100%
	<i>Rank 2</i>	CV8	CV8		
	<i>Rank 3</i>	CV3	CV3		
JD 3	<i>Rank 1</i>	CV8	CV8	100%	100%
	<i>Rank 2</i>	CV6	CV6		
	<i>Rank 3</i>	CV5	CV5		
JD 4	<i>Rank 1</i>	No one matches the job	No one matches the job	100%	100%
	<i>Rank 2</i>				
	<i>Rank 3</i>				

Table 3: Accuracy of I-Recruiter.

## References

- [1] M. B. R. Devi and Dr. Mrs. P. V. Banu, "Introduction to Recruitment," SSRG Int. J. Econ. Manag. Stud. SSRG-IJEMS, vol. 1, no. 2, p. 4, 2014. [Online]. Available: <https://www.internationaljournalssrg.org/IJEMS/2014/Volume1-Issue2/IJEMS-V1I2P102.pdf>
- [2] A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla, "PROSPECT: a system for screening candidates for recruitment," in Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10, Toronto, ON, Canada, 2010, p. 659. doi: <https://doi.org/10.1145/1871437.1871523>.
- [3] M. L. Gusdorf, "Recruitment and Selection: Hiring the Right Person," p. 19, 2008. [Online]. Available: <https://www.shrm.org/certification/educators/Documents/Recruitment%20and%20Selection%20IM.pdf>

- [4] G. Phillips-Wren, “Intelligent Decision Support Systems,” in *Multicriteria Decision Aid and Artificial Intelligence*, M. Doumpos and E. Grigoroudis, Eds. Chichester, UK: John Wiley & Sons, Ltd, 2013, pp. 25–44. doi: <https://doi.org/10.1002/9781118522516.ch2>.
- [5] W. Ertel, *Introduction to Artificial Intelligence*. Cham: Springer International Publishing, 2017. doi: <https://doi.org/10.1007/978-3-319-58487-4>.
- [6] J. Zhang and P. S. Yu, “Machine Learning Overview,” in *Broad Learning Through Fusions*, Cham: Springer International Publishing, 2019, pp. 19–75. doi: [https://doi.org/10.1007/978-3-030-12528-8\\_2](https://doi.org/10.1007/978-3-030-12528-8_2).
- [7] S. Singh, “Natural Language Processing for Information Extraction,” *ArXiv180702383 Cs*, Jul. 2018, Accessed: Sep. 11, 2020. [Online]. Available: <http://arxiv.org/abs/1807.02383>
- [8] J. Hurwitz, *Machine Learning For Dummies®*, IBM Limited Edition. 2018. [Online]. Available: <https://www.ibm.com/downloads/cas/GB8ZMQZ3>
- [9] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, “Natural language processing: an introduction,” *J. Am. Med. Inform. Assoc.*, vol. 18, no. 5, pp. 544–551, Sep. 2011, doi: <https://doi.org/10.1136/amiajnl-2011-000464>.
- [10] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent Trends in Deep Learning Based Natural Language Processing,” *ArXiv170802709 Cs*, Nov. 2018. [Online]. Available: <https://arxiv.org/pdf/1708.02709.pdf>
- [11] A. Mandelbaum and A. Shalev, “Word Embeddings and Their Use In Sentence Classification Tasks,” *ArXiv161008229 Cs*, Oct. 2016, Accessed: Sep. 17, 2020. [Online]. Available: <http://arxiv.org/abs/1610.08229>
- [12] C. Wang, P. Nulty, and D. Lillis, “A Comparative Study on Word Embeddings in Deep Learning for Text Classification,” in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, Seoul Republic of Korea, Dec. 2020, pp. 37–46. doi: <https://doi.org/10.1145/3443279.3443304>.
- [13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *ArXiv13013781 Cs*, Sep. 2013, Accessed: Sep. 11, 2020. [Online]. Available: <http://arxiv.org/abs/1301.3781>
- [14] P. Sitikhu, K. Pahi, P. Thapa, and S. Shakya, “A Comparison of Semantic Similarity Methods for Maximum Human Interpretability,” *ArXiv191009129 Cs*, Oct. 2019, Accessed: Sep. 11, 2020. [Online]. Available: <http://arxiv.org/abs/1910.09129>
- [15] V. M.K and K. K, “A Survey on Similarity Measures in Text Mining,” *Mach. Learn. Appl. Int. J.*, vol. 3, no. 1, pp. 19–28, Mar. 2016, doi: <https://doi.org/10.5121/mlajj.2016.3103>.
- [16] A. Mohamed, W. Bagawathinathan, U. Iqbal, S. Shamrath, and A. Jayakody, “Smart Talents Recruiter - Resume Ranking and Recommendation System,” in *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, Colombo, Sri Lanka, Dec. 2018, pp. 1–5. doi: <https://doi.org/10.1109/ICIAfS.2018.8913392>.
- [17] S. N, S. V, A. S, and S. P, “Validating effective resume based on employer’s interest with recommendation system,” *Int. J. Pure Appl. Math.*, vol. 119, 2018, [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01826687/document>
- [18] S. T. Gopalakrishna and V. Varadharajan, “Automated Tool for Resume Classification Using Semantic Analysis,” *Int. J. Artif. Intell. Appl.*, vol. 10, no. 01, pp. 11–23, Jan. 2019, doi: <https://doi.org/10.5121/ijaia.2019.10102>.
- [19] G. Deepak, V. Teja, and A. Santhanavijayan, “A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm,” *J. Discrete Math. Sci. Cryptogr.*, vol. 23, no. 1, pp. 157–165, Jan. 2020, doi: <https://doi.org/10.1080/09720529.2020.1721879>.
- [20] P. K. Roy, S. S. Chowdhary, and R. Bhatia, “A Machine Learning approach for automation of Resume Recommendation system,” *Procedia Comput. Sci.*, vol. 167, pp. 2318–2327, 2020, doi: <https://doi.org/10.1016/j.procs.2020.03.284>.
- [21] C. Daryani, G. S. Chhabra, H. Patel, I. K. Chhabra, and R. Patel, “An Automated Resume Screening System Using Natural Language Processing And Similarity,” in *Ethics And Information Technology*, Jan. 2020, pp. 99–103. doi: <https://doi.org/10.26480/et.02.2020.99.103>.
- [22] M. Maroun and A. Ivanova, “Ontology-based approach for cybersecurity recruitment,” *Tomsk, Russia*, 2021, p. 070014. doi: <https://doi.org/10.1063/5.0042320>.
- [23] “Natural Language Toolkit, NLTK 3.5 documentation, 2020. <https://www.nltk.org/>
- [24] T. Kenter and M. de Rijke, “Short Text Similarity with Word Embeddings,” in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management - CIKM ’15*, Melbourne, Australia, 2015, pp. 1411–1420. doi: <https://doi.org/10.1145/2806416.2806475>.
- [25] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, “Cosine similarity to determine similarity measure: Study case in online essay assessment,” in *2016 4th International Conference on Cyber and IT Service Management*, Bandung, Indonesia, Apr. 2016, pp. 1–6. doi: <https://doi.org/10.1109/CITSM.2016.7577578>.
- [26] F. Rahutomo, T. Kitasuka, and M. Aritsugi, “Semantic Cosine Similarity,” in *The 7th International Student Conference on Advanced Science and Technology ICAST*, 2012, p. 3. [Online]. Available: [https://www.researchgate.net/profile/Faisal-Rahutomo/publication/262525676\\_Semantic\\_Cosine\\_Similarity/links/0a85e537ee3b675c1e000000/Semantic-Cosine-Similarity.pdf](https://www.researchgate.net/profile/Faisal-Rahutomo/publication/262525676_Semantic_Cosine_Similarity/links/0a85e537ee3b675c1e000000/Semantic-Cosine-Similarity.pdf)



# A Practical Framework for Real Life Webshop Sales Promotion Targeting

Gábor Kőrösi and Tamás Vinkó

University of Szeged, Institute of Informatics, Hungary

E-mail: korosig@inf.u-szeged.hu, tvinko@inf.u-szeged.hu

## Technical paper

**Keywords:** promotion targeting, behavior analysis, hybrid recommendation system

**Received:** February 28, 2020

*In recent years, online marketing has become increasingly extensive and effective. Product recommender systems are often deployed by e-commerce websites to improve user experience and increase sales. To address this, more and more e-commerce started to use machine learning models to predict customers' purchase behaviors. In the scientific literature there are only few real-life studies to date which give solutions for recommendation systems for online advertising. The demand from the owners of such websites is given, however, it is hard for them to choose a method or model to predict from an endless number of options for some specific circumstances. The aim of this paper is to propose a practical guideline as a hybrid approach that predicts customers' purchase behaviors and helps to target advertisement, sales form in user level. To this end, we have designed a robust hybrid model to predict interested sales form based on user behavior within a large e-commerce website. The paper details a real-life practical solution and build a structure that can be used in a large variety of e-commerce systems.*

*Povzetek: Opisan je razvoj modela nakupovanja po spletu z namenom ciljnega oglaševanja.*

## 1 Introduction

One of the most important and dynamically developing areas today is e-commerce and related services. While in a traditional shop tracking customer is difficult (e.g., loyalty card program), a webshop's back-end offers countless solutions to solve this problem. For example, we could use cookies, spent checking, newsletter and product tracking [4, 15, 1]. The main driving force behind this fast evolution is the fact that we can understand and anticipate user behavior better, and we can answer the related questions in real-time. The key goal is to get the highest response from users by spending as little money and time on it as possible, and create customer-oriented services [2]. That is named personalization and targeting [13], where the objective is to find the best matching ads or form of sales promotion to be displayed for each user. The solution is not new, as we could see similar solutions at the first generation webshops, but nowadays the amount of data is much higher than before.

When the task is to efficiently process huge amount of data, it is useful to try and find a solution in those research papers written based on similar task. For example, [26] analyzing clickstream, [15] uses email sending history, while [1] collects user activity to predict user's future behavior. At first glance, the task does not seem to be a difficult one, as using data mining and data science in e-commerce is not new, and there is a huge amount of papers published with the same purpose. These papers refer to a waste amount of machine learning (ML) tools and solutions which are able

to help with this optimization. For instance, classification can predict the occurrence of an event, or regression techniques that can help us to predict the time or amount of money the user will spend on the website. More sophisticated solutions are offered by collaborative filtering or content-based approaches. The repository of toolkits may seem endless, but solving a problem is never the same, and it is seldom enough to use just one tool to solve a problem. Many recent publications have introduced some kind of hybrid solution for this complex problem in which one has to combine and embed simple methods to find a proper model. For example, in [5] we could see a typical hybrid recommendation model that integrates user-based and item-based collaborative filtering, content-based filtering together with contextual information to get rid of the disadvantages of each approach.

Thorough literature review on these subjects can lead to an impression that most of the scientific papers are theoretical model descriptions instead of accurate and practical model descriptions. One could find vague model formulations that make it difficult or impossible to rebuild a presented solution in real life. Along this line of thought, we have concluded that, besides theoretical models, there is a huge demand for publications that document a case study and provide the opportunity for anyone to reproduce it the results on their database. Our goal is to make and document a case study that demonstrates an ML-based recommendation system, which classifies users and provides an individual-level approach for ads form. Based on our literature review we found that a hybrid recommendation system

provides the most accurate solution for that. We combined and embedded various classification and regression models, including Logistic Regression, Random Forest, GBM, and XGBoost to get the most accurate solution.

The rest of the paper describes our approach as follows. Sections 2 and 3 describe the background of the problem. In Section 4 the dataset and the generated features are detailed. The model ensemble is briefly described in Section 5. In Section 7, the importance of features is studied, top features are listed, and our solution is given. Finally, Section 8 concludes the results of the study.

## 2 Background

As data is increasing, more and more companies are demanding high quality solutions from their data scientists. The use of recommendation systems has become a daily concept in product suggestion, product group selection, promotional message content generation which is supported by machine learning techniques. Common examples of applications include the recommendation of movies (e.g., Netflix, Amazon Prime Video), music (e.g., Pandora), videos (e.g., YouTube), news content (e.g., Outbrain) or advertisement (e.g., Google) [25].

In this paper we give a detailed description of a recommendation system which can make user-level marketing letter or offer sales promotion. Note that *recommendation system* is a quite general concept. It could be based on the collaborative filter solution, the content-based method, the classification or regression, and their embedding in different depths and widths. What follows is an outline of what a recommendation system might consist of.

Collaborative filtering (CF) is probably one of the most used and well-known technologies. Behind the basic idea, the solution is that based on users' historical data, the users are put into an  $n$ -th dimensional space which makes possible to then measure the distance between them. In light of this, we could make recommendations based on the data of the users closest to each other [7]. This CF technique proved its power, but on the other hand, a huge amount of work pointed out the disadvantages of it. These are the followings: cold start problem, data sparsity, and scalability [29].

Besides collaborative filtering, the second most popular solution is the content-based method. It is a technique which operates with unique characteristics and behaviors of each customer, and in turn, delivering personalized content for each user, based on their content consumption history across channels. Another interesting way is the community-based method. This approach assumes that the content coming from a user's friends or authoritative users is more likely to be interesting for a user than the rest.

While collaborative filtering and content-based models, used only a static 'user states' we could find many papers which are using uni- or multivariate user event sequences, time-series to build a predictive model. Koehn *et al.* [20]

divided the user event sequence prediction problem into four groups, namely the 'predict the product group', 'classify a sequence', 'predict the outcome of an incomplete session', and 'click-through rate prediction'. In our work we are focusing to predicting the users' interest, which was created based on some initial observations on the users' purchase behavior during the shopping process, meaning that our task is rather similar to the 'predict the product group' task of the recommender systems. Koehn *et al.* [20] summarized the methods of event sequence data preprocessing, highlighting their advantages and disadvantages. One of the most often implemented methods is to create aggregated, cumulated data, which, however, results in data loss and requires manual feature engineering by the domain experts. Another common method is to create sequence segments or sliding a window, where we use only a chunk/fixed-length part of the data. Lastly, there are neural networks and embedding layers, where we can work with partially or completely raw data. In the field of sequence prediction approach, we could find many papers.

Perhaps one of the most promising paper which related to our work is created by Yu *et al.* [28]. They used recurrent neural network on sequenced data to identify web shop users habits and made the next basket recommendation. They applied recurrent layers in the temporal domain and proved their effectiveness for handling the temporal dimension for time series classification. Deep learning based (DLL) solution with time series have proven efficiency in many areas, however, web-shop log data often includes variables that contain mixed continuous and discrete variables. Even these kinds of data can be easily handled by a decision tree-based solution, in neural network this is not so easy. In deep learning based approaches, the discrete-valued sequences must be transformed into the numeric space. Using one-hot encoding might not prove to be overly useful, as it explores the dimensionality of the input feature vector and dramatically increases its sparsity. Inspired by Natural Language Processing, we managed to transform our categorical data into a dense space utilizing embeddings. These methods encode categories as vectors based on contextual similarities and then feed them into the recurrent or convolutional neural network. The embedded vectors are usually trained together with the time-series/sequence model training process [21]. The embedding of discrete-valued sequences was successfully applied in user behavior analysis. For instance, An *et al.* [3] presented their neural user embedding approach which was capable of learning informative user embeddings by using the unlabeled browsing-behavior. Koehn *et al.* [20] proposed their impressive clickstream classification results where they applied RNN architectures and embedding layers. Cheng *et al.* [9] introduced the Wide and Deep feature representation method. In their terminology, Wide representations were one-hot encoded features which could memorize sparse feature coincidences, while Deep representations consisted of dense embeddings which gave generalization power to deep learning systems.



Although content-based and community-based methods have proven their worth in many case studies, in our case it was almost impossible to apply these methods due to the lack of data. Another approach could be the deep learning based solution, but as many paper shows (e.g. [8, 17]) when the dataset is based only short sequences (as our dataset), a traditional ML model can outperform a DLL based model. Based on these paper even a XGboost based classification model or regression would provide a good solution in an optimum prediction system, but unlike simple patterns, things are always more complicated in real life.

To solve the backward of the traditional and DLL based methods, the concept of hybrid or combined systems are becoming more popular in many papers. Bozanta and Kutlu [5] summarized that while each filtering approach has different drawbacks, a hybrid approaches combines the existing approaches and aim to minimize or remove the drawbacks of existing approaches, which may occur when they are used individually. The exact description does not exist for a hybrid solution, but we could certainly use the aforementioned tools at different depths and widths. There are quite many papers proving that a hybrid approach provides a better solution than the single method, see, e.g., in [5, 7, 12]. Thus, we have chosen this solution for our workflow, and we decided to use use such a hybrid model for our system which used both regression and classification method.

Our goal was to solve the problem of predicting the user behaviors about the sales promotion. Similar goals is solved by Martínez *et al.* [23] and Liu *et al.* [22]. They created a model that can predict future customer behavior which based on the set of customer-relevant features that derives from times and values of previous purchases. As our solution, they apply machine learning algorithms including logistic Lasso regression, the extreme learning machine and gradient tree boosting for predicting whether the customer makes a purchase in the upcoming month. Although these two cited papers are very similar to the solution we used, however, unlike them, we tried to create a prediction algorithm not just by using one method but by a (hybrid) combination of them.

### 3 Problem statement

The main objective of this paper is to solve the problem of predicting the purchase behaviors of users who have known the history on an e-commerce website. More closely, we aim at forecasting which ads group or form of sales promotion user will most likely to use based on purchase history and profile information. This form of sales promotion could be: buy two, get one free; price deal; sampling, etc.

Although we did not directly use others' work to design our system, the solution we came up with is strikingly similar to the description of [29]. That is, a predictive system would help in several practical scenarios such as

- build a cold start recommender system, by providing

high-level recommendations to users who connect for the first time to an e-commerce website;

- improve existing product recommendation engines, by providing category-level priors that can guide the recommender system to and domains of interest for the user;
- provide e-commerce companies with tools for targeted email/social media campaigns.

Our paper has two main goals. The first is to explore which information is correlated with the form of sales promotion which the users most likely to use (see in Table 1 for an illustrative example.) Based on this we have built and tested a hybrid model which optimizes a user-level table, in order to propose the form of sales promotion to users that fit the best to their interests and preferences, see Table 2. The second goal is to back-test and document well each critical point of hybrid machine learning algorithms which could be used as a base structure for those who want to replicate our model or build a similar system.

### 4 Datasets

We have used data which has been recorded from a health and beauty webshop. The data has contained near millions of users, from different markets (countries), however, in order to obtain the richest data possible, we have filtered it by the oldest market which includes 230,000 user-profiles and their purchase history. Data consists of seven years of user interaction logs with the webshop. Each event has a user identifier, a timestamp, and an event type. The purchase data contains 5 categories of events: pageview of a product, basket view, buy, ordered timestamp, and delivered timestamp.

There are around 240 different types of products. In the case of a buy or a basket view, we have information about the price and extra details. An average customer has been used the shop two or three times yearly, which leads to very sparse and high dimensional dataset. This is not surprising as it is extremely common in recommender systems [25]. As a solution, there are two obvious ways to reduce the dimensionality of the data: either by marginalizing the time (aggregate pageviews per user over the period) or the product pageviews (aggregate products viewed per time frame) [26]. In this work, we follow both approaches.

As a first step, our solution connected unique events with sessions. We used homogeneous like purchase history only and heterogeneous example clicks, profile data in nature. These events are then cleansed and ordered by their timestamps to form the action chain.

