02

UPORABNA

INFORMATIKA

2020 ‹ ŠTEVILKA 2 ‹ APR. MAJ. JUN. ‹ LETNIK XXVIII ‹ ISSN 1318-1882

U P O R A B N A
I N F O R M A T I K A
2020 ŠTEVILKA 2 APR/MAJ/JUN LETNIK XXVIII ISSN 1318-1882

Znanstveni prispevki

Jaka Stavanja, Matej Klemen, Lovro Šubelj:
Napovedovanje umorov v seriji Igra prestolov z uporabo analize omrežij 55

Anej Svete, Jakob Hostnik, Lovro Šubej:
Ne gre le za melodijo: kako Evropa glasuje za svoje najljubše skladbe 66

Simon Kolmanič, Maršenka Markset, Domen Mongus, Borut Žalik:
Tehnologije navidezne in obogatene resničnosti kot orodje za predstavitev novih idej in produktov na sejmih: primer Mahepa 85

Strokovni prispevki

Alenka Brezavšček, Maja Minič:
Izboljšanje ozaveščenosti na področju informacijske varnosti z uporabo metod igrifikacije 94

Pogled v zgodovino

Saša Divjak:
Spomini na prve korake v robotiko v Sloveniji 102

U P O R A B N A
I N F O R M A T I K A

2020 ŠTEVILKA 2 APR/MAJ/JUN LETNIK XXVIII ISSN 1318-1882

Ustanovitelj in izdajatelj
Slovensko društvo INFORMATIKA
Litostrojska cesta 54, 1000 Ljubljana

Predstavnik
Niko Schlamberger

Odgovorni urednik
Saša Divjak

Tehnični urednik
Slavko Žitnik

Lektoriranje angleških izvlečkov
Marvelingua (angl.)

Oblikovanje
KOFEIN DIZAJN, d. o. o.

Prelom in tisk
Boex DTP, d. o. o., Ljubljana

Naklada
200 izvodov

Naslov uredništva
Slovensko društvo INFORMATIKA
Uredništvo revije Uporabna informatika
Litostrojska cesta 54, 1000 Ljubljana
www.uporabna-informatika.si

Revija izhaja četrtletno. Cena posamezne številke je 20,00 EUR. Letna naročnina za podjetja 85,00 EUR, za vsak nadaljnji izvod 60,00 EUR, za posameznike 35,00 EUR, za študente in seniorje 15,00 EUR. V ceno je vključen DDV.

Revija Uporabna informatika je od številke 4/VII vključena v mednarodno bazo INSPEC.

Revija Uporabna informatika je pod zaporedno številko 666 vpisana v razvid medijev, ki ga vodi Ministrstvo za kulturo RS.

Revija Uporabna informatika je vključena v Digitalno knjižnico Slovenije (dLib.si).

Vabilo avtorjem

V reviji Uporabna informatika objavljamo kakovostne izvirne članke domačih in tujih avtorjev z najširšega področja informatike v poslovanju podjetij, javni upravi in zasebnem življenju na znanstveni, strokovni in informativni ravni; še posebno spodbujamo objavo interdisciplinarnih člankov. Zato vabimo avtorje, da prispevke, ki ustrezajo omenjenim usmeritvam, pošljejo uredništvu revije po elektronski pošti na naslov ui@drustvo-informatika.si.

Avtorje prosimo, da pri pripravi prispevka upoštevajo navodila, objavljena v nadaljevanju ter na naslovu http://www.uporabna-informatika.si.

Za kakovost prispevkov skrbi mednarodni uredniški odbor. Članki so anonimno recenzirani, o objavi pa na podlagi recenzij samostojno odloča uredniški odbor. Recenzenti lahko zahtevajo, da avtorji besedilo spremenijo v skladu s priporočili in da popravljeni članek ponovno prejmejo v pregled. Uredništvo pa lahko še pred recenzijo zavrne objavo prispevka, če njegova vsebina ne ustreza vsebinski usmeritvi revije ali če članek ne ustreza kriterijem za objavo v reviji.