As a next step, we transformed unique events into a feature list (e.g., number of purchases, the distance between two logins, etc.). Beside of the evident data (number of, sum of, mean of purchases), the script accumulated other data such as:

Table 1: Illustration: problem statement as a binary classification.

1st purchase	2nd purchase	3rd purchase	...	<i>n</i> th purchase
		time of prediction	⇒	Likely to buy with <i>sales promotion</i> (user who use more than 50% promotion for buying something)

Table 2: Illustration: problem statement as a recursion; the distribution of sales promotion types.

1st purchase	2nd purchase	3rd purchase	...	<i>n</i> th purchase	
		Time of prediction	⇒	SPTypel	35%
				SPTypel2	25%
				⋮	⋮
				SPTypen	50%

- distance (in time) between first and second, third, etc. actions;
- number of purchases in first, second, etc. months;
- increase or decrease in purchases compared to the previous month by month;
- the reaction times between advertising letters and a purchase.

#### 4.1 Feature engineering

One of the most important steps for better performance of a classifier is to preprocess the data correctly. Besides the regular data cleaning process, we transformed features by scaling each feature to a given range with min-max scaling. As a last preprocessing step, we calculated feature importance with tree-based ensemble method namely `ExtraTreesClassifier` [14]. Based on the obtained results by this method, our model uses only the top 20 features, which significantly increased the accuracy of the results.

## 5 Methodology

In order to handle the popularity-bias, we divided the problem into two subtasks:

- i) predict if a user is sensitive for the sales promotion or not, and
- ii) predict which kind of form of sales promotion is more interested in it.

As a solution, we have created a hybrid model which used both regression and classification method, see Figure 1.

The recommendation model returns two lists. The first list gives information about the users, if they are likely to

use or not any of the forms of sales (the sensitivity for sales promotion). The second list provides us with the data to calculate the probability for every sale (which form of sales likely to use).

For the results we propose a novel hybrid recommendation algorithm where similarity measurement is performed between a user and form of sales on features derived from their profile and history information. As a result, we obtain a table where every single user gets his/her predicted value, as we can see in Table 3.

## 6 Experimental setup

As we want to use raw log data to make a prediction for recommendation, we have to handle the data sparsity problem. As already mentioned, our dataset contains 230,000. However, only 33,000 of them have data of sufficient quality. So, in our experiment, we used only this reduced and filtered dataset. To conduct experiments, we split the entire dataset into test (20%) and training (80%) sets.

In the first step, we have trained various classification models, including Logistic Regression [11], Random Forest [6], LightGBM [19], and XGBoost [8], where grid search was used to select the optimal parameters. As the final results proved, XGBoost classifier and XGBRegressor performed the best. Additionally, the majority of classifier (MC) [18] is used as a baseline for comparison with the above learning algorithms.

For the regression problem, we used the central tendency measure as the baseline for all predictions. Based on these, we inspected the hybrid models using the training set and adjusted the predictive algorithms' parameters achieving the best performance on the validation set. Predictions were made for each instance in the test set and the forecasted results were compared with the true values by computing corresponding performance metrics. To obtain the best evaluations we have used *K*-fold validation where

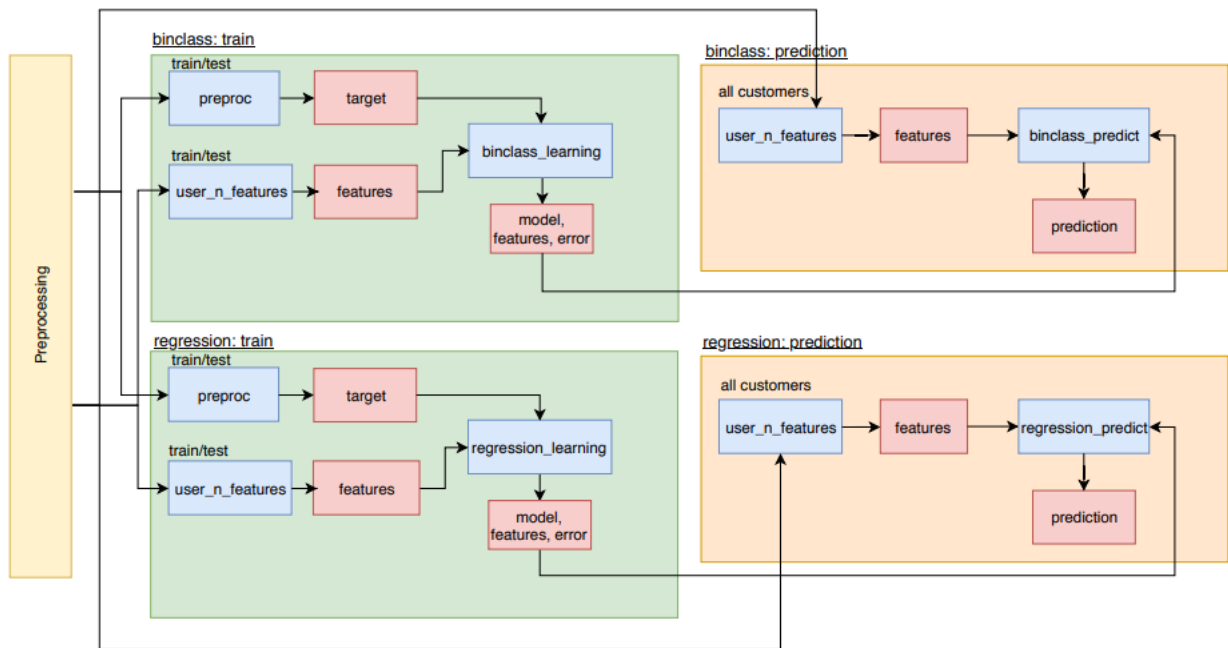


Figure 1: State diagram of our hybrid solution.

Table 3: Example of model outcome.

user id	likely to use sales promotions	likely to use sales promotion type				
		type1	type2	type3	type4	type5
1000	YES	35%	50%	5%	3%	7%
1001	NO	0%	0%	0%	0%	0%

both training and validation sets were also used for prediction.

### Handling problem with an ensemble classification and regression tree

The first goal is to predict if a user is likely to use or not a sales promotion, which is a binary classification problem. To find the best solution we have trained and tested classification models as many times as we could. In the end, we have found that the XGBoost ensemble classifier [8] gives the best results. It is not surprising, because tree boosting is a highly effective and widely used machine learning method.

Another important feature is that the algorithm has a good performance as it includes an efficient linear model solver and can also exploit parallel computing capabilities [8]. Ensemble learning to provide a systematic solution to merge the power of multiple learners. The prediction value of XGB can have different interpretations, depending on the task, i.e., regression or classification. XGB is a tree ensemble model which set of classification and regression

trees. It could classify our data into one of a finite number of values, that while called a regression (nonlinear model). Besides XGB, we compared our results with Linear regression [10], Lasso [24] and Ridge regression [16].

## 7 Results

**Classification.** It is well known that the main problem of the recommendation system is the cold start problem. It could appear when the user has started his/her initial steps, or in our case when a shop owner started a new sales promotion type, which makes very sparse data. To solve this problem, we filtered (dropped out) those users and promotions from the training dataset which has too sparse or no data. Based on our model, we made a binary classification with XGBoost to predict user likely to use a sales promotion or not. The parameters of the estimator used to apply optimization by cross-validated grid-search over a parameter grid.

To find the most accurate model, we have tried more models and settings. The results are reported in Table 4, where the window size (number of purchase) was 3 for all

Table 4: Results of classifications.

model	ACC	F1	precision	recall
Baseline	0.587	0.342	0.351	0.337
Logreg_all	0.676	0.409	0.620	0.306
Logreg_top10	0.685	0.404	0.661	0.291
XGBoost_all	0.706	0.527	0.652	0.436
XGBoost_top10	0.703	0.518	0.657	0.419
XGBoost_all_HPT	0.768	0.519	0.666	0.423
XGBoost_top10_HPT	0.771	0.509	0.658	0.417
XGBoost_top10_HPT(4)	0.790	0.624	0.713	0.554

Table 5: Error rates of regression models.

model	Sales promotion Type1			Sales promotion Type2			Sales promotion Type3		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
Baseline_CV	5.840	53.568	7.313	9.970	161.948	12.723	12.679	256.261	16.001
DNN	5.906	53.275	7.298	9.870	158.492	12.589	12.600	259.132	16.097
LR_all_CV	5.039	46.029	6.779	8.927	131.045	11.442	11.368	206.408	14.365
LGBMReg_CV	4.715	42.469	6.551	8.446	118.202	10.869	10.946	191.677	13.843
StackReg_CV_TOP	4.778	43.153	6.564	8.720	125.506	11.200	11.092	196.392	14.013
LR_CV_TOP	4.986	44.844	6.691	8.829	127.842	11.301	11.234	203.471	14.284
LGBMReg_CV_TOP	4.700	42.349	6.501	8.602	112.589	11.067	10.895	191.120	13.824

the methods, except in the last configuration.

During the first phase, we have used XGboost with all, and with only the top-10 features, which achieved 70% accuracy. To improve this, we have applied hyperparameter tuning, namely cross-validated grid-search over a parameter grid which gains better accuracy.

We wanted to make further improvements, but the sparsity of the data did not allow it. The main problem is that we want to predict user feature habits as soon as possible. For that reason, we used the user's first 3 purchase history to train the model, but this was (as expected) not enough to improve the results. To get better results, we need more data, such as we expected it. The solution to this problem is simple: we have to wait for more information, or encourage clients to fill the profile table. To prove this concept, we have trained our model with the user's first 4 purchases, which achieves 0.79 accuracy (last row in Table 4).

To find another way for this challenge, we changed our method like many researchers suggest: if we don't have accurate enough classification model, we have to change our point of view. To use this idea, we retested our solution as a regression with XGBRegression (as a regression problem). As a result, it affords  $RMSE = 16.77$ , which is not offering better outcome, because if we transform this result into a classification result, we got accuracy: 0.686, precision: 0.578, recall: 0.546, and F1: 0.562.

**Regression.** In our second phase, we were looking forward to determining which type of sales promotion will prefer most of our users (see in Figure 1). It is a regression problem, where we have to predict every SP type for

every user. To make a measurable result, we did not test all the types of SP, instead of that, we chose only 3 types of promotion:

- Type1 is an SP type which has a long history in our webshop;
- Type2, which has only a year background, and
- Type3 is the youngest SP type (less than 6 months is using).

Based on this idea we obtained the results reported in Table 5.

To get the best outcome, we tested more models with different settings, like linear regression (LR), LightGBM (LGBMReg), and a simple deep neural network (DNN). In the initial step, our model used all ( $n = 129$ ) normalized, scaled and skewed feature sets. Based on this method LGBMReg made the most accurate solution.

As a second step, we wanted to increase our model's accuracy. To solve this, we wanted to find the most important features. For this purpose, we used wrapper method, namely backward elimination. As the name suggests, we gave all the possible data to the model at first. We track the performance of the model and then repetitively remove the worst performing features one by one until the overall performance of the model comes in a suitable range. To calculate feature importance, we are using the ordinary least squares (OLS) model [27]. After many attempts and settings, the best solution is made by LGBMReg which is a tree-based regression model, which made a much more accurate model than the random choice.

**Discussion.** Our problem and its solution to predict acceptance of the sales promotion is unique, since we do not predict a repeat purchase but a reaction to advertising letters. Regardless, we wanted to somehow compare the performance of our model with other models as well. The results, and methodology of our paper is similar to the results obtained by Martínez *et al.* [23], so we compared our results with theirs. Our goal was to predict if a user is likely to use or not a sales promotion, which was same as their binary classification problem. While our model reaches 79% accuracy, their solution reached 86.68%. The difference in accuracy between the two models is not surprising, since we used only the first 4 purchases, they used 24 months for the same task. As they noted, it is difficult to make an accurate prediction model from short data and few purchases, however, over time, as data is collected, we could produce more accurate results.

## 8 Conclusions

In this work, the goal was to build and share a structure of the model for predicting user habits about using sales promotions. As we saw in the literature review it is not a trivial case. There is a lot of gaps that we have to handle, for example, feature with different types (time, numeric, categorical, etc.) or scale. Based on human habits, the webshop's data is often log scaled, and sparse which makes it difficult for the model to find optimal parameters. There are now countless solutions to deal with this problem, like scaling, normalizing, skewing data, or find the most relevant features. Based on these methods, finally we identified a solution for our problem with relatively good accuracy results. For the classification problem we have found that XGBoost gives the best model, while the second solution is not that clear. Based on our results as at first glance LightGBM (LGBM) could be the right choice.

Before making our decision, we need to know the structure of the model. LGBM is a very popular solution, because of its speed and accuracy. It has happened because LGMB grows tree vertically while other algorithms like Xgboost, Gboost grow trees horizontally. Put it differently, LGBM grows tree leaf-wise while another algorithm grows level-wise. LGBM is giving the best solution for our task, but there is some gap, which overshadows our success. However, it is sensitive to overfitting, especially on small dataset. There is no threshold on the number of rows but researchers suggest to use it only for data with 10,000+ rows. This model hence cannot be used for new promotions that only used by a small amount of user. In the light of this, in the final model, we used linear regression which gives almost the same results as LGBM.

## References

[1] Ahmed, A., Low, Y., Aly, M., Josifovski, V., and Smola, A. J. Scalable distributed inference of dy-

namic user interests for behavioral targeting, In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011

<https://doi.org/10.1145/2020408.2020433>

[2] Aly, M., Hatch, A., Josifovski, V., and Narayanan, V. K. Web-scale user modeling for targeting. Proceedings of the 21st International Conference on World Wide Web, 2012 <https://doi.org/10.1145/2187980.2187982>

[3] An, M., Kim, S. Neural User Embedding from Browsing Events. In: Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track. ECML PKDD 2020 [https://doi.org/10.1007/978-3-030-67667-4\\_11](https://doi.org/10.1007/978-3-030-67667-4_11)

[4] Banerjee, A., and Ghosh, J. Clickstream Clustering using Weighted Longest Common Subsequences. In: The Web Mining Workshop at the 1st SIAM Conference on Data Mining, 2001

[5] Bozanta, A., and Kutlu, B. Developing a Contextually Personalized Hybrid Recommender System, Mobile Information Systems, Article ID 3258916, 2018 <https://doi.org/10.1155/2018/3258916>

[6] Breiman L. Random forests - random features. Technical Report 567, Statistics Department, University of California, Berkeley, 1999

[7] Burke, R. Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370, 2002 <https://doi.org/10.1023/A:1021240730564>

[8] Chen T. and Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016 <https://doi.org/10.1145/2939672.2939785>

[9] Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, T., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., Shah H. Wide & Deep Learning for Recommender Systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016 <https://doi.org/10.1145/2988450.2988454>

[10] Cook R.D. Detection of influential observations in linear regression. Technometrics, 19(1):15-18, 1977 <https://doi.org/10.2307/1268249>

[11] John N. D. and Ratcliff D. Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics, 43(5):1470-1480, 1972

- [12] Çano, E., Morisio, M. Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, 21(6), 1487-1524, 2017 <https://doi.org/10.3233/IDA-163209>
- [13] Essex, D. Matchmaker, matchmaker. *Communications of the ACM*, 52(5):16-17, 2009. <https://doi.org/10.1145/1506409.1506415>
- [14] P. Geurts, D. Ernst., and L. Wehenkel, Extremely randomized trees. *Machine Learning*, 63(1), 3-42, 2006. <https://doi.org/10.1007/s10994-006-6226-1>
- [15] Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., and Sharp, D. E-commerce in Your Inbox: Product Recommendations at Scale. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015 <https://doi.org/10.1145/2783258.2788627>
- [16] Hoerl, A. E., Kennard, R. W., Ridge Regression: Applications to Non-Orthogonal Problems. *Technometrics* 12(1), 69-82, 1970 <https://doi.org/10.2307/1267352>
- [17] A. Ibrahim, A. Osman, A. N. Ahmed, M. F. Chow, Y. F. Huang, A. El-Shafie. Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia, *Ain Shams Engineering Journal*, 12(2), 1545–1556, 2021
- [18] James, G. Majority vote classifiers: theory and applications. PhD thesis, Stanford University, 1998
- [19] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Advances in neural information processing systems*, pp. 3146-3154, 2017.
- [20] Koehn, D., Lessmann, S., Schaal, M. Predicting online shopping behaviour from clickstream data using deep learning. *Expert Systems with Applications*, 150, 113342, 2020 <https://doi.org/10.1016/j.eswa.2020.113342>
- [21] Li, Z., Kulhanek, R., Wang, S., Zhao, Y., Wu, S. Slim Embedding Layers for Recurrent Neural Language Models. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.
- [22] G. Liu, T. T. Nguyen, G. Zhao, W. Zha, J. Yang, J. Cao, M. Wu, P. Zhao, W. Chen. Repeat Buyer Prediction for E-Commerce. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, 2016. <https://doi.org/10.1145/2939672.2939674>
- [23] A. Martínez, C. Schmuck, S. Pereverzyev, C. Pirker, M. Haltmeier. A machine learning framework for customer purchase prediction in the non-contractual setting. *European Journal of Operational Research*, 281(3):588–596, 2020 <https://doi.org/10.1016/j.ejor.2018.04.034>
- [24] Park, T., Casella, G., The Bayesian Lasso. *Journal of the American Statistical Association* 103(482), 681-686, 2008 <https://doi.org/10.1198/016214508000000337>
- [25] Sidana, S. Recommendation systems for online advertising. *Computers and Society [cs.CY]*. Université Grenoble Alpes, 2018.
- [26] Vieira, A. Predicting online user behaviour using deep learning algorithms. arXiv preprint arXiv:1511.06247, 2015
- [27] Weiss, A. A Comparison of Ordinary Least Squares and Least Absolute Error Estimation. *Econometric Theory*, 4(3), 517-527, 1988 <https://doi.org/10.1017/S0266466600013438>
- [28] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 729–732, 2016 <https://doi.org/10.1145/2911451.2914683>
- [29] Zhang, Y., and Pennacchiotti, M. Predicting purchase behaviors from social media, In: *Proceedings of the 22nd International Conference on World Wide Web*, 2013 <https://doi.org/10.1145/2488388.2488521>

# Evaluation of Multimedia User Interface Design Method for M-learning (MobLearn): A Comparative Study

Shimaa Nagro

College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia

E-mail: s.nagro@seu.edu.sa

Maysoon Aldekhail

College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia

E-mail: msdkhail@imamu.edu.sa

**Keywords:** M-learning, user interface, comparative study

**Received:** December 9, 2020

*The multimedia m-learning method (MobLearn method) is a holistic and functional method that includes the key steps, methods, and techniques for using multimedia to create m-learning interfaces. In this paper, the MobLearn process was evaluated by comparing it to one of the latest m-learning methods by Stanton and Ophoff (2013) which has the same function. A comparative case study was employed to examine the degree to which two m-learning approaches are similar in terms of interface architecture and their main characteristics, their differences and the primary function of features that occur in one of them but not in the other. Based on this comparative study, the final version of MobLearn method was introduced in two forms: high-level framework and method steps. In this version, the mapping strategies of m-learning were applied where the mapping techniques deal with a different set of information types and a large variety of media.*

*Povzetek: Primerjana je multimedijska metoda MobLearn z m-metodo Stanton in Ophoffa.*

## 1 Introduction

We are currently undergoing a knowledge revolution in which computer-based communication and technology are increasingly evolving and affecting all aspects of our lives. One of the most important impacts of the technical revolution has been in the field of education and learning. E-learning can occur as synchronously or asynchronously either inside or outside a classroom. More recently, the introduction of M-learning as an extension of E-learning has led to great developments in mobile technologies. M-learning is essentially the use of various mobile technologies to provide opportunities for learning anywhere and anytime using mobile phones, smart phones, PDAs, tablets and wireless networking technologies [1].

The recent emergence of technology such as mobile devices, multimedia interfaces, and m-learning has resulted in many mobile educational apps being developed. These kinds of programs have been developed to enable non-programmers to develop their own software for m-learning. Although these applications are effective in creating m-learning applications, they do not provide affordable, essential, design-based procedures to meet student requirements in design media selection.

The multimedia m-learning method (MobLearn method) presented in Nagro and Campion forms a complete and user-friendly method that encompasses the main steps, sub-steps, tools, and techniques required to produce m-learning interfaces using multimedia [2]. It includes all the stages required to build software

applications that specialize in m-learning systems. The effectiveness of this method was evaluated using a case study on a historical topic, and the evaluation confirmed that the method's steps were not only effective for designers but also had a positive impact on multimedia user interface design for mobile learning. In addition, these authors published another research paper [3] explaining the steps of the proposed method, and evaluated the usability of the method using think-aloud protocol. The method encompasses the steps, tools, and techniques necessary to produce M-learning interface using multimedia.

In this particular research study, the authors want to apply a comparative case study to investigate the extent to which two m-learning methods are similar, and their main characteristics when designing interfaces. It also investigated their differences and the main purpose of those features that exist in one but not the other. This chapter also aims to evaluate the MobLearn method by comparing it to one of the existing m-learning methods that has the same goal. The researcher chose Stanton and Ophoff's method [4], which describes a high-level M-learning design method containing eight steps. Comparative case studies include more than one case to produce generalizable knowledge about a specific topic. They emphasize comparison within and across various contexts. They encompass the analysis and synthesis of the similarities, differences, and the patterns across two or more cases that share a common focus or goal [4].

In this paper, section 2 describes the related work, section 3 discusses some key updates to the MobLearn method version 6 published in [3], section 4 explains the comparative study criteria, section 5 provides a discussion and an analysis of the results, section 6 explores the final version of the MobLearn method's framework, including both the general and detailed steps, section 7 offers a comparison with other methods, and finally, section 8 offers a conclusion.

## 2 Related work

Online learning styles have evolved from E-learning to M-learning which enables unlimited accessibility to learning content. This is majorly due to the advancement of technologies and enhanced mobile phones such as, smart devices and tablets. Consequently, traditional learning materials are being updated and redesigned to include compatibility with M-learning [5]. Nevertheless, the effects of certain mobile device features may hinder the efficacy of M-learning with issues relating to wireless internet connectivity, input/output systems, smaller sized screens, and battery life [6]. Taking these limitations into consideration is important during the modification of information to be delivered through mobile devices. Although several educational institutions have created their own applications, these apps have not taken into account several factors which contribute to the successful delivery of course materials through mobile devices [7]. In this section, there will be a description of the current methods, frameworks and guidelines that are available to design M-learning.

SOAP protocol enables the use of XQuery language, which then allows contents from a website to be adapted to the user's particular mobile device which works by transferring information structures from one operating system to another. However, this makes adaptation only possible for text formatted materials instead of rich formats as those contain multimedia [8]. Similar to the XQuery language, Huang et al. recommend the FWA algorithm which enables the conversion of different styles of web content to be adapted to any device [9]. Most research work has been focused on this area. For instance, an intelligent agent that can automatically alter and send all electronic messages to any device was created by Ally et al. In this case, the agent finds and chooses the optimal tool for conversion based on the software requirements and hardware limitations of each device [10]. On the other hand, this process can take anywhere from 10 to 30 minutes to convert each webpage for the particular mobile device, making it extremely time consuming. Nonetheless, a content adaptation system was designed that enables it to select the best version of the converted webpage for the mobile device in question. However, this process also has the major disadvantage of being very time consuming, offering users the option to skip any multimedia to be displayed in order to save time [11]. A system architecture for learning resource adaptation and delivery framework produced by a different study looks at some of the issues behind M-learning, for example, internet connectivity and wireless speed [12]. The study proposed engaged two

process layers to guarantee the good quality of the materials being delivered: 1) a multimedia adaptation layer that considers the quality of the multimedia item, and 2) a learning object adaptation later that considers the quality of the learning objects. But while the research study considered the adaptation of existing websites to mobile devices, it did not take into account the design of original content delivery through mobile devices. Therefore, there are many issues due to this limitation, the process of transferring, and resulting interfaces system.

Many studies have also been conducted on creating different approaches on design and development of multimedia interfaces. Heller et al. divided multimedia into three main aspects: 1) media (text, sound, graphics, and motion), 2) context which considers target audience, discipline, interactivity, quality, usefulness, and aesthetics, and 3) media expression which includes elaboration, representation, and abstraction [13]. Their study proposed certain guidelines on how to plan particular media types based on information types. On the other hand, their study did not provide or design a method; they simply assessed educational multi-media. Nevertheless, according to Chen et al. combining e-learning materials online with different mobile technologies that can help contribute to an effective learning environment is certainly possible [6].

The Universal Instructional Design (UID) was adapted to M-learning concepts by Elias. The features of UID were designed to guide and help interface developers and designers who need to create educational material interfaces for a varied group of students [14]. Particularly useful in educational settings, the UID principles include the following:

1. equitable use,
2. flexible use,
3. simple and intuitive,
4. perceptible information,
5. tolerance for error,
6. low physical and technical effort,
7. community of learners and support, and
8. instructional climate.

The adapted UID has demonstrated good results with effective interfaces.

Campion suggests a method for designing tasks with multimedia integration using rules to guide the educational materials to a particular type of media [15]. A method and advisor tool for designing a multimedia user interface was also proposed by Sutcliffe et al. which considers user requirements, media selection, and data design when representing information [16]. Moreover, the latter method's advisor tool is a useful addition that helps in selecting the appropriate media to display the information content. Nonetheless, both of these methods do not consider the limitations of M-learning. Sutcliffe et al. offered a high-level M-learning approach that takes into account the course proposal and design as well as the objectives and usability measures [16]. Despite this approach, this method failed to take into account the following: the course structure, mapping information, and lesson analysis. In order to teach Dijkstra's shortest path algorithm using mobile devices, Seraj and Wong proposed



a designing flash-based user interface; moreover, they also urge all designers to use UID principles when designing educational materials for mobile devices [17]. But while the strategy focuses on teaching Dijkstra's shortest path algorithm, it cannot be applied to design user interfaces for general educational material. Lee asserts that both technical and design should be factored in the decision to use mobile devices in learning environments [18]. The technical limitations can be attributed to the device functionality, and the design problems are based on the development of appropriate educational materials delivered via mobile devices. Lee provides a set of comprehensible instructions to follow when designing educational platforms on interfaces, taking into account the website browsers, and within the screen where to best position the features of the interface. Moreover, he also recommends UID designers to take into account which particular device they will consider before starting the design work.

There have been some research studies that concentrated on the production of design principles; in 2012, for example, there was design proposal for M-learning practice tips [19]. Moreover, later in the same year, Ryokai et al. conducted a study that created the following design principles: "connect, contextualize access, capture, and multimodal" aiming to connect the gap between M-learning and what actually goes on in the classroom [20]. There was yet another study published in 2011 that included suggestions for designing M-learning messages for different devices [21]. Wang and Shen's put 4 principles included:

Principle 1: "Design for the least common denominator"

M-learning materials should be designed in a format that is appropriate for all mobile devices.

Principle 2: "Design for E-learning, adapt for M-learning"

Utilize the same processes in E-learning for M-learning, such as the iterative design approach which consists of design, creation and evaluation.

Principle 3: "Design short and condensed materials for smart phones"

Shorten and consolidate the course materials on the mobile devices screen by including images, audio notes and a summary of the materials.

Principle 4: "Be creative with 3G and 4G technologies"

3G and 4G technologies have offered the designers the chance to run more sophisticated programs on mobile devices.

In 2010, web interfaces with responsive design were first initiated by [22]. Responsive web design entails flexible design interface that can be utilized pragmatically on all types of mobile device screens without compromising on content [23]. However, there is a lack of consideration for M-learning design principles; thus, this approach merely presents the design aesthetically on the mobile device. While it may be useful technology for designing M-learning materials, it can be further enhanced

by taking into account certain M-learning design principles. These days, it is more common for app developers to consider design for mobile devices before desktop computers; even though designing for mobile devices may have more limitations. Essentially, this design strategy entails starting to create or design for the smallest screen first and progress towards larger formats.

To sum up, there is a dearth of studies on exploring design for M-learning interfaces that integrate multimedia formatting. It is recommended that further examination be conducted to analyze and explore other potential methods that could resolve design limitations. An ideal solution would include a comprehensible set of instructions to guide the selection and integration of media, address implementation issues, and evaluate other areas that need improvement while considering the features of mobile devices.

### 3 Method for updating comparative study

The researchers decided to update the MobLearn method presented in [2] & [3]. This update improved this method so that it incorporated certain evolving features. The new version of the proposed method considers gamification and virtual reality in the mapping table and lesson plan in the information type table. It also includes an interaction table, which states all the possible technologies that could be used when interacting with multi-media.

The proposed method also considers pedagogical analysis; these techniques in the first step of the MobLearn method provides some guidance on how to analyse the educational material [24]. Pedagogical analysis is a process of breaking down the lesson into smaller sections and is defined as "the analysis of a given content material in any subject any topic carried out well in the spirit of the science of teaching (Pedagogy) is known by the term pedagogical analysis of the contents" [24].

#### 3.1 Comparative study design

According to Goodrick (2014), a comparative case study fundamentally engages six steps, which are preferably undertaken in the following order:

##### I. State the key evaluations

This step is important to determine whether or not the use of comparative case studies techniques is a convenient design. The purpose of using comparative case studies may be either be an explanation, an interpretation, and/or a comparison [25]:

- An explanation and an interpretation of the similarities and differences is determined between the cases in order to produce a holistic understanding of how the method functioned and how to direct additional implementation. The key evaluation question for the conducted comparative study in this research is as follows: what are the similarities and differences between these m-learning methods? The aim of this question is to develop a new, comprehensive version of the proposed method with the most beneficial characteristics after the interpretation of the results.

- Comparisons can uncover certain explanatory propositions regarding how and why a method functioned in specific contexts. The key evaluation question for the conducted comparative study was what are the characteristics that make both of these methods suitable/unsuitable for the m-learning interface design? The aim of this question is to be able to identify any significant features not included by the MobLearn method that are seemingly significant for the m-learning interface design.

## II. Determine initial proposition

This step aimed to clarify some of the initial propositions, which include a clarification of how the study was assumed to contribute to the results that will generate the intended goals [25]. In order to specify initial propositions, first a selection between a “within-group design” and a “between-group design” was made. This is a critical decision in all comparative studies as it has a direct impact on the quality of the data collected [26]. The within-group design was used to conduct this study. The time restrictions of the study, the sample size, and the non-existence of individual differences are the reasons behind this choice of strategy. Additionally, this study was not funded so the researcher was unable to find a larger sample size. To overcome the disadvantages of this type of design, the researcher provided a quick tutorial to design participants before they started the study; this helped them understand how to use each method to design interfaces for m-learning. To reduce fatigue, a 10-minute break was allowed after each step and the researcher helped clarify any unclear ideas. The impact of the learning effect was considered very low in this study as the participants used two different methods to design the interfaces, which is the main topic of the study.

As discussed in Nagro and Campion, in a comparative study, the cases could be as few as two cases [25]. The researchers decided to conduct five comparative cases with different domains (anatomy, chemistry, genetic engineering, software engineering and mechanical engineering). Using several domains for the design process increases the opportunity to examine more of the information type and mapping tables.

The participants’ criteria must be as represented in Table 1 to ensure the suitability and generality of the MobLearn method for most available mobile devices, the researcher decided to use five different models of mobile devices, as follows:

- In the first sub-study (anatomy): the designer designed a set of interfaces for the Samsung Galaxy Tab Pro 5.
- In the second sub-study (chemistry): the designer designed a set of interfaces for the Samsung Galaxy S8+.
- In the third sub-study (genetic engineering): the designer designed a set of interfaces for Apple iPhone 8.
- In the fourth sub-study (software engineering): the designer designed a set of interfaces for the Nokia 5.

Age	20+
Gender	Male or female
Mental ability	Average
Educational background	School education as a minimum
Physical attributes	Normal
Motivation	Must have positive attitude toward designing
Personality	Patient
Job	Representative designers, Junior designers, lecturers has designing experiences.
Computer usage	Average
Interface designing experience	Any

Table 1: User Profile.

- In the fifth sub-study (mechanical engineering): the designer designed a set of interfaces for the Huawei P10 lite Black.

## III. Conducted the case study process

The cases took place in a quiet office with a computer on a desk and A4 paper to draw sketches. The researcher provided a quick tutorial on how to use both methods, and then presented the steps and the aim of the studies. Subsequently, the participants executed the design work before starting the initial sketches. After designing the interfaces using one of the provided methods, they were asked to conduct the other condition with the other method. After each case, the participants were interviewed to identify any difficulties, advantages, and disadvantages they had faced during the cases. During the interview, each participant needed to answer the Key Evaluation Questions. These two questions were as follows:

- What are the similarities and differences between these m-learning methods?
- What are the characteristics that make both of these methods suitable/not suitable for m-learning interface design?

The researchers conducted five cases, as mentioned previously, as the within-group design allows a small number of participants [26]. As a deep understanding of each case is required, comparative case studies tend to allow as few as two cases to be involved [25]. Each participant conducts one case.

## 3.2 Collection and analysis of results

After finishing all cases, the data was collected from the participants by interviewing them and encouraging them to comment on each method. The information was then analysed as qualitative data to improve the proposed method. The comparison spurred some new questions regarding the similarities and differences between the two cases, as discussed in the following sections.

## 4 Findings

The results of the five comparative cases conducted are summarized in Tables 2 and 3 below. Accordingly, the results were divided into negative and positive points for

Step	Interviewee	MobLearn method	Stanton & Ophoff's method
Negative points			
Step 1		Nothing identified.	
Step 2	1	- In case the designer wants to use another media, it is not allowed by this method.	
	2		- Without an actual decision support theory
	3	Needs to be automated.	- Does not have any guidance and depends on designer's opinion - Does not account for information types - Does not know if it is the most appropriate media - No guidance on how to choose the most appropriate media
	4		- Just considers 3 types of media (audio, video, & image)
	5		- Suggests using three main media (audio, video, image)
Step 3	Nothing identified.		
Step 4			
Generally	1	- Too long to follow, time-consuming and requires lots of concentration - Is apparently not based on theory. If it is, then may be it is better to explain when and where the method uses the theories	- Confusing - Does not give details on how to apply the theories provided - If the designer is unfamiliar with these theories, they will struggle to apply them - More general and needs more experienced designers
	2	- Compatibility and interaction design unifies the method of empty slots and means it is compatible	- More learning strategy based - More decision support statements instead of referencing other theories - Provides processes and sign posts for each process steps
	3	Needs concentration and is time consuming	- Lack of restrictions led to lack of accuracy
	4	Too long	- Difficult to apply, because it is too general
	5	Consumes time	- Not good for non-experts - Consider more types of media and some instructions on how to follow the steps

Table 2: Negative points identified by comparative study.

comparison, and were divided based on each step of the four steps method as presented in [2].

As shown in Tables 2 and 3, the results were categorized into negative and positive points to show the comparison.

### 5 Discussion

The comparative studies examined two m-learning methods that aim to design m-learning interfaces for a lesson. Both methods consisted of a set of steps based on several theories to guide the designing process. The participant designers were asked to identify the similarities and differences between both methods and specify the better choices for them. As argued by some participants, both of these methods produce interfaces suitable for m-learning. This is because both considered user-centred design [27], and the pedagogical analysis for the provided lesson. Moreover, both are based on all the steps in m-learning theories. Thus, both methods heed the importance of the lesson objectives and how to divide the lesson into smaller parts that are suitable to be represented on relatively small screens. Finally, both methods encourage designers to evaluate and test the system before finalizing it.

Most participants argued that the MobLearn method is time consuming as it has many steps and focused on more details. These reasons make this method more suitable for non-expert designers and designers with

limited knowledge of how to use media for m-learning interfaces. Yet, one participant argued that the MobLearn method could be used by experts and non-experts, unlike the m-learning method by [4], which is not suitable for non-experts as it refers to a considerable number of design theories without explaining them. However, another designer thought that the MobLearn method has many restrictions to consider; thus, they inferred that it was more accurate in designing. This is why some participants suggested producing an automated version of the mapping process, so that when the designer chooses any information type, the system can show all the possible media they can utilize. This will be considered in our future research studies.

The main difference between these m-learning methods is in choosing multimedia. In the m-learning method by [4], the researchers referred to the design principles in both [19] and [20]. The principles are as follows: connect, contextualize access, capture, and multimodal. These principles connect the classroom with what is delivered via mobile devices for students and the importance of creating a personal connection to the material for the students by ensuring that it is relevant and meaningful. Multimodal refers to making the content accessible via multiple learning styles: visual, auditory and kinaesthetic [19]. These concepts are important in m-learning design; however, the researchers did not guide the designer on how to choose multimedia. Instead, they only

Step	Interviewee	MobLearn method	Stanton & Ophoff's method
Positive points			
Step 1	1	<ul style="list-style-type: none"> <li>- Encourages the designer to look at the material and divide them using some pedagogical analysis</li> <li>- Step 1 is a critical step. I consider it as the base of the designing process. Designers need to determine objectives, functional requirements and information type</li> </ul>	<ul style="list-style-type: none"> <li>- Encourages the designer to look at the material and divide them using some pedagogical analysis</li> <li>- In the other method the first step is indicated by green boxes. It is concerned with analysing the pedagogy and it relates them to the objective, which also gives the key points for the designing process</li> </ul>
	2	<ul style="list-style-type: none"> <li>- Provides steps that are divided in a logical order to start planning a lesson.</li> </ul>	
	3	<ul style="list-style-type: none"> <li>- Organises the delivery of the lesson depending on objectives</li> <li>- Depends on information type to choose media</li> <li>- Shows the possible combinations of media</li> </ul>	<ul style="list-style-type: none"> <li>- Organises the delivery of the lesson depending on objectives</li> </ul>
	5	<ul style="list-style-type: none"> <li>- Supports the designers to know the objective before designing commences, which leads them to discover the content of the lesson</li> <li>- Analysing the content is a good start</li> <li>- Information type table made it easier for me to choose multimedia</li> </ul>	<ul style="list-style-type: none"> <li>- Supports the designers to know the objective before designing commences, which leads them to discover the content of the lesson</li> <li>- Analysing the content is a good start</li> </ul>
Step 2	1	<ul style="list-style-type: none"> <li>- More determined by using information type table and mapping table</li> <li>- I found all multimedia I want in the mapping table</li> <li>- When it is my first time to design I will use the MobLearn method, which makes me more knowledgeable of all information types and multimedia available</li> <li>- Detailed mapping process is beneficial</li> <li>- Suitable for mobile learning design as it guarantees the choice of the most correct media</li> </ul>	<ul style="list-style-type: none"> <li>- Good as it gives the designer freedom to choose the multimedia</li> </ul>
	2	<ul style="list-style-type: none"> <li>- Supports decision making in the process</li> <li>- Connecting between the learning material and the multimedia choice</li> </ul>	
	3	<ul style="list-style-type: none"> <li>- Guidance to choose the most appropriate media, and it guides you step by step to do this</li> </ul>	
	4	<ul style="list-style-type: none"> <li>- Mapping table is helpful, it helped me to decide which media to apply</li> </ul>	
	5	<ul style="list-style-type: none"> <li>- Considers more than 3 media and on the positive side it assists the designer with choosing the media</li> <li>- Considers the delivery of the content using multimedia</li> </ul>	<ul style="list-style-type: none"> <li>- Considers the delivery of the content using multimedia</li> <li>- Easy to decide which media to use</li> </ul>
Step 3	4	<ul style="list-style-type: none"> <li>- Analysing the lesson considering mobile screen size</li> </ul>	
Step 4			
Generally	1	<ul style="list-style-type: none"> <li>- Steps and sub-steps are clear to follow</li> <li>- Suitable for mobile learning design</li> </ul>	<ul style="list-style-type: none"> <li>- Not time consuming</li> <li>- Not long to follow</li> <li>- States some theory which is good to build a method on</li> <li>- Suitable for mobile learning design</li> </ul>
	2	<ul style="list-style-type: none"> <li>- Appropriate for both lessons or complete courses</li> <li>- Colour coding to distinguish the steps</li> <li>- Table layout</li> </ul>	<ul style="list-style-type: none"> <li>- Colour coding</li> <li>- Clarity of the main Processes</li> </ul>
	3	<ul style="list-style-type: none"> <li>- Clear</li> <li>- Accuracy in choosing media</li> </ul>	<ul style="list-style-type: none"> <li>- Easier to follow as it does not contain many restrictions</li> <li>- Not time consuming</li> </ul>
	4	<ul style="list-style-type: none"> <li>- A long method, which requires attention to detail</li> </ul>	
	5	<ul style="list-style-type: none"> <li>- Follows some defaults steps, any expert designer can follow them</li> <li>- Could be used for non-expert and expert designers, especially the second step</li> <li>- Produces well-designed interfaces, I mean the variety of multimedia available to choose from</li> </ul>	<ul style="list-style-type: none"> <li>- Did not take long to apply</li> </ul>

Table 3: Positive points identified by comparative study.

considered which three multimedia (video, audio, and image) could have been used [4]. On the other hand, the MobLearn method starts this process by specifying for each piece of information a type by using a table that

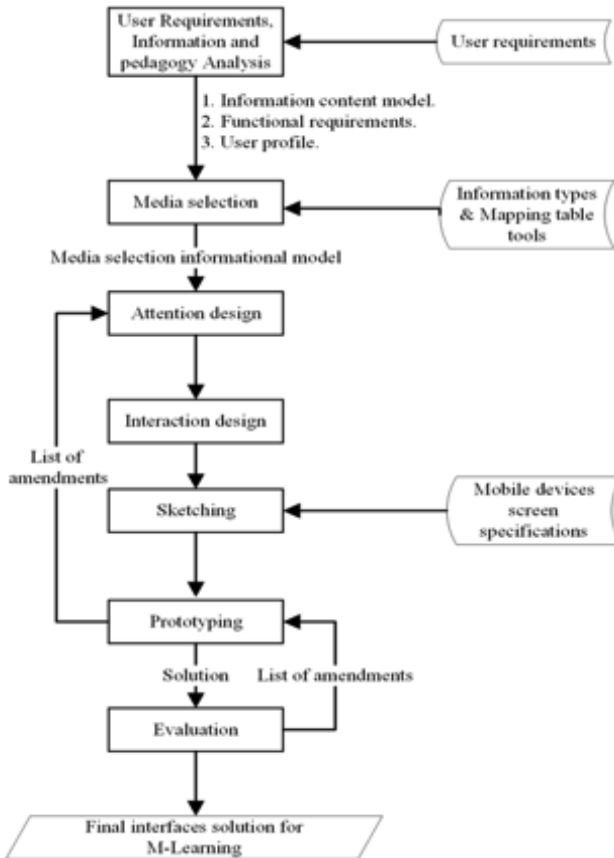


Figure 1: The Framework of the Method.

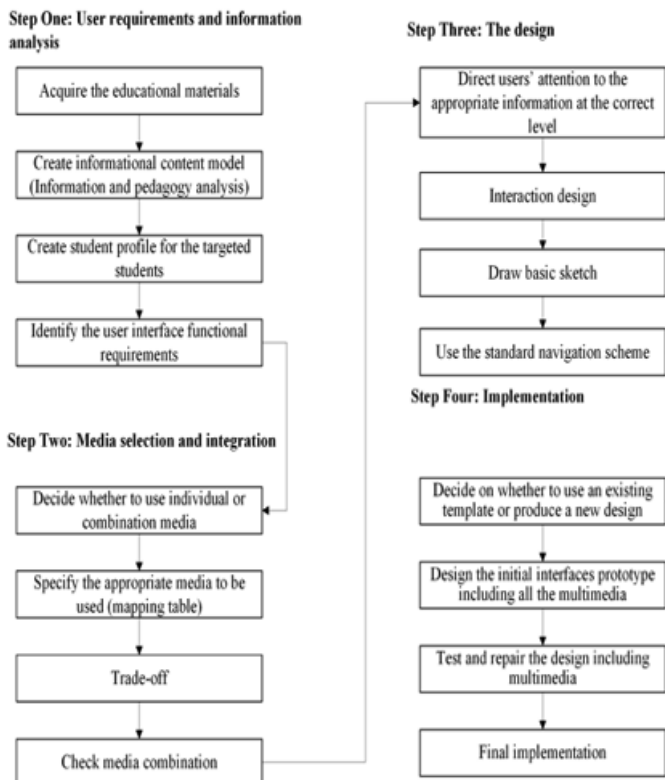


Figure 2: The Method's steps.

indicates all the possible information types, a description of these types, and some examples to help the designer with the decision making process. The MobLearn method

then directs the designer to map each of the information types to set of possible multimedia by following a set of instructions provided. The participants of the comparative study agreed that these steps were clear, and that they supported the designers in choosing media by connecting the actual education material with the multimedia choices.

## 6 Final method

The following section presents the method's final version. As with the original prototype method in [2] this final version consists of three stages: 1) user requirements and information analysis, 2) media selection, integration and design, and 3) system implementation. The following section presents the method's framework.

### 6.1 Method's framework and method's steps

The method's framework shows the basic structure of the method (see Figure 1). It generally represents the main steps with the input and output associated with each process. The changes made to the final version were based on the results of the comparative study. Compared to the previous versions in [2], the first phase became user requirements, information, and pedagogy analysis, and the input for the media selection phase becomes information types and mapping table tools.

Figure 2 below portrays the steps of the method, where the first stage addresses step one, stage two addresses steps two and three in the method, and finally the third stage is addressed in step four.

## 7 Comparison with related methods

Table 4 below shows the main weaknesses in existing methods, found after conducting the literature review, and the gaps found and filled by the MobLearn method.

## 8 Conclusion

In this study, the authors used a comparative case study to explore the degree to which two methods of m-learning are identical as well as their key characteristics with reference to interface design. It also discussed their variations and the primary purpose of characteristics that occurs in one of them but not in the other.

The paper also assessed the process of MobLearn by contrasting it to one of the latest methods of m-learning that has the same purpose. The researchers chose the method of Stanton and Ophoff (2013), which defines a high-level method of m-learning design consisting of eight steps. The paper presented the final version of the method in two forms: high-level and framework overview of method steps. It gave an overview of the method, which represented the method as a framework. It also gave an overview of the method in the form of steps. In the final version, the research adds the mapping techniques for m-learning design to the body of knowledge. Specifically, it supports studies that have suggested that mapping the information type to suitable media should also consider students' cognitive ability. The mapping techniques deal

Weaknesses and Gaps	Other Methods	MobLearn Method
Interaction techniques for mobile devices	A multimedia design method for task requirement formulation, media integration, and device combination (TRIUMPH) by Campion (1999) and a method and advisor tool for multimedia user interfaces by Sutcliffe et al. (2006). Both consider interaction techniques. However, both methods do not consider new interaction techniques such as accelerometer, haptic and gestural. The second method has a limitation, as was necessary more advice on interaction and dialogue design (Sutcliffe et al., 2006).	MobLearn method overcomes these limitations in the interaction table
Task scenarios for mapping information to multimedia	A multimedia design method for task requirement formulation, media integration, and device combination (TRIUMPH) by Campion (1999), a method and advisor tool for multimedia user interfaces by Sutcliffe et al. (2006) and a framework for mapping multimedia to educational concepts by Onyekaba, Campion and Atkins (2016). All of these consider task scenarios to produce multimedia interfaces but not for m-learning devices. Sutcliffe et al. (2006) method did not consider mapping information to multimedia as it depends on heuristics and a decision tree. The framework for mapping multimedia to educational concepts by Onyekaba, Campion and Atkins did not consider m-learning [28].	MobLearn method considers task scenarios to produce multimedia m-learning interfaces. In the second step, it considers task scenarios to map the educational concepts to multimedia for m-learning devices.
Further analysis		The proposed method gives details on how to analyze the educational materials (pedagogical analysis), following a method adapted from Bhowmik et al. (2013). The pedagogical analysis takes place in the first step and advises how to decide on information type.
Selection guidelines, general and validating heuristics for mapping	The multimedia design method for task requirement formulation, media integration, and device combination (TRIUMPH) by Campion (1999) depends on media model for mapping. Sutcliffe et al. (2006) method depends on guidelines and heuristics. The framework for mapping multimedia to educational concepts by Onyekaba, Campion and Atkins (2016) depends on a framework.	The proposed method gathers all the information related to mapping in existing methods and produces a mapping table to map the educational concept to multimedia for m-learning devices, not for general interfaces. It also suggests mapping to new technological practice such as gamification and virtual reality.
Does the method address integration?	Some of the methods address integration; however, they leave it to the designer to decide which media combination to use.	The proposed method allows the designer to go through several steps to produce one optimal solution for integration, depending on the most frequently occurring media and recommended media.
New technology considered	N/A	The proposed method considers gamification and virtual reality in the mapping table, to help the designer to decide when to use them and what educational concepts are most suitable for them.

Table 4: Results of comparison of MobLearn method with related methods.

with a clear set of information types and the following set of media: animation, non-realistic audio, charts, diagrams, graphs, lists, realistic audio, network charts, maps, music, photographs, sketches, speech, tables, text, videos, captions, gamification and virtual reality.

## Acknowledgement

We thank Dr. Russell Campion for his tremendous support and feedback during this research study.

## References

- [1] M. AlDekhal (2015). "E-Learning Assistance and Application for the Auditory-Impaired Population: A Review with Recommendations." *Journal of Basic Applied Scientific Research (JBASR)* 5(3), pp. 36-53 [online]. Available at: <https://www.textroad.com/>
- [2] S. Nagro and R. Campion (2017) 'A method for multimedia user interface design for mobile learning.' *IEEE: Computing Conference*. London, United Kingdom, pp. 585-590. [online]. Available at: DOI: 10.1109/SAL.2017.8252155
- [3] S. Nagro and R. Campion (2017) "Evaluation of multimedia user interface design method for mobile learning." In: *Proceeding of The 11th annual International Technology, Education and Development Conference (INTED)*. Valencia, Spain, 6-8 March 2017, pp. 1533-1541. [online] Available at: DOI: 10.21125/inted.2017.0494
- [4] G. Stanton and J. Ophoff, (2013) 'Towards a Method for Mobile Learning Design.' *Informing Science and Information Technology*, 10, pp. 501-522. [online] Available at: DOI: 10.28945/1825
- [5] Y Jeng, T. Wu, Y. Huang, Q. Tan, and S. Yang (2010). "The Add-on Impact of Mobile Applications" in *Learning Strategies: A Review Study.* *Educational Technology & Society*, 13 (3), 3–11. [online] Available at <https://www.researchgate.net/>
- [6] G.D. Chen, C.K. Chang, and C.Y. Wang (2008) "Ubiquitous learning website: Scaffold learners by mobile devices with information-aware techniques",

- Computers & Education, vol. 50, no. 1, pp. 77-90. [online] Available at:  
DOI: 10.1016/j.compedu.2006.03.004
- [7] J. Cheon, S. Lee, S. Crooks, and J. Song (2012). "An Investigation of Mobile Learning Readiness in Higher Education Based on the Theory of Planned Behavior.: Computers & Education 59(3), pp. 1054–1064. [online] Available at:  
DOI: 10.1016/j.compedu.2012.04.015
- [8] T. Lemlouma and N. Layaida (2004) "Context-aware adaptation for mobile devices", in IEEE International Conference on Mobile Data Management, Saint Martin, France, pp. 106 – 111. [online] Available at: hal.inria.fr/inria-00423390
- [9] Y. Huang, Y. Kuo, Y. Lin, and S. Cheng (2008) "Toward interactive mobile synchronous learning environment with context-awareness service", Computers & Education, vol. 51, no. 3, pp. 1205-1226. [online]. Available at:  
DOI:10.1016/j.compedu.2007.11.009
- [10] M. Ally, F. Lin, R. McGreal, B. Woo, and Q. Li (2005) 'An intelligent agent for adapting and delivering electronic course materials to mobile learners', in Mobile Technology: The future of learning in your hands 2005, Capetown, South Africa. [online]. Available at: <https://citeseerx.ist.psu.edu/>
- [11] W.Y. Lum, and F.C.M. Lau (2002). A context-aware decision engine for content adaptation. IEEE Pervasive Comput., 1(3), pp.41- 49. [online] Available at: DOI:10.1109/MPRV.2002.1037721
- [12] Z. Gang and Y. Zongkai (2005) "Learning Resource Adaptation and Delivery Framework for Mobile Learning", in 35th ASEE/IEEE Frontiers in Education Conference, Indianapolis, IN. [online] Available at: DOI:10.1109/FIE.2005.1612035
- [13] R. Heller, C. Martin, N. Haneef, and S. Gievska-Krliu (2001) "Using a theoretical multimedia taxonomy framework", Journal of Educational Resources in Computing, 1(1), p. 6. [online] Available at: DOI: 10.1145/376697.376701
- [14] T. Elias (2011) "Universal Instructional Design Principles for Mobile Learning", The International Review of Research in Open and Distributed Learning, 12 (2), pp. 144-156. [online] Available at: <files.eric.ed.gov/fulltext/EJ920738.pdf>
- [15] R. Campion 'TRIUMPH: A multimedia design method for task requirement formulation, media integration, device combination, and practical implementation design issues', Ph.D., Staffordshire University, 1999. Available [online] Electronic Dissertation
- [16] A. Sutcliffe, S. Kurniawan, and J. Shin (2006). "A method and advisor tool for multimedia user interface design." in International Journal of Human-Computer Studies, 64(4), pp.375-392. [online] Available at: DOI: 10.1016/j.ijhcs.2005.08.016
- [17] M. Seraj and C. Wong (2015) "A study of User Interface Design Principles and Requirements for Developing a Mobile Learning Prototype", in Proceeding of the International Conference on Computer and Information Science, Kuala Lumpur, Malaysia, pp. 1014-1019. [online] Available at: DOI:10.1109/ICCISci.2012.6297174
- [18] S. Lee (2015) "Design Guidelines to Consider for Mobile Learning", Info.alleninteractions.com, 2015. [Online]. Available at:  
<https://blog.alleninteractions.com/>
- [19] J. Killilea (2012). Leveraging mobile devices for asynchronous learning: Best practices. scs.org. University of Central Florida. [online] Available at: <http://www.scs.org/>
- [20] K. Ryokai, A. Agogino, and L. Oehlberg (2012) 'Mobile learning with the engineering pathway digital library.' International Journal of Engineering Education, 28(5), pp.1119–1126. [online] Available at: <https://people.ischool.berkeley.edu/>
- [21] M. Wang and R. Shen (2011) "Message design for mobile learning: Learning theories, human cognition and design principles", British Journal of Educational Technology, 43(4), pp. 561-575. [online] Available at: DOI: 10.1111/j.1467-8535.2011.01214.x
- [22] E. Marcotte (2010) "Responsive Web Design", Alistapart.com, [Online]. Available at: <http://alistapart.com>
- [23] K.V. Natda (2013). "Responsive Web Design", Eduvantage, 1(1). [online] Available at: DOI:10.11635/2319-9954/1/1/18
- [24] M. Bhowmik, B. Banerjee, and J. Banerjee (2013) "Role of Pedagogy in Effective Teaching," Education Research and Review, 2(1), pp. 1-5. [online] Available at: <http://basicresearchjournals.org/>
- [25] D. Goodrick (2014) Comparative Case Studies: Methodological Briefs - Impact Evaluation No. 9, Methodological Briefs (No. innpub754). [online] Available at: <https://www.unicef-irc.org/>
- [26] J. Lazar, J. Feng and H. Hochheiser (2017) Research methods in human computer interaction. 2nd ed. USA: Morgan Kaufmann.
- [27] A. Errity (2016) "Human-computer interaction," in I. Connolly, M. Palmer, H. Barton, and G. Kirwan (Eds.), An introduction to cyberpsychology. Oxon: Routledge, pp.241- 254.
- [28] C. Onyekaba, R. Campion, and A. Atkins (2016) "A framework for mapping multimedia to educational concepts", Proceedings of INTED2016, Valencia, Spain, 7 - 9 March, 2016. pp. 7987 - 7996. Available at: DOI:10.21125/inted.2016.0880





# Impact of Gaussian Noise for Optimized Support Vector Machine Algorithm Applied to Medicare Payment on Raspberry Pi

Shrirang Ambaji Kulkarni, Varadraj Gurpur, Christian King and Andriy Koval

School of Global Health Management and Informatics, University of Central Florida, 32816, Orlando, Florida, USA

E-mail: sakulkarni@ucf.edu, varadraj.gurupur@ucf.edu, christian.king@ucf.edu, Andriy.v.koval@gmail.com

**Keywords:** medicare analysis, internet of things, data, statistical feature optimization techniques, support vector machine, pipelined models

**Received:** September 11, 2021

*A relatively large dataset coupled with efficient but computationally slow machine learning algorithm poses a great deal of challenge for Internet of Things (IoT). On the contrary, Deep Learning Neural Networks (DLANNs) are known for good performances in terms of accuracy, but by nature are computationally intensive. Based on this argument, the purpose of this article is to apply a pipelined Support Vector Machine (SVM) learning algorithm for benchmarking public health data using Internet of Things (IoT). Support Vector Machine (SVM) a very good performing machine learning algorithm but has constraints in terms of huge training time and its performance is also susceptible to noise. The applied software pipelined architecture to SVM was to minimize its computational time under a resource constrained device like raspberry pi. It was tested with a medicare dataset with Gaussian noise to assess the impact of noise. The classification results of Total Medicare Standardized Payment Amount obtained indicated that the proposed pipelined SVM model was optimal in performance compared to DLANN model by 79.74% in terms of computational time. Also the performance of SVM in terms of area under curve (AUC) was better compared to other models and outscored Logistic Regression by 7.2%, and DLANN model by 22.65%.*

*Povzetek: Analiziran je vpliv Gaussovega šuma na SCM metodo za plačevanje medijskih storitev.*

## 1 Introduction

Allhoffa and Henschke indicate that [1] Internet of Things (IoT) will become one of the greatest technologies that will revolutionize information capabilities and will have tremendous impact on the society at large. It is to be noted that IoT has limitations in terms of processing, memory and secondary storage capacities as compared to laptops, workstations and servers. Haller et al. [2][3] define IoT as “a world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business process.” On the other hand, Gokhale et al., [4] define IoT simply as a “network of physical objects.” Here they indicate that generally speaking devices, vehicles, buildings and other forms of hardware and their embedded software can be conceived as physical objects. IoT has been of special importance to the world of healthcare where organizations pertaining to the healthcare ecosystem are working towards reduction of costs and improving productivity. IoT is especially useful in decision support, transmitting information, and device control. Much of this pertains to the field of healthcare informatics. Healthcare informatics is defined by Wan and Gurupur [5] as “a transdisciplinary study of the data flow and processing into more abstract forms such as information, knowledge, and wisdom along with the associated systems needed to synthesize or develop decision support systems for the

purpose of helping the healthcare management processes achieve better outcomes in healthcare delivery.” The processes involved in synthesizing and developing decision support systems from knowledge and information requires innovative computational solutions and bolsters the need to advance data science especially pertaining to machine learning. Machine learning can be effectively performed in a suitable computational environment.

It is to be noted that edge computing or fog computing is becoming popular day by day as advanced biomedical devices are involved in collecting patient medical data thereby further improving processes associated with healthcare delivery. The advantages in terms of reduced latency between users, edge infrastructure and cloud are evident as described by Shukla et al., [6]. The central storage and sophisticated processing facilities provided by cloud facilities at times may suffer from network latency issues for real-time applications and may act as a single point of failure. It is to be noted that Machine Learning (ML) algorithms are being applied in plethora of applications in relation to the context discussed.

In the work delineated in this article the investigators explore Raspberry Pi as an edge computing device for benchmarking a popular ML algorithm Support Vector Machine (SVM). The SVM is defined by Noble [7], as “a

computer algorithm that learns by example to assign labels to objects.” As explained by Noble [7] SVM is a key algorithm that can be effectively used to identify patterns that can be used to train and label data for the purpose of classification. Here the classifiers performance is measured using the concept of Area under the Curve (AUC) as explained by Bradley [8]. This attribute brings about a key desired characteristic for analysing healthcare data. In the recent past many investigators have used the combination of Raspberry Pi and SVM to identify noise and patterns.

For experimentation and demonstration the investigators have used health care data with 40,662 rows and 28 variables, logistic regression algorithm for computational time and Deep Learning Neural Network (DLANN) for testing the accuracy of the classification results of Total Medicare Standardized Payment Amount. The reason for choosing SVM is its ability to produce results at higher level of accuracy; however, SVM tends to be constrained by high computational time and memory complexities for larger size training data [9]. This problem is compounded by the constrained computational resources of a Raspberry Pi and the presence of noisy data. The solution explored is an application of the pipeline architecture for SVM and its performances against the benchmarks set by of logistic regression and deep learning neural network on the same dataset.

The specific research objectives of the analysis are as follows:

- To analyse the performance of SVM with other benchmarks such as Deep Learning Machine Algorithm, and Logistic Regression in terms of accuracy and computational time under optimized and selected variable dataset for a resource constrained environment of Raspberry Pi and
- To implement a pipelined architecture model for SVM with feature selection and ascertain the consistency of performance in terms of metrics and robustness by evaluating the performances on a Gaussian Noise based dataset.

The presentation of a pipelined architecture is to contribute to the science of applying SVM to Medicare and Medicaid type datasets. Here the investigators are mindful of the fact that different datasets of different sizes and complexities require different approaches for analysis in terms of machine learning. More importantly it is important to state that the key targeted contribution of the experimentation explained in this article is to provide a computational method that can be effectively used in analysing healthcare data.

## 2 Related work

SVM suffer from high time required for training datasets [9][10]and memory complexities issues. These problems are compounded for large datasets and for noisy data were SVM had disadvantages in terms of performance, SVM was applied by Cheng-Lung Huang [11] for credit scoring. They proposed a SVM with Genetic Algorithms (SVM-GA). One of the drawbacks which they observed that SVM-GA took large training times and proposed SVM-

GA to be suitable for parallel architectures. Yazici et.al [12], in their work observed the performances of machine learning algorithms on raspberry pi as a part of their study on edge computing paradigm. Some of their results proved that SVM algorithm was slightly faster in inference and also efficient in power consumption. The above work’s motivated us to reduce SVM’s computational time by integrating it with a pipeline architecture model for working on moderately large datasets for a resource constrained environment like raspberry pi.

Nguyen and Torre [13] in their work discussed that feature selection aided Support Vector Machines towards generalization and computational efficiency. The authors proposed a convex energy-based framework towards feature selection and parameter selection. Experiments on seven different datasets and with feature selection helped them to retain the desired performances. Sanz et.al, [14] discussed in their work that predictor models with most relevant variables was one of the important criteria for biomedical research. They proposed the extension of Recursive Feature Elimination (RFE) based on non-linear SVM kernels. The proposed methods when applied on 3 different datasets performed better as compared to classical RFE.

Logistic regression a supervised learning is one of the popular models applied for classifying medical healthcare data. Logistic regression usually works on large sample size and thus the motivation to apply the same to our 2014 Medicare Provider Utilization and Payment Data [15]. Zardo and Collien [16] successfully used logistic regression to successfully identify critical predictor variables in public health policy research in Australia. Incidentally, Sheets et.al, [17] demonstrated the use of logistic regression in identifying attributes associated with high utilization of Medicare payments, thereby creating a burden on US taxpayer dollars. This research is focused on chronic patients and managed care and proactively identify high risk patients to reduce the cost of healthcare. Thus the present study would like to extend logistic regression to resource constrained environment of raspberry pi.

Deep Learning Artificial Neural Networks (DLANN) are more specialized forms of artificial neural networks and can also learn on their own and handle huge datasets to provide superior classification accuracy, but they also need huge computational resources. Sakr et.al, [18] in their work applied Convolutional Neural Networks (CNN) and SVM for automation of sorting waste on raspberry pi 3.SVM appeared to have higher classification accuracy as compared to CNN by outscoring CNN by 11.8%. Ravi et.al, [19] also studied the impact of Deep Learning algorithms on Health Informatics. They summarized that most of the deep learning algorithms were applied to balanced or synthetic datasets. Also, deep learning algorithms required large amounts of training data.

Thus, with algorithms like logistic regression, deep learning the investigators would like to benchmark the classification accuracy and related performances of support vector machine on a pipeline architecture on a resource constrained device like raspberry pi which holds lot of promise for edge devices. This analysis was carried

on a dataset of 40,662 records [15]. Gangsar and Tiwari [20] studied the impact of noise for fault diagnosis of electric machines. They found for perfect original signal SVM predicted with greater accuracy for all speeds. However, when white Gaussian noise was applied to the raw signal, the overall prediction accuracy fell by 10%. They considered 2% external noise for their study. Pei et.al, [21] in their studies considered the impact of images with white Gaussian noise and their performance effects on convolutional neural networks (CNNs). As the percentage of noise addition increased, the accuracy started to decrease. Wu and Zhu[22] analysed real world data in terms of noise handling features of data mining algorithms. They said error-aware data mining algorithms improved the data mining results. Last but not the least, in their work Zuolkernan et.al., [23] considered the application of remote cameras for monitoring animals. They considered an IoT based system whereby images captured on a camera are processed on the edge using Raspberry Pi and the accuracy results are moved to the cloud database system. To summarize application of SVM and other methods related to data science has immense potential that needs to be further explored and the experimentation presented in this article is a step taken in that direction.

### 3 Method and experiments

The block architecture of the experimental setup is as illustrated in Figure.1

The experiments were executed once the platform was laid, this included implementing the pipelined model for SVM, installing tensor flow for Deep Learning algorithms and a computational time model on a resource constrained environment of Raspberry Pi.

#### 3.1 Statistical optimization and performance

The dataset used for experimentation is a medical healthcare data that contains records for physical therapy patients and amounts paid to the physical therapists in each case Gurupur et al., [15]. It becomes imperative to consider feature section techniques for dataset pruning as

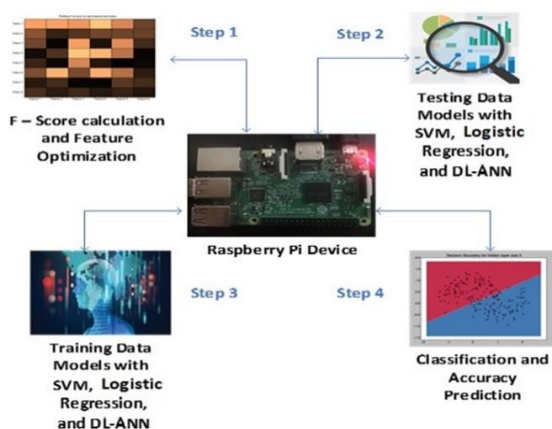


Figure 1: Block architecture of the experimental setup.

an optimization technique for resource constrained environment. Hardware Platform used for the experiment was Raspberry Pi B; Quad Core ARM Cortex A53 CPU 1.2GHz 64bit CPU with 1 GB RAM. From a software perspective, a python program was written using numpy, pandas and scikit-learn [24] along with keras and tensorflow; to apply logistic regression, SVM and DLANN for all variables in order to model them as a classification problem under supervised learning. This software platform was also used to execute metrics like K-Fold Cross Validation, Confusion Matrix and Area Under Curve (AUC).

The reason for applying statistical techniques for the dataset as follows:

- To optimize the data features so that it helps the machine learning algorithm to classify with a lesser number of variables.
- To identify outlier's and remove those from the dataset so that we have statistically a more normalized dataset.

Feature selection is an important step in the application of machine learning to achieve at times better performance from the models in terms of computational execution speed. The presence of irrelevant features may negatively affect this application. This creates the need for developing parsimonious models. The advantages could be minimizing the impact of overfitting, accurate results and reduce timing. Therefore, feature selection was the first step in the process. This was implemented using Python scikit-learn library [24] that provides a class called SelectKBest and to this the investigators further utilized the `f_classif` score function. Finally, SelectKBest retains the first K features of the input dataset X minus the target variable. In our case the value of k was 10. Using this process the investigators listed the features with top 10 `F_Score` in Table 1.

This was followed by the statistical determination of the presence of outliers [25]. As defined by Zhao [26] an "outlier is considered as a data point which is far from other observations." Here the investigators believe that the presence of outliers may have an impact on the final results of machine learning models. With this in mind, the investigators applied Interquartile range (IQR) to detect the presence of the outliers. Technically, as applied in [27] the IQR is measured as the difference between the third Quartile and the first Quartile i.e.  $IQR = IQ3 - IQ1$ . After applying the operation to remove outliers from the dataset the investigators removed 6,579 entries. The skewness of the dataset was measured. Skewness as indicated by [27] attempts to indicate the normal distribution of the values. Finding outliers and removing them from the dataset is one of the ways of handling skewness, this process was outlined by [29]. Thus, we measure skewness of the selected features before and after removing outliers from our dataset (Table 2).

It can be observed in Table 2 that after removing outliers the skewness of the selected features has reduced. The analysis of binary classification for selected variables for logistic regression, SVM and DLANN is as illustrated in Figure 2.

Metrics applied were K-Fold validation test, confusion matrix metrics and Area Under Curve (AUC). Cross validation is used to gauge the effectiveness of the model. It involves using a sample of the dataset for testing and training the model on the remaining part of the dataset [30]. The value of k determines the number of groups that a data can be split into. In our case we have set the value of k to 10; therefore, the name 10-fold cross-validation.

Additionally, investigators have used a confusion matrix also termed as an error matrix to analyse the performance of a machine learning algorithm in a matrix format [31]. It is as shown in Table 3.

In the confusion matrix, TP stands for true positive, TN stands for true negative, FN stands for false negative and FP stands for false positive. The assumptions made

Feature variable names	F_Score
Number of Services	22369.69
Total Medicare Standardized Payment Amount	22184.17
Total Medicare Allowed Amount	22119.67
Total Submitted Charge Amount	19193.84
proxy for # of new patients	19177.12
Number of Medicare Beneficiaries	18581.63
Average Medicare Standardized Amount per Beneficiary	7535.67
Number of HCPCS	6275.17
Physical therapy services that involve Physical Agents	1998.79
Physical therapy services that involve Therapeutic Practice	1998.79

Table 1: Feature selection based on F-Score.

Feature variable names	With Outliers Skewness	Without Outliers Skewness
Number of HCPCS	0.59	0.26
Number of Medicare Beneficiaries	2.70	0.98
Average Medicare Standardized Amount per Beneficiary	2.05	0.66
Physical therapy services that involve Physical Agents	1.53	1.17
Physical therapy services that involve Therapeutic Practice	-1.53	-1.17
proxy for # of new patients	2.87	0.78
Number of Services	3.96	1.06
Total Submitted Charge Amount	3.97	1.07
Total Medicare Allowed Amount	4.15	1.01
Total Medicare Standardized Payment Amount	4.55	1.05

Table 2: Measuring skewness with and without outliers.

are  $S_{TP}$  denotes the Samples of True Positive,  $S_{TN}$  are the samples which denote True Negatives,  $S_{FP}$  denotes the Samples for False positive and  $S_{FN}$  gives the samples for False Negatives.

```

Input: Medicare data from CSV file
Output: Measure Accuracy Score
1. Select the features using F_Score
# SelectKBest() is a function under
# feature_selection under sklearn library
# f_classif uses Anova F-value for classification
# purposes
2. selec_features ← SelectKBest(f_classif, k = 10)
3..Remove the outliers using Z_Score
# zscore a function available in Scipy python
# package under stats module
4..z ← np.abs(stats.zscore(data))
5..Compute the Skewness to determine normal
distribution of values
#Pandas library in Python to measure unbiased
#skewness.
6.skw ← data.skew()
7. Remove the outliers by identifying anything
that is not the range of lower and upper bound
IQR ← IQR3 – IQR1
l_bound ← IQR1 - (IQR * 1.5)
u_bound ← IQR3 + (IQR * 1.5)
8. Assign X to columns and Y to target
9. Split X and Y into training and testing dataset
in the ratio 80 to 20%
10. Train the models (Logistic,SVM and DLANN
Model)
11. Predict the target for the above models.
12. Compute K-Fold accuracy for the models
# KFold from sklearn library will split data into 10
# folds where 9 folds are used for training and
# 1 fold for validation in an iterative manner;
# random state=7 is seed for random number
# generator
13. kfold ← KFold(n_splits=10, random_state=7)
14. Compute confusion matrix metrics and ROC
for the above models.
15. Plot the area under receiver operating
characteristic curve from the metrics module
under sklearn library
16. auc_score ← metrics.roc_auc_score(y_test,
y_pred_prob)
    
```

Algorithm 1.

**Accuracy** of the classification model [32] is determined in the present study by correctness of the confusion matrix and is as given in Equation 1.

$$Accuracy_{model} = \frac{(S_{TP} + S_{TN})}{S_{Total}} \quad (1)$$

where  $Accuracy_{model}$  gives the classification accuracy. A higher accuracy of 99% is good but at times it also depends on the dataset.

**Precision** of the classification model gives the percentage the correct results among all the returned results and is as given in Equation 2

$$Precision_{model} = \frac{S_{TP}}{S_{TP} + S_{FP}} \quad (2)$$

where  $Precision_{model}$  gives precision of a machine learning model for classification problem

	Predicted	
Actual	TP	FP
	FN	TN

Table 3: Layout of confusion matrix.

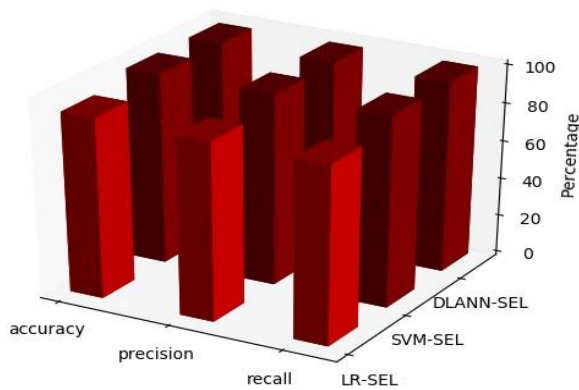


Figure 2: Confusion matrix metrics for logistic regression, SVM and DLANN for selected feature dataset.

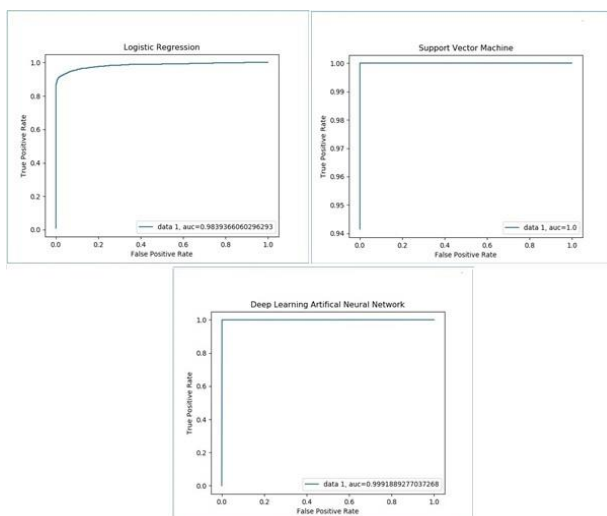


Figure 3: AUC for logistic regression, SVM and DLANN for selected feature dataset.

**Recall** is the capacity of the model to find data points of interest and is as given in Equation 3

$$Recall_{model} = \frac{S_{TP}}{S_{TP} + S_{FN}} \quad (3)$$

where  $Recall_{model}$  gives the correct classification of positive samples by the machine learning model for the given binary classification problem.

One of the limitations of accuracy is its constraints in terms of test sample size which in our experiments has been considered as 20%. Thus, for a binary classifier as in our experiments, where we have pitted true positives against false negatives; Area Under Curve (AUC) gives a more generic approach as it evaluates the binary classifier model for random guesses. Thus, AUC provides a better perceived measure as compared to accuracy which is more tightly coupled to a threshold. In an event when accuracy cannot be used to clearly distinguish machine learning models AUC can work as an alternative deciding parameter [33]. The experimentation conducted provided K-Fold validation scores of 94.10% and 99.97% for logistic regression and SVM respectively.

Thus, the K-Fold accuracy of SVM is superior to Logistic Regression by 12.15%. We now consider the confusion matrix metrics for the selected feature dataset as illustrated in Figure 2.

It is further observed in Figure 2 that SVM was the top performer and marginally outscored DLANN which is an interesting observation which needs to be analysed further.

Figure 3 shows AUC for Logistic Regression, SVM and DLANN. From Figure 3 it is observed that SVM has the highest AUC of 1.0 followed by DLANN with an AUC of 0.99. The AUC of Logistic Regression is relatively least with a score of 0.98.

### 3.2 Computational time analysis

Based on the observations made from the binary classifier model it becomes imperative that apart from scoring high on accuracy and other associated metrics computational efficiency on resource constrained IoT environment is a necessary attribute for a low-cost data analysis system. Therefore, the investigators decided to compare the computational time of each model used for analysis. The hardware platform used for this aspect of analysis was a Raspberry Pi with Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM. The results of this analysis is as illustrated in Table 4.

As mentioned before, the application of feature selection and removal of outliers led to the reduction of dataset size from 8.7 MB to 2.9 MB. Therefore, it is common sense that for a dataset with selected variables the computational time will be naturally lower. This is of significance for resource constrained environments IoT environments such as the Raspberry Pi. It is observed under dataset with selected variables Logistic Regression outperforms SVM by 99.04% and DLANN by 98.02%. This clearly indicates that Logistic Regression is most computationally efficient as compared to SVM and DLANN. Also, SVM outperformed DLANN and Logistic Regression in terms of AUC, confusion matrix metrics and

Binary classifier Model	Computational Time in seconds
	Raspberry Pi B
Logistic Regression – Selected Dataset	37.81
SVM – Selected Dataset	3949.02
DLANN – Selected Dataset	1918.59

Table 4: Computational time of machine learning and deep learning models.

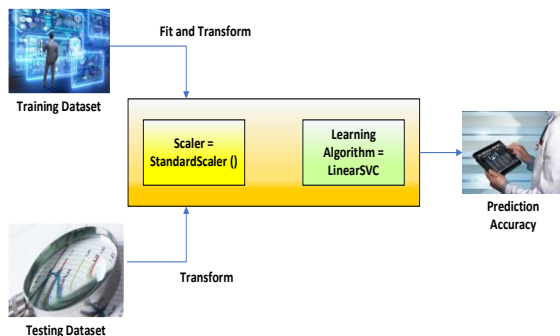


Figure 4: Pipeline architecture for SVM on Raspberry Pi.

```

Output: Pipelined Architecture of SVM
1. pipe_lrSVC ← Pipeline(['scaler',
StandardScaler()), ('clf', LinearSVC())]) #Build the
pipeline
2. r ← pipe_lrSVC.fit(X_train, y_train)
3. y_pred ← pipe_lrSVC.predict(X_test) #predict
    
```

Algorithm 2

```

Input : newmeddata.csv # The original dataset
Output: noisy_data.csv # The noisy dataset
1. σ ← 0.1 # standard deviation is 0.1
2. μ ← 0 # mean is 0
3. noise ← actual_data + σ * random (size
(actual_data)) + μ
4. noisy_data.csv ← actual_data + noise
#noisy_data.csv is the data with added Gaussian
noise
5. target_variable ← int (actual_target_variable +
noise)
    
```

Algorithm 3

K-Fold validation tests. This motivated the investigators for further analysis where they built a model where SVM provides robust performance and also is computationally time efficient.

### 3.3 Pipelined support vector machine architecture and Gaussian noise

Pipeline allows us to fit a model by combining a number of transformations and executing a predictor once. The software pipeline architecture as provided by scikit-learn [24] is as illustrated in Figure. 4.

In Python the Pipeline class [34] allows the collation of multiple processes into a single estimator. Therefore, we can fit the pipeline to the whole training data and also transform it to test data without the need for doing the same individually. Linear Support Vector Classification abbreviated as LinearSVC uses a linear kernel, is faster and can also scale rapidly. These parameters were fed to the pipeline to reduce the computational time required for SVM on raspberry pi.

The algorithm implemented in our model of pipelined SVM is as illustrated in Algorithm 2.

Here Gaussian noise is added to the dataset to benchmark the performance of SVM against Logistic Regression and Deep Learning Artificial Neural Network. The presence of Additive Gaussian Noise [35][36] is known to have impact on the distribution of the data. To check the robustness of the different classifier models a common data corruption technique through Gaussian noise was applied. Many such analysis were conducted in [37] to benchmark neural network robustness. In our work the noise signal was set with mean 0 and standard deviation of 0.1. To simulate the Gaussian Noise the NumPy Random Normal function was used, which generates values from the Gaussian distribution. The values assumed for  $\mu$  was 0 and  $\sigma = 0.1$ . The additive noise is as generalized [38] in Equation 4.

$$M_{Rno,Fno} = O_{Rno,Fno} + \epsilon_{Rno,Fno} \quad (4)$$

where  $M_{Rno,Fno}$  is the modified data point;  $O_{Rno,Fno}$  is the original data point and  $\epsilon_{Rno,Fno}$  is the random noise approximately equal to the distribution  $(\mu, \sigma^2)$ ; where  $\mu$  is mean and  $\sigma^2$  is the variance. The algorithm for Gaussian Noise implementation is illustrated in Algorithm 3.

The analysis of K-Fold validation tests came with a result of 78.88% for Logistic Regression and 78.58% for SVM which indicated the similar performance of both the models in presence of Gaussian Noise. The performance of Logistic Regression dropped by 15.22% and performance of SVM dropped by 21.39%. This clearly indicates that in the presence of noise logistic regression performed at an acceptable level. We further continued our experiments for results with confusion metrics as illustrated in Figure 5.

From Figure 5 it is observed that the performance of SVM in terms of accuracy is least 58.44% in presence of Gaussian Noise. The precision of Logistic Regression and DLANN was good and exhibited similar performances of 49.79%. and 50.79%. However, it could be observed that the precision was one of the worst affected metrics and the performance for Logistic Regression dropped by 46.8%, SVM by 66.53%, and DLANN by 49.11%. This performance was compared with performances of machine learning models run on dataset with selected features. A low precision for SVM could basically indicate a large number of false positives. On the contrary, a high value of recall of 99.16% indicates that SVM was very sensitive and could successfully identify true positive observations. The analysis was continued for AUC metric.

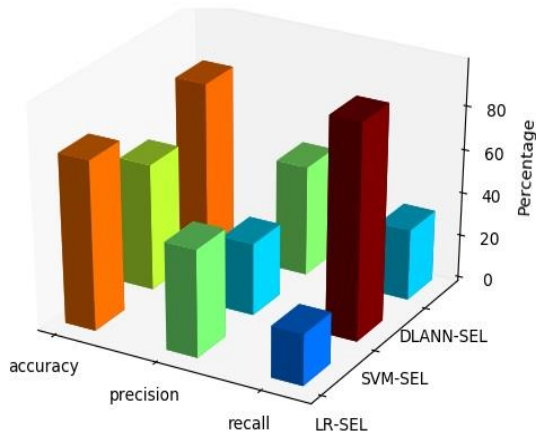


Figure 5: Confusion matrix metrics for logistic regression, SVM and DLANN for selected feature dataset and with gaussian noise.

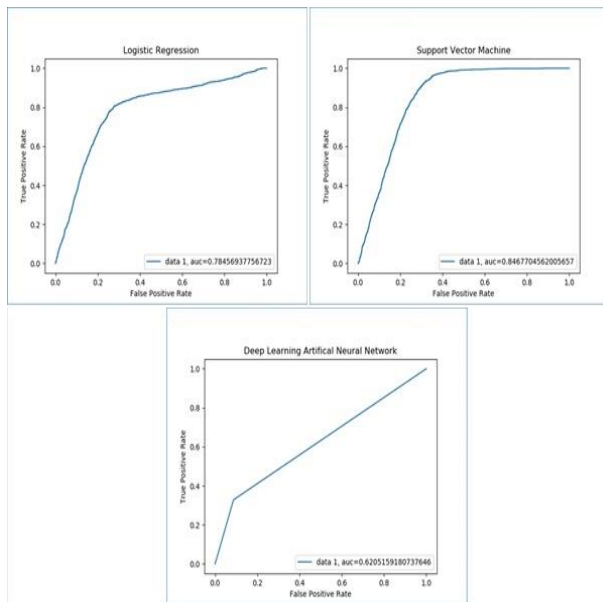


Figure 6: AUC for logistic regression, SVM and DLANN for selected feature dataset with gaussian noise.

Binary classifier Model	Computational Time in seconds
	Raspberry Pi B
Logistic Regression – Selected Dataset with Gaussian Noise	23.57
SVM – Selected Dataset with Gaussian Noise	382.34
DLANN – Selected Dataset – Gaussian Noise	1887.36

Table 5: Computational time of machine learning and deep learning models.

From Figure 6 the investigators observe that the performance of SVM is better compared to other models. It outscores Logistic Regression by 7.2%, and DLANN model by 22.65%.

### 3.3.1 Computational time analysis

As indicated in the introduction section the investigators performed the computational time analysis of different methods. This computational time analysis is illustrated in Table 5.

Here we observe that Logistic Regression was the most computationally efficient in terms of execution time. However, with a pipeline SVM outperformed its nearest competitor DLANN by 79.7 4% and was inferior to Logistic Regression by 93.83%. Therefore, SVM improved its performance in terms of computational execution time. Additionally, it was observed that in presence of Gaussian Noise, the accuracy of most of the models dropped and DLANN emerged as slight winner with little bit of consistency and SVM exhibited low recall and high precision thereby exhibiting its fitness for the dataset under consideration. Also, the proposed Pipelined model of SVM achieved a better performance in terms of computational time to its nearest competitor the DLANN model.

## 4 Discussion

The investigators in the present work implemented a pipeline SVM model to test it against known benchmarks of Logistic Regression and Deep Learning Neural Network for performance optimization in terms of computational time and accuracy metrics for a resource constrained environment of Raspberry Pi. Therefore, the investigators explored statistical technique of F Score for feature selection and could shortlist top 10 features. The investigators further processed outliers by applying Inter Quartile Range. This helped the investigators to balance the skewness of the data. Thus, the modified dataset with reduced storage requirements was tested on Raspberry PI for machine learning models like logistic regression, SVM and DLANN for binary classification and performance benchmarking. K-Fold accuracy of SVM was superior to Logistic Regression by 12.15%. Confusion matrix metrics where further applied to test the machine learning models and SVM achieved better performance and at times was at par with Deep Learning Neural Network. The uniqueness of the present work is that it dealt with the training time that SVM takes which is usually large. Thus reducing training time was of paramount importance as the platform were, SVM was to be implemented was Raspberry Pi. This was achieved by implementing SVM with a pipelined architecture. Thus SVM achieved a better performance in terms of computational time to its nearest competitor the DLANN model by 79.74%. SVM is prone to noise, thus the optimized and pipelined architecture of SVM was benchmarked with Deep Learning in the presence of Gaussian noise. The accuracy of most of the models dropped and DLANN emerged as slight winner with little bit of consistency and SVM exhibited low recall and high precision thereby exhibiting its fitness for the dataset under consideration. The better accuracy of DLANN with selected features and under noise may be attributed to the fact that noise could have added as a regularization factor thus boosting the performance of DLANN. This clearly provides some pathway for future work in terms of

extending pipeline architectures for Deep Learning algorithms [39],[40],[41], which are efficient but slow and are visualized for working in resource constrained environments of IoT.

#### 4.1 Limitations of the present work

The analysis was considered for a single medical dataset. In future the capabilities of the models could be generalized for a range of datasets. With parallel environments for machine learning models and with IoT clusters based on graphical processing units (GPU's) for remote computing the models could be made much more computationally feasible. Also, techniques like PCA for feature selection and its interaction for deep learning algorithms was not explored in the present work.

## 5 Conclusion

Overall, the investigators conclude that SVM exhibited its robustness in terms of relatively good performances for all computational setups of optimized, and corrupted datasets in resource constrained environments of IoT. The impact of additive noise had distressing effects on most models and may be a concern in an environment where devices collect data from sensors. As stated, the analysis was conducted on a single dataset thereby limiting the validation of the conclusions derived. The feature selection of dataset resulted in reduction of dataset size by 67% but had a minor loss in terms of accuracy of the classifier models like Logistic Regression, SVM and DLANN. Therefore, we can safely suggest that SVM had a relatively stable performance across all the scenarios and at times was better than DLANN model. Additionally, we suggest that pipeline architectures and automating machine learning models had a good impact on resource constrained environments like Raspberry Pi. SVM pipelined model outscored DLANN model by 79.94% for a featured selected and Gaussian noise added dataset in terms of computational time. Thereby, the investigators have concluded SVM as the model of choice for analysing similar datasets. Therefore, the core contributions of this work were: i) implementing a pipelined Support Vector Machine model for performance benchmarking against Logistic Regression and Deep Learning Neural Network for computational time efficiency and accuracy metric for a relatively largest dataset, and ii) a brief analysis of computational time analysis for these general methods for SVM using Raspberry Pi. In future, the investigators would like to explore how the machine learning and deep learning models that can detect noise and outliers and automatically improve their learning abilities for complex pipelined models, in a constrained environment of an IoT device enabled by Graphics Processing Unit (GPU).

#### Acknowledgments

The authors would like to thank the School of Global Health Management and Informatics for the permission to use the University of Central Florida (UCF), Decision Support Systems and Informatics Laboratory facilities to conduct the research work and related documentation.

Research Study	Analysis Techniques	Results
This project	Pipelined Support Vector Machine, Logistic Regression and Deep Learning Artificial Neural Network on Raspberry Pi environment	Pipelined SVM achieved a better performance in terms of computational time measurement to its nearest competitor the DLANN model by 79.74%.
Sheets et.al, 2017 [17]	Combination of contrast mining and Logistic Regression was used.	Electronic Health Record (EHR) contrast mining with Logistic Regression predicted 5% of patients contributing to 50% of healthcare expenses.
Nalepa & Kawulok J., 2018 [9]	Trained Support Vector Machine for large datasets with different kernels.	SVM has been successful in solving a variety of pattern recognition tasks; its main drawbacks were the huge time and memory related complexities.
Sakr et.al, 2016 [18]	Deep Learning Convolutional Neural Network (CNN) and Support Vector Machine	SVM model achieved high classification accuracy of 94.8% while CNN could achieve 83%
Pei et.al., 2021 [21]	Deep Learning Convolutional Neural Network (CNN) and White Gaussian Noise	Classification performance of Deep Learning CNN drops significantly when noise is added.

Table 6: Comparison of research projects and analysis methods.

## References

- [1] Allhoffa F. & Henschke A (2018). The Internet of Things: Foundational ethical issues, *Internet of Things*, pp. 55–66. <https://doi.org/10.1016/j.iot.2018.08.005>
- [2] Haller S., Karnouskos S., & Schroth C (2009). "The Internet of Things in an Enterprise Context," in *Future Internet – FIS 2008, Lecture Notes in Computer Science*, vol. 5468, pp 14-28. [https://doi.org/10.1007/978-3-642-00985-3\\_2](https://doi.org/10.1007/978-3-642-00985-3_2)
- [3] Zhang Z-K., Cho M , Wang C-W., Hsu C-W, Chen C-K, & Shieh S (2014). IoT Security: Ongoing Challenges and Research Opportunities,



- Proceedings of the 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, pp. 230-234.  
<https://doi.org/10.1109/SOCA.2014.58>
- [4] Gokhale P., Bhat O., Bhat S (2018). Introduction to IOT, *International Advanced Research Journal in Science, Engineering and Technology*, vol. 5(1), pp. 41- 44.  
<https://doi.org/10.17148/iarjset.2018.517>
- [5] Wan T.T.H, Gurupur V (2020). Understanding the Difference between Healthcare Informatics and Healthcare Data Analytics in the Present State of Health Care Management, *Health Services Research & Managerial Epidemiology*, vol. 7, pp. 1-3.  
<http://dx.doi.org/10.1177/2333392820952668>
- [6] Shukla S., Hassan M.F., Khan M.K., Jung L.T., Awang A (2019).An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment, *PLoS ONE*, pp.1-31.  
<http://dx.doi.org/10.1371/journal.pone.0224934>
- [7] Noble W.S (2006). What is a support vector machine? *Nature Biotechnology*, Vol.24, pp. 1565–1567.  
<https://doi.org/10.1038/nbt1206-1565>
- [8] Bradley A.P (1997).The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms, *Pattern Recognition*, vol. 30(7), pp. 1145-1159.  
[https://doi.org/10.1016/S0031-3203\(96\)00142-2](https://doi.org/10.1016/S0031-3203(96)00142-2)
- [9] Nalepa J. , Kawulok M (2019). Selecting training sets for support vector machines: a review. *Artif Intell Rev* 52, pp. 857–900.  
<https://doi.org/10.1007/s10462-017-9611-1>
- [10] Papadonikolakis M., Bouganis C. & Constantinides G (2009). "Performance comparison of GPU and FPGA architectures for the SVM training problem," *2009 International Conference on Field-Programmable Technology*, pp. 388-391.  
<https://doi.org/10.1109/FPT.2009.5377653>
- [11] Huang C-L, Chen M-C, Wang C-J (2007). Credit scoring with a data mining approach based on support vector machines, *Expert Systems with Applications*, vol. 33, pp. 847–856  
<https://doi.org/10.1016/j.eswa.2006.07.007>
- [12] Yazici M T. , Basurra S. & .Gaber M M (2018). Edge Machine Learning: Enabling Smart Internet of Things Applications, *Big Data and Cognitive Computing*, vol. 2: 26, pp. 1-17.  
<https://doi.org/10.3390/bdcc2030026>
- [13] Nguyen M H. Torre F de la (2010). Optimal feature selection for support vector machines, *Pattern Recognition*, vol.43, pp. 584–591  
<https://doi.org/10.1016/j.patcog.2009.09.003>
- [14] Sanz H, Valim C., Vegas E, Oller J M. & Reverter F (2018). SVM-RFE: selection and visualization of the most relevant features through non-linear kernels, *BMC Bioinformatics*, vol. 19:432, pp 1-18.  
<https://doi.org/10.1186/s12859-018-2451-4>
- [15] Gurupur V. P, Kulkarni S. A., Liu X., Desai U., & Nasir A (2018). Analysing the power of deep learning techniques over the traditional methods using medicare utilisation and provider data, *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 99-115.  
<https://doi.org/10.1080/0952813X.2018.1518999>
- [16] Zardo P., Collie A (2014). Predicting research use in a public health policy environment: results of a logistic regression analysis, *Implementation Science*, vol. 9, pp. 1-10.  
<https://doi.org/10.1186/s13012-014-0142-8>
- [17] Sheets L., Petroski G.F., Zhuang Y., Phinney M.A., Ge B, Parker J.C., Shyu C-R (2017). Combining Contrast Mining with Logistic Regression to Predict Healthcare Utilization in a Managed Care Population, *Applied Clinical Informatics*, vol. 8: 2, pp. 430-446.  
<https://doi.org/10.4338/aci-2016-05-ra-0078>
- [18] Sakr G. E, Mokbel M., Darwiche A., Khneisser M. N & Hadi A (2016). Comparing deep learning and support vector machines for autonomous waste sorting, *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pp. 207-212.  
<https://doi.org/10.1109/IMCET.2016.7777453>
- [19] Ravi D., Wong C., Deligianni F., Berthelot M., Andreu-Perez J., Lo B., & Yang G-Z (2017). Deep Learning for Health Informatics, *IEEE Journal of Biomedical and Health Informatics*, vol. 21: (1), pp.4-21.  
<https://doi.org/10.1109/jbhi.2016.2636665>
- [20] Gangsar P. & Tiwari R (2018). Effect of noise on support vector machine based fault diagnosis of IM using vibration and current signatures, *MATEC Web of Conferences*, vol. 211.  
<http://dx.doi.org/10.1051/mateconf/201821103009>
- [21] Pei Y., Huang Y., Zou Q., Zhang X. & Wang S (2021). Effects of Image Degradation and Degradation Removal to CNN-Based Image Classification," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43: 4, pp. 1239-1253.  
<https://doi.org/10.1109/TPAMI.2019.2950923>
- [22] Wu X. & Zhu X (2008). Mining with Noise Knowledge: Error-Aware Data Mining, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol.38: (4), pp.15-19.  
<https://doi.org/10.1109/CIS.2007.7>
- [23] Zualkernan A., Zualkernan I A., Dhou S, Judas J, Sajun A R, Gomez B R., Hussain L A., Sakhnini D (2020), Towards an IoT-based Deep Learning Architecture for Camera Trap Image Classification, *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, pp. 1-6.  
<https://doi.org/10.1109/GCAIoT51063.2020.9345858>
- [24] Scikit-learn Machine Learning in Python. [Online]. Available: <https://scikit-learn.org/stable/>
- [25] Tukey J (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading MA
- [26] Zhao Q., Zhou G., Zhang L., Cichocki A. & Amari S (2016).Bayesian Robust Tensor Factorization for

- Incomplete Multiway Data, *IEEE Transactions on Neural Networks and Learning Systems*, vol.27:(4), pp.736-748  
<http://dx.doi.org/10.1109/TNNLS.2015.2423694>
- [27] Khan Z., Naeem M., Khalil U., Khan D. M., Aldahmani S. & Hamraz M (2019). Feature Selection for Binary Classification Within Functional Genomics Experiments via Interquartile Range and Clustering, *IEEE Access*, vol. 7, pp.78159-78169.  
<https://doi.org/10.1109/ACCESS.2019.2922432>
- [28] Yusoff S. B. & Wah Y. B (2012). Comparison of conventional measures of skewness and kurtosis for small sample size, *2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE)*, pp.1-6.  
<https://doi.org/10.1109/ICSSBE.2012.6396619>
- [29] Heymann S., Latapy M. & Magnien C (2012). Outskewer: Using Skewness to Spot Outliers in Samples and Time Series, *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp.527-534.  
<https://doi.org/10.1109/ASONAM.2012.91>
- [30] Xu L., Hu O., Guo Y., Zhang M., Lu D., Cai C. B., Xie S., Goodarzi M., Fu H. Y., She Y. B (2018). Representative splitting cross validation, *Chemometrics and Intelligent Laboratory Systems*, vol.183, pp.29-35.  
<https://doi.org/10.1016/j.chemolab.2018.10.008>
- [31] Tharwat A (2018). Classification assessment methods, *Applied Computing and Informatics*, pp.1-13.  
<https://doi.org/10.1016/j.aci.2018.08.003>
- [32] Fatourehchi M., Ward R. K., Mason S. G., Huggins J., Schlög A., & Birch G. E (2008). Comparison of Evaluation Metrics in Classification Applications with Imbalanced Datasets, *Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications*, pp.777 – 782.  
<https://doi.org/10.1109/ICMLA.2008.34>
- [33] Huang J. & Ling C (2005). Using AUC and Accuracy in Evaluating Learning Algorithms, *IEEE Transactions on Knowledge & Data Engineering*, vol.17:(3), pp.299-310.  
<https://doi.org/10.1109/TKDE.2005.50>
- [34] Pipelines and composite estimators, <https://scikit-learn.org/stable/modules/compose.html>
- [35] Nadarajah S. & Kotz S (2007). On the Generation of Gaussian Noise, *IEEE Transactions on Signal Processing*, vol. 55 (3), pp.1172-1172.  
<http://dx.doi.org/10.1109/TSP.2006.888061>
- [36] Zhuang L. & Ng M. K (2020). Hyperspectral Mixed Noise Removal By  $\ell_1$ -Norm-Based Subspace Representation, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.13 , pp.1143-1157.  
<https://doi.org/10.1109/JSTARS.2020.2979801>
- [37] Hendrycks D., & Dietterich T. G (2018). Benchmarking Neural Network Robustness to Common Corruptions and Surface Variations, *arXiv: Learning*, pp.1-13  
<https://arxiv.org/abs/1807.01697v5>
- [38] Domingo-Ferrer J., Seb'e F., & Castell`a-Roca J (2004). On the Security of Noise Addition for Privacy in Statistical Databases, *International Workshop on Privacy in Statistical Databases*, pp.149-161.  
[http://dx.doi.org/10.1007/978-3-540-25955-8\\_12](http://dx.doi.org/10.1007/978-3-540-25955-8_12)
- [39] Yao S., Zhao Y., Zhang A., Hu S., Shao H., Zhang C., Su L., Abdelzaher T (2018). Deep Learning for the Internet of Things, *Computer*, vol. 51: 5, pp. 32-41.  
<https://doi.org/10.1109/MC.2018.2381131>
- [40] Ma X., Yao T., Hu M., Dong Y., Liu W., Wang F., Liu J (2019). A Survey on Deep Learning Empowered IoT Applications, in *IEEE Access*, vol. 7, pp. 181721-181732.  
<https://doi.org/10.1109/ACCESS.2019.2958962>
- [41] Ahmed I., Din S., Jeon G., Piccialli F (2020). Exploring Deep Learning Models for Overhead View Multiple Object Text of the second section, in *IEEE Internet of Things Journal*, vol. 7: 7, pp. 5737-5744.  
<http://dx.doi.org/10.1109/JIOT.2019.2951365>

# Textual Entailment for Modern Standard Arabic

Maytham Alabbas

Department of Computer Science, College of Computer Science and Information Technology

University of Basrah, Basrah, Iraq

E-mail: ma@uobasrah.edu.iq, http://faculty.uobasrah.edu.iq/faculty/534

## Thesis summary

**Keywords:** recognizing textual entailment, extended tree edit distance, tree edit distance, arabic textual entailment

**Received:** June 12, 2021

*This paper summarizes the Doctoral Thesis that examines various techniques to recognizing Arabic textual entailment, deciding whether one fragment of text entails another, where there is an exceptional level of structural and lexical ambiguities. As far as we know, the current work is the first study to apply this task for Arabic. For this purpose, we firstly describe a semi-automatic method for constructing a first Arabic textual entailment dataset. Then, we have investigated various system combination techniques for improving tagging and parsing depending on having accurate linguistic analyses. Finally, we have improved the standard tree edit distance (TED) algorithm. This extended version of TED, ETED, calculates the distance between two trees by applying operations on subtrees and single nodes. The current work also uses the artificial bee colony (ABC) algorithm to automatically guess the edit operations cost for both subtrees and single nodes and to decide thresholds. The current findings were encouraging for Arabic and English RTE-2 test sets. It should be noted most of the methodologies presented here could be utilized in research projects on poorly resources languages.*

*Povzetek: Predstavljena je doktorska disertacija za obdelavo arabskih besedil.*

## 1 Introduction

One of the essential tasks for natural language systems is to decide whether one natural text snippet entails another. Nowadays, textual entailment (TE) is considered as one of the most popular generic tasks in this regard. TE can be described as a relation between two natural sentences in which one sentence's truth, the entailing expression (T), compels the truth of the other, what is entailed (H). For instance, 'The president was assassinated.' entails 'The president is dead.', whereas the reverse does not hold.

TE definition contrasts with the standard entailment definition, i.e., T entails H if H is true whenever T is. The TE recognition task is in some ways easier than the classical entailment task. It has led to different techniques that diverge from the tradition of translating from natural language into logical forms and using standard approaches of theorem proving to determine the relationships between these logical forms.

## 2 Methodology

The current work [1,2] aims to see how well existing approaches for recognizing textual entailment (RTE) work when utilized to the Arabic language; and to provide suggestions for improvements that teat with the particular issues posed by the language. We used the TE architecture system that is illustrated in Figure 1. At each stage, we aimed to take advantage of variations on the standard machinery to assist us in overcoming the additional challenges posed by written Arabic.

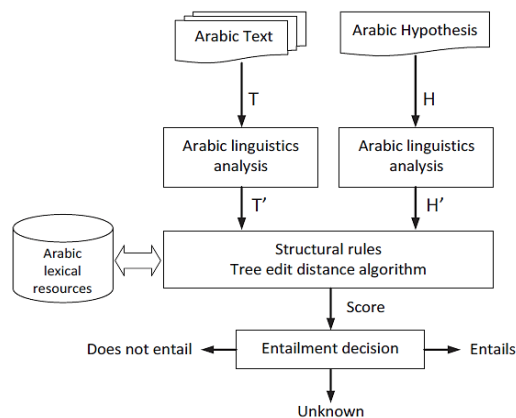


Figure 1: General diagram of current system [1].

### 2.1 Arabic linguistic analysis

Such a system depends on the presence of accurate linguistic analyses. It is notoriously difficult to obtain such analyses for the Arabic language. Concerning these problems, we looked into solutions that used system combination strategies to improve tagging and parsing to overcome these issues [3]. These strategies outperform any of the contributing tools by a significant margin [4]. A tagger and a parser are implemented as preprocessing tools to represent each sentence as dependency trees. We use the method described by [5] of merging the three taggers (MADA, AMIRA, and a maximum-likelihood tagger) based on their confidence levels, using the built-in

tokenizer from MADA to preprocess the text. In [6], we show that the combination strategy achieves 99.5% accuracy for the ‘Bies’ tagset. We then use a combination of three parsers (MSTParser plus two MALTParser algorithms) as described by [7]. This gives around 85% for labeled accuracy, which is the best Penn Arabic treebank (PATB) result we have seen. We apply these combinations in all our series of experiments.

## 2.2 Arabic TE dataset

To evaluate our Arabic TE system, an appropriate dataset is required. As far as we know, no Arabic datasets are available for the TE task; therefore, we have had to create one. We have utilized one of the approaches applied for collecting the T-H pairs in the RTE tasks, with a slight alteration. We developed in [8] a semi-automatic approach for producing a first Arabic textual entailment dataset relying on an improved version of the ‘headline-lead paragraph’ technique. We outlined the challenges that come with depending on volunteer inter-annotators to make the judgment and developed a regime to address some of these issues. There are 600 pairings in the preliminary testing dataset, each with a binary annotation of ‘yes’ or ‘no’ (a 50-50 split). This dataset is similar in size to the RTE-2 dataset but with typically longer sentences.

## 2.3 Tree matching

We investigate various systems for the task of Arabic TE, starting with basic and reliable but approximate systems and proceeding to more advanced systems. There are two primary groups of these systems [1]: surface string similarity systems (bag-of-words system and Levenshtein distance systems) and syntactic similarity systems (tree edit distance systems, our extended version of TED with subtree operations systems (ETED) [9,10,11], hybrid ETED with optimization algorithms such as ABC algorithm). Six systems out of 10 are reimplementations of existing methods that have been implemented for other languages. These serve as baselines and indicate that when applied to Arabic, the findings are comparable to those achieved with English. While four systems cover our contributions, each representing a distinct version of our system.

## 2.4 Entailment decision

This part is responsible for making the final entailment decision depending on the final score. To evaluate if this score should lead to a certain judgment, one threshold (entails/not-entail tests) or two thresholds (entails/unknown/not-entail tests) are utilized [9].

## 3 Conclusion

The current findings were extremely encouraging on the Arabic test set, notably the F-score improvement. The fact that some of these findings were replicated for the RTE2 test set, where we did not have any control over the dependency trees parser, gives some evidence for the

current approach’s robustness [1]. In both circumstances, we anticipate that having a more accurate parser (our Arabic parser achieves approximately 84% accuracy on PATB, whereas MINIPAR is estimated to reach around 80% accuracy on the Suzanne tested corpus) would improve the performance of both versions of TED.

## References

- [1] Alabbas, M. (2013). Textual Entailment for Modern Standard Arabic. PhD Thesis, The University of Manchester, Manchester, UK.
- [2] Alabbas, M. (2011). ArbTE: Arabic textual entailment. RANLP Student Research Workshop 2011, Hissar, Bulgaria, pp. 48–53.
- [3] Alabbas, M. and Ramsay, A. (2012). Combining black-box taggers and parsers for modern standard Arabic. In Proceedings FedCSIS-2012, IEEE, Wrocław, Poland, pp. 19–26.
- [4] Alabbas, M. and Ramsay, A. (2014). Combining strategies for tagging and parsing Arabic. In Proceedings of the EMNLP 2014 Workshop on ANLP 2014, pp. 73–77, doi:10.3115/v1/W14-3609.
- [5] Alabbas, M. and Ramsay, A. (2012). Improved POS-tagging for Arabic by combining diverse taggers. In Proceedings of AIAI, volume 381, Springer Berlin, pp. 107–116, doi: 10.1007/978-3-642-33409-2\_12.
- [6] Alabbas, M. and Ramsay, A. (2014). Improved Parsing for Arabic by Combining Diverse Dependency Parsers. LTC 2011, Revised Selected Papers, Lecture Notes in Computer Science, Springer, Vol. 8387, pp. 43–54, doi:10.1007/978-3-319-08958-4\_4.
- [7] Alabbas, M. and Ramsay, A. (2011). Evaluation of combining data-driven dependency parsers for Arabic. In Proceedings of LTC 2011, Poznań, Poland, pp. 546–550.
- [8] Alabbas, M. (2013). A dataset for Arabic textual entailment. RANLP Student Research Workshop 2013, Hissar, Bulgaria, pp. 7–13.
- [9] Alabbas, M. and Ramsay, A. (2013). Natural language inference for Arabic using extended tree edit distance with subtrees. Journal of Artificial Intelligence Research, 48:1-22, doi:10.1613/jair.3892.
- [10] Alabbas, M. and Ramsay, A. (2013). Optimising tree edit distance with subtrees for textual entailment. In Proceedings of RANLP2013, Hissar, Bulgaria, pp. 9–17.
- [11] Alabbas, M. and Ramsay, A. (2012). Dependency tree matching with extended tree edit distance with subtrees for textual entailment. In Proceedings of FedCSIS-2012, IEEE, Wrocław, Poland, pp. 11–18.

## CONTENTS OF *Informatica* Volume 45 (2021) pp. 1–658

### Papers

- TARUNA & , H.D. ARORA, V. KOMAR. 2021. Study of Fuzzy Distance Measure and Its Application to Medical Diagnosis. *Informatica* 45:143–148.
- ABAKER, A.A. & , F.A. SAEED. 2021. A Comparative Analysis of Machine Learning Algorithms to Build a Predictive Model for Detecting Diabetes Complications. *Informatica* 45:117–125.
- ABDALLAH BENSALLOUA, C. & , A. BENAMEUR. 2021. Towards NoSQL-based Data Warehouse Solution Integrating ECDIS for Maritime Navigation Decision Support System. *Informatica* 45:415–431.
- ALABBAS, M. & . 2021. Textual Entailment for Modern Standard Arabic. *Informatica* 45:653–654.
- ALAM, T. & . 2021. IBchain: Internet of Things and Blockchain Integration Approach for Secure Communication in Smart Cities. *Informatica* 45:477–486.
- ALDUKHAIL, M. & . 2021. Evaluation of Multimedia User Interface Design Method for M-learning (MobLearn): A Comparative Study. *Informatica* 45:633–641.
- ALQUDAH, A.M. & , M. AL-HASHEM, A. ALQUDAH. 2021. Reduced Number of Parameters for Predicting Post-Stroke Activities of Daily Living Using Machine Learning Algorithms on Initiating Rehabilitation. *Informatica* 45:571–581.
- ALSAIF, S.A. & , A. HIDRI. 2021. Impact of Data Balancing During Training to Predict the Risk of Over-Indebtedness. *Informatica* 45:223–230.
- AMPOMAH, E.K. & , G. NYAME, Z. QIN, P.C. ADDO, E.O. GYAMFI, M. GYAN. 2021. Stock Market Prediction with Gaussian Naïve Bayes Machine Learning Algorithm. *Informatica* 45:243–256.
- AZEROUAL, O. & , M.J. ERSHADI, A. AZIZI, M. BANIHASHEMI, R.E. ABADI. 2021. Data Quality Strategy Selection in CRIS: Using a Hybrid Method of SWOT and BWM. *Informatica* 45:65–80.
- BAADEL, S. & , F. THABTAH, J. LU. 2021. Cybersecurity Awareness: A Critical Analysis of Education and Law Enforcement Methods. *Informatica* 45:335–345.
- BANG, B.H. & . 2021. A Metaheuristic for the Bounded Single-Depot Multiple Traveling Repairmen Problem. *Informatica* 45:93–104.
- BHARIMALLA, P.K. & , H. CHOUDHURY, S. PARIDA, D.K. MALLICK, S.R. DASH. 2021. A Blockchain and NLP Based Electronic Health Record System: Indian Subcontinent Context. *Informatica* 45:605–616.
- BITAT, A. & , S. MERNIZ. 2021. Formal Verification of Pipelined Cryptographic Circuits: A Functional Approach. *Informatica* 45:583–591.
- BOSE, S.K. & . 2021. Routing Algorithm in Networks on the Globe. *Informatica* 45:297–302.
- BOUDJIDJ, A. & . 2021. Towards a Formal Multi-Agent Organizational Modeling Framework Based on Category Theory. *Informatica* 45:277–288.
- BOUMAZA, K. & . 2021. Formal Verification of Emergent Properties. *Informatica* 45:463–475.
- CHEN, F. & , S. ZHANG. 2021. Information Visualization Analysis of Public Opinion Data on Social Media. *Informatica* 45:157–162.
- CHERBAL, S. & . 2021. Load Balancing Mechanism Using Mobile Agents. *Informatica* 45:257–266.
- DEBBI, H. & . 2021. Modeling and Performance Analysis of Resource Provisioning in Cloud Computing using Probabilistic Model Checking. *Informatica* 45:529–541.
- DHAINI, M. & , M. JABER, A. FAKHERELDINE, S. HAMDAN, R. A. HARATY. 2021. Green Computing Approaches - A Survey. *Informatica* 45:1–12.
- DÖMÖSI, P. & , B. BORSOS, Y. ALHAMMADI, N. TIHANYI, J. GÁLL, G. HORVÁTH. 2021. A Pseudorandom Number Generator with Full Cycle Length Based on Automata Compositions. *Informatica* 45:179–189.
- GADRI, S. & . 2021. Developing an Efficient Predictive Model Based on ML and DL Approaches to Detect Diabetes. *Informatica* 45:433–440.
- GJORESKI, M. & . 2021. A Method for Combining Classical and Deep Machine Learning for Mobile Health and Behavior Monitoring. *Informatica* 45:169–170.
- GRUBER, I. & , M. ŽELEZNÝ, A. KARPOV. 2021. Heterogeneous Face Recognition from Facial Sketches. *Informatica* 45:487–488.
- HARAHSHEH, H. & , M. SHRAIDEH, S. SHARAEH. 2021. Performance of Malware Detection Classifier Using Genetic Programming in Feature Selection. *Informatica* 45:517–528.
- JANKO, V. & . 2021. Three Methods for Energy-Efficient Context Recognition. *Informatica* 45:315–316.

- JIAN, Y. & , X. DONG, L. JIAN. 2021. Detection and Recognition of Abnormal Data Caused by Network Intrusion Using Deep Learning. *Informatica* 45:441–445.
- KAKAR, S. & , D. DHAKA, M. MEHROTRA. 2021. Value-Based Retweet Prediction on Twitter. *Informatica* 45:267–276.
- KALABAKOV, S. & . 2021. Analysis of Deep Transfer Learning Using DeepConvLSTM for Human Activity Recognition from Wearable Sensors. *Informatica* 45:289–296.
- KŐRÖSI, G. & , T. VINKÓ. 2021. A Practical Framework for Real Life Webshop Sales Promotion Targeting. *Informatica* 45:625–632.
- KOURBANE, I. & . 2021. Skeleton-aware Multi-scale Heatmap Regression for 2D Hand Pose Estimation. *Informatica* 45:593–604.
- KULKARNI, S.A. & . 2021. Impact of Gaussian Noise for Optimized Support Vector Machine Algorithm Applied to Medicare Payment on Raspberry Pi. *Informatica* 45:643–652.
- LI, J. & , Y. WANG, J. WANG. 2021. An Analysis of Emotional Tendency Under the Network Public Opinion: Deep Learning. *Informatica* 45:149–156.
- LIU, J. & . 2021. Research on Campus Network Equipment Environment Monitoring Based on Internet of Things. *Informatica* 45:303–307.
- MELLAL, N. & , T. GUERRAM, F. BOUHALASSA. 2021. An Approach for Automatic Ontology Enrichment from Texts. *Informatica* 45:81–91.
- MIJOSKA, M. & , B. RISTEVSKI. 2021. Possibilities for Applying Blockchain Technology – A Survey. *Informatica* 45:319–333.
- MISHRA, S. & , D. MISHRA, P.K. MALLICK, G.H. SANTRA, S. KUMAR. 2021. A Novel Borda Count Based Feature Ranking and Feature Fusion Strategy to Attain Effective Climatic Features for Rice Yield Prediction. *Informatica* 45:13–31.
- MISHRA, S. & , D. MISHRA, P.K. MALLICK, G. SANTRA, S. KUMAR. 2021. A Classifier Ensemble Approach for Prediction of Rice Yield based on Climatic Variability for Coastal Odisha Region of India. *Informatica* 45:367–380.
- NAJJAR, A. & , B. AMRO, M. MACEDO. 2021. An Intelligent Decision Support System For Recruitment: Resumes Screening And Applicants Ranking. *Informatica* 45:617–623.
- NASIRI, S. & , F. SADOUGHI, A. DEHNAD, M.H. TADAYON, H. AHMADI. 2021. Layered Architecture for Internet of Things-based Healthcare System: A Systematic Literature Review. *Informatica* 45:543–562.
- NGO LE HUY, H. & , H. HO MINH, T. NGUYEN VAN, H. NGUYEN VAN. 2021. Keyphrase Extraction Model: A New Design and Application on Tourism Information. *Informatica* 45:563–569.
- PANDA, D. & , D. PANDA, S.R. DASH, S. PARIDA. 2021. Extreme Learning Machines with Feature Selection Using GA for Effective Prediction of Fetal Heart Disease: A Novel Approach. *Informatica* 45:381–392.
- PATEL, R. & , S. TANWANI, C. PATIDAR. 2021. Relation Extraction Between Medical Entities Using Deep Learning Approach. *Informatica* 45:359–366.
- PENG, X. & . 2021. Research on Emotion Recognition Based on Deep Learning for Mental Health. *Informatica* 45:127–132.
- PETROVSKA, B. & , T.A. PACEMSKA, N. STOJKOVIK, A. STOJANOVA, M. KOCALEVA. 2021. Machine Learning with Remote Sensing Image Data Sets. *Informatica* 45:347–358.
- PINTÉR, Á. & , S. SZÉNÁSI. 2021. Index Dependent Nested Loops Parallelization with an Even Distributed Number of Steps. *Informatica* 45:493–505.
- PODGORELEC, D. & , A. NERAT, B. ŽALIK. 2021. Statistics-Based Chain Code Compression with Decreased Sensitivity to Shape Artefacts. *Informatica* 45:205–212.
- RATHEE, A. & , J.K. CHHABRA. 2021. Extraction and Evaluation of Software Components from Object-Oriented Artifacts. *Informatica* 45:171–172.
- SAMANT, S.S. & , N.B. MURTHY, A. MALAPATI. 2021. Categorization of Event Clusters from Twitter Using Term Weighting Schemes. *Informatica* 45:405–414.
- SEIKH, M.R. & , U. MANDAL. 2021. Some Picture Fuzzy Aggregation Operators Based on Frank t-norm and t-conorm: Application to MADM Process. *Informatica* 45:447–461.
- SIVASLIOGLU, S. & , F.O. CATEK, K. ŞAHINBAŞ. 2021. A Generative Model Based Adversarial Security of Deep Learning and Linear Classifier Models. *Informatica* 45:33–64.
- ŠOBERL, D. & . 2021. Automated Planning with Induced Qualitative Models in Dynamic Robotic Domains. *Informatica* 45:489–490.
- SORNKLIANG, W. & , T. PHETKAEW. 2021. Performance Analysis of Test Path Generation Techniques Based on Complex Activity Diagrams. *Informatica* 45:231–242.
- SUBRAMANYARAO, S. & . 2021. Comments About a Paper Titled A Model and Framework for Online Security Benchmarking. *Informatica* 45:173–173.
- SUN, J. & . 2021. Prediction and Estimation of Book Borrowing

in the Library: Data Mining. Informatica 45:163–168.

TLILI, A. & , S. CHIKHI. 2021. Risks Analyzing and Management in Software Project Management Using Fuzzy Cognitive Maps with Reinforcement Learning. Informatica 45:133–141.

VIGNEAU, E. & . 2021. Clustering of Variables for Enhanced Interpretability of Predictive Models. Informatica 45:507–516.

VLAHEK, D. & , T. STOŠIĆ, T. GOLOB, M. KALC, T. LIČEN, M. VOGRIN, D. MONGUS. 2021. Method for Estimating Tensiomyography Parameters from Motion Capture Data. Informatica 45:213–222.

WAKEEL, S. & , S. BINGOL, Z. DING, S. AHMAD, M. BASHIR, M.S. MOHAMMAD MOHSEN EMAMAT, H. FAYAZ. 2021. A New Hybrid LGPMBWM-PIV Method for Automotive Material Selection. Informatica 45:105–115.

YAYLA, R. & , T.T. BILGIN. 2021. Determining of the User Attitudes on Mobile Security Programs with Machine Learning Methods. Informatica 45:393–403.

ZAVALNIJ, B. & , S. SZABO. 2021. Estimating Clique Size via Discarding Subgraphs. Informatica 45:197–204.

ZHANG, G. & . 2021. Research on the Efficiency of Intelligent Algorithm for English Speech Recognition and Sentence Translation. Informatica 45:309–314.

ZOMBORI, D. & . 2021. ParallelGlobal with Low Thread Interactions. Informatica 45:191–196.

## Editorials

BRODNIK, A. & , G. GALAMBOS. 2021. Introduction to "Middle-European Conference on Applied Theoretical Computer Science (MATCOS-19)". Informatica 45:177–177.

## JOŽEF STEFAN INSTITUTE

*Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.*

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of Slovenia (or S<sup>lo</sup>venia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park “Ljubljana” has been proposed as part of the national strategy for technological development to foster synergies between research and

industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park “Ljubljana”. The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana, Slovenia  
Tel.:+386 1 4773 900, Fax.:+386 1 251 93 85  
WWW: <http://www.ijs.si>  
E-mail: [matjaz.gams@ijs.si](mailto:matjaz.gams@ijs.si)  
Public relations: Polona Strnad



# INFORMATICA

## AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

### INVITATION, COOPERATION

#### Submissions and Refereeing

Please register as an author and submit a manuscript at: <http://www.informatica.si>. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica L<sup>A</sup>T<sub>E</sub>X format and figures in .eps format. Style and examples of papers can be obtained from <http://www.informatica.si>. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

#### SUBSCRIPTION

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: [drago.torkar@ijs.si](mailto:drago.torkar@ijs.si)

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than twentyseven years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica web edition is free of charge and accessible at <http://www.informatica.si>.

Informatica print edition is free of charge for major scientific, educational and governmental institutions. Others should subscribe.



## Informatica WWW:

<http://www.informatica.si/>

### Referees from 2008 on:

A. Abraham, S. Abraham, R. Accornero, A. Adhikari, R. Ahmad, G. Alvarez, N. Anciaux, R. Arora, I. Awan, J. Azimi, C. Badica, Z. Balogh, S. Banerjee, G. Barbier, A. Baruzzo, B. Batagelj, T. Beaubouef, N. Beaulieu, M. ter Beek, P. Bellavista, K. Bilal, S. Bishop, J. Bodlaj, M. Bohanec, D. Bolme, Z. Bonikowski, B. Bošković, M. Botta, P. Brazdil, J. Brest, J. Brichau, A. Brodnik, D. Brown, I. Bruha, M. Bruynooghe, W. Buntine, D.D. Burdescu, J. Buys, X. Cai, Y. Cai, J.C. Cano, T. Cao, J.-V. Capella-Hernández, N. Carver, M. Cavazza, R. Ceylan, A. Chebotko, I. Chekalov, J. Chen, L.-M. Cheng, G. Chiola, Y.-C. Chiou, I. Chorbev, S.R. Choudhary, S.S.M. Chow, K.R. Chowdhury, V. Christlein, W. Chu, L. Chung, M. Cigliarić, J.-N. Colin, V. Cortellessa, J. Cui, P. Cui, Z. Cui, D. Cutting, A. Cuzzocrea, V. Cvjetkovic, J. Cyprianski, L. Čehovin, D. Čerepnalkoski, I. Čosić, G. Daniele, G. Danoy, M. Dash, S. Datt, A. Datta, M.-Y. Day, F. Debili, C.J. Debono, J. Dedič, P. Degano, A. Dekdouk, H. Demirel, B. Demoen, S. Dendamrongvit, T. Deng, A. Derezsinska, J. Dezert, G. Dias, I. Dimitrovski, S. Dobrišek, Q. Dou, J. Doumen, E. Dovgan, B. Dragovich, D. Dragic, O. Drbohlav, M. Drole, J. Dujmović, O. Ebers, J. Eder, S. Elaluf-Calderwood, E. Engström, U. riza Erturk, A. Farago, C. Fei, L. Feng, Y.X. Feng, B. Filipič, I. Fister, I. Fister Jr., D. Fišer, A. Flores, V.A. Fomichov, S. Forli, A. Freitas, J. Fridrich, S. Friedman, C. Fu, X. Fu, T. Fujimoto, G. Fung, S. Gabrielli, D. Galindo, A. Gambarara, M. Gams, M. Ganzha, J. Garbajosa, R. Gennari, G. Georgeson, N. Gligorić, S. Goel, G.H. Gonnet, D.S. Goodsell, S. Gordillo, J. Gore, M. Grčar, M. Grgurović, D. Grosse, Z.-H. Guan, D. Gubiani, M. Guid, C. Guo, B. Gupta, M. Gusev, M. Hahsler, Z. Haiping, A. Hameed, C. Hamzaçebi, Q.-L. Han, H. Hanping, T. Härder, J.N. Hatzopoulos, S. Hazelhurst, K. Hempstalk, J.M.G. Hidalgo, J. Hodgson, M. Holbl, M.P. Hong, G. Howells, M. Hu, J. Hyvärinen, D. Ienco, B. Ionescu, R. Irfan, N. Jaisankar, D. Jakobović, K. Jassem, I. Jawhar, Y. Jia, T. Jin, I. Jureta, Đ. Juričić, S. K, S. Kalajdziski, Y. Kalantidis, B. Kaluža, D. Kanellopoulos, R. Kapoor, D. Karapetyan, A. Kassler, D.S. Katz, A. Kaveh, S.U. Khan, M. Khattak, V. Khomenko, E.S. Khorasani, I. Kitanovski, D. Kocev, J. Kocijan, J. Kollár, A. Kontostathis, P. Korošec, A. Koschmider, D. Košir, J. Kovač, A. Krajnc, M. Krevs, J. Krogstie, P. Krsek, M. Kubat, M. Kukar, A. Kulis, A.P.S. Kumar, H. Kwašnicka, W.K. Lai, C.-S. Lai, K.-Y. Lam, N. Landwehr, J. Lanir, A. Lavrov, M. Layouni, G. Leban, A. Lee, Y.-C. Lee, U. Legat, A. Leonardis, G. Li, G.-Z. Li, J. Li, X. Li, X. Li, Y. Li, Y. Li, S. Lian, L. Liao, C. Lim, J.-C. Lin, H. Liu, J. Liu, P. Liu, X. Liu, X. Liu, F. Logist, S. Loskovska, H. Lu, Z. Lu, X. Luo, M. Luštrek, I.V. Lyustig, S.A. Madani, M. Mahoney, S.U.R. Malik, Y. Marinakis, D. Marinčič, J. Marques-Silva, A. Martin, D. Marwede, M. Matijašević, T. Matsui, L. McMillan, A. McPherson, A. McPherson, Z. Meng, M.C. Mihaescu, V. Milea, N. Min-Allah, E. Minisci, V. Mišić, A.-H. Mogos, P. Mohapatra, D.D. Monica, A. Montanari, A. Moroni, J. Mosegaard, M. Moškon, L. de M. Mourelle, H. Moustafa, M. Možina, M. Mrak, Y. Mu, J. Mula, D. Nagamalai, M. Di Natale, A. Navarra, P. Navrat, N. Nedjah, R. Nejabat, W. Ng, Z. Ni, E.S. Nielsen, O. Nouali, F. Novak, B. Novikov, P. Nurmi, D. Obrul, B. Oliboni, X. Pan, M. Pančur, W. Pang, G. Papa, M. Paprzycki, M. Paralič, B.-K. Park, P. Patel, T.B. Pedersen, Z. Peng, R.G. Pensa, J. Perš, D. Petcu, B. Petelin, M. Petkovšek, D. Pevec, M. Pičulin, R. Piltaver, E. Pirogova, V. Podpečan, M. Polo, V. Pomponiu, E. Popescu, D. Poshyvanik, B. Potočnik, R.J. Povinelli, S.R.M. Prasanna, K. Pripužič, G. Puppis, H. Qian, Y. Qian, L. Qiao, C. Qin, J. Que, J.-J. Quisquater, C. Rafe, S. Rahimi, V. Rajković, D. Raković, J. Ramaekers, J. Ramon, R. Ravnik, Y. Reddy, W. Reimche, H. Rezankova, D. Rispoli, B. Ristevski, B. Robič, J.A. Rodriguez-Aguilar, P. Rohatgi, W. Rossak, I. Rožanc, J. Rupnik, S.B. Sadek, K. Saeed, M. Saeki, K.S.M. Sahari, C. Sakharwade, E. Sakkopoulos, P. Sala, M.H. Samadzadeh, J.S. Sandhu, P. Scaglioso, V. Schau, W. Schempp, J. Seberry, A. Senanayake, M. Senobari, T.C. Seong, S. Shamala, c. shi, Z. Shi, L. Shiguo, N. Shilov, Z.-E.H. Slimane, F. Smith, H. Sneed, P. Sokolowski, T. Song, A. Soppera, A. Sornioti, M. Stajdohar, L. Stanescu, D. Strnad, X. Sun, L. Šajn, R. Šenkeřík, M.R. Šikonja, J. Šilc, I. Škrjanc, T. Štajner, B. Šter, V. Štruc, H. Takizawa, C. Talcott, N. Tomasev, D. Torkar, S. Torrente, M. Trampuš, C. Tranoris, K. Trojancanec, M. Tschienschke, F. De Turck, J. Twycross, N. Tziritas, W. Vanhoof, P. Vateekul, L.A. Vese, A. Visconti, B. Vlaovič, V. Vojisavljević, M. Vozalis, P. Vračar, V. Vranić, C.-H. Wang, H. Wang, H. Wang, H. Wang, S. Wang, X.-F. Wang, X. Wang, Y. Wang, A. Wasilewska, S. Wenzel, V. Wickramasinghe, J. Wong, S. Wrobel, K. Wrona, B. Wu, L. Xiang, Y. Xiang, D. Xiao, F. Xie, L. Xie, Z. Xing, H. Yang, X. Yang, N.Y. Yen, C. Yong-Sheng, J.J. You, G. Yu, X. Zabulis, A. Zainal, A. Zamuda, M. Zand, Z. Zhang, Z. Zhao, D. Zheng, J. Zheng, X. Zheng, Z.-H. Zhou, F. Zhuang, A. Zimmermann, M.J. Zuo, B. Zupan, M. Zuqiang, B. Žalik, J. Žižka,

# *Informatica*

## An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: <http://www.informatica.si>.

**Subscription Information** Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.

The subscription rate for 2021 (Volume 45) is

- 60 EUR for institutions,
- 30 EUR for individuals, and
- 15 EUR for students

Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut, Peter and Jaša Žnidar; [borut.znidar@gmail.com](mailto:borut.znidar@gmail.com).

Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email ([drago.torkar@ijs.si](mailto:drago.torkar@ijs.si)), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):

Slovene Society for Pattern Recognition (Vitimir Štruc)

Slovenian Artificial Intelligence Society (Sašo Džeroski)

Cognitive Science Society (Olga Markič)

Slovenian Society of Mathematicians, Physicists and Astronomers (Dragan Mihailović)

Automatic Control Society of Slovenia (Giovanni Godena)

Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Mark Pleško)

ACM Slovenia (Nikolaj Zimic)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math
---

# *Informatica*

**An International Journal of Computing and Informatics**

Index Dependent Nested Loops Parallelization with an Even Distributed Number of Steps	Á. Pintér, S. Szénási	<b>493</b>
Clustering of Variables for Enhanced Interpretability of Predictive Models	E. Vigneau	<b>507</b>
Performance of Malware Detection Classifier Using Genetic Programming in Feature Selection	H. Harahsheh, M. Shraideh, S. Sharaeh	<b>517</b>
Modeling and Performance Analysis of Resource Provisioning in Cloud Computing using Probabilistic Model Checking	H. Debbi	<b>529</b>
Layered Architecture for Internet of Things-based Healthcare System: A Systematic Literature Review	S. Nasiri, F. Sadoughi, A. Dehnad, M.H. Tadayon, H. Ahmadi	<b>543</b>
Keyphrase Extraction Model: A New Design and Application on Tourism Information	H. Ngo Le Huy, H. Ho Minh, T. Nguyen Van, H. Nguyen Van	<b>563</b>
Reduced Number of Parameters for Predicting Post-Stroke Activities of Daily Living Using Machine Learning Algorithms on Initiating Rehabilitation	A.M. Alqudah, M. Al-Hashem, A. Alqudah	<b>571</b>
Formal Verification of Pipelined Cryptographic Circuits: A Functional Approach	A. Bitat, S. Merniz	<b>583</b>
Skeleton-aware Multi-scale Heatmap Regression for 2D Hand Pose Estimation	I. Kourbane	<b>593</b>
A Blockchain and NLP Based Electronic Health Record System: Indian Subcontinent Context	P.K. Bharimalla, H. Choudhury, S. Parida, D.K. Mallick, S.R. Dash	<b>605</b>
An Intelligent Decision Support System For Recruitment: Resumes Screening and Applicants Ranking	A. Najjar, B. Amro, M. Macedo	<b>617</b>
A Practical Framework for Real Life Webshop Sales Promotion Targeting	G. Kőrösi, T. Vinkó	<b>625</b>
Evaluation of Multimedia User Interface Design Method for M-learning (MobLearn): A Comparative Study	M. Aldukhail	<b>633</b>
Impact of Gaussian Noise for Optimized Support Vector Machine Algorithm Applied to Medicare Payment on Raspberry Pi	S.A. Kulkarni	<b>643</b>
Textual Entailment for Modern Standard Arabic	M. Alabbas	<b>653</b>