Pred objavo članka mora avtor podpisati izjavo o avtorstvu, s katero potrjuje originalnost članka in dovoljuje prenos materialnih avtorskih pravic. Nenaročenih prispevkov ne vračamo in ne honoriramo. Avtorji prejmejo enoletno naročnino na revijo Uporabna informatika, ki vključuje avtorski izvod revije in še nadaljnje tri zaporedne številke.

S svojim prispevkom v reviji Uporabna informatika boste prispevali k širjenju znanja na področju informatike. Želimo si čim več prispevkov z raznoliko in zanimivo tematiko in se jih že vnaprej veselimo.

Uredništvo revije

Navodila avtorjem člankov

Članke objavljamo praviloma v slovenščini, članke tujih avtorjev pa v angleščini. Besedilo naj bo jezikovno skrbno pripravljeno. Priporočamo zmernost pri uporabi tujk in – kjer je mogoče – njihovo zamenjavo s slovenskimi izrazi. V pomoč pri iskanju slovenskih ustreznic priporočamo uporabo spletnega terminološkega slovarja Slovenskega društva Informatika Islovar (www.islovar.org).

Znanstveni članek naj obsega največ 40.000 znakov, strokovni članki do 30.000 znakov, obvestila in poročila pa do 8.000 znakov.

Članek naj bo praviloma predložen v urejevalniku besedil Word (*.doc ali *.docx) v enojnem razmaku, brez posebnih znakov ali poudarjenih črk. Za ločilom na koncu stavka napravite samo en prazen prostor, pri odstavkih ne uporabljajte zamika.

Naslovu članka naj sledi za vsakega avtorja polno ime, ustanova, v kateri je zaposlen, naslov in elektronski naslov. Sledi naj povzetek v slovenščini v obsegu 8 do 10 vrstic in seznam od 5 do 8 ključnih besed, ki najbolje opredeljujejo vsebinski okvir članka. Pred povzetkom v angleščini naj bo še angleški prevod naslova, prav tako pa naj bodo dodane ključne besede v angleščini. Obratno velja v primeru predložitve članka v angleščini.

Razdelki naj bodo naslovljeni in oštevilčeni z arabskimi številkami.

Slike in tabele vključite v besedilo. Opremite jih z naslovom in oštevilčite z arabskimi številkami. Vsako sliko in tabelo razložite tudi v besedilu članka. Če v članku uporabljate slike ali tabele drugih avtorjev, navedite vir pod sliko oz. tabelo. Revijo tiskamo v črno-beli tehniki, zato barvne slike ali fotografije kot original niso primerne. Slik zaslonov ne objavljamo, razen če so nujno potrebne za razumevanje besedila. Slike, grafikoni, organizacijske sheme ipd. naj imajo belo podlago. Enačbe oštevilčite v oklepajih desno od enačbe.

V besedilu se sklicujte na navedeno literaturo skladno s pravili sistema APA navajanja bibliografskih referenc, najpogosteje torej v obliki (Novak & Kovač, 2008, str. 235). Na koncu članka navedite samo v članku uporabljeno literaturo in vire v enotnem seznamu po abecednem redu avtorjev, prav tako v skladu s pravili APA. Več o sistemu APA, katerega uporabo omogoča tudi urejevalnik besedil Word 2007, najdete na strani http://owl.english.purdue.edu/owl/resource/560/01/.

Članku dodajte kratek življenjepis vsakega avtorja v obsegu do 8 vrstic, v katerem poudarite predvsem strokovne dosežke.

Napovedovanje umorov v seriji Igra prestolov z uporabo analize omrežij

Jaka Stavanja, Matej Klemen, Lovro Šubelj
University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana
js8927@student.uni-lj.si, mk3141@student.uni-lj.si, lovro.subelj@fri.uni-lj.si

Izvleček
Serija Igra prestolov ima veliko oboževalcev prav zaradi dramatičnih smrtih pomembnih likov. V našem delu želimo na podlagi podatkov o preteklih umorih in ubojih v seriji ter dodatnih metapodatkih o likih napovedati kdo ubije koga. Iz podatkov o umorih v seriji zgradimo omrežje, kjer vozlišča predstavljajo like v seriji, usmerjene povezave pa predstavljajo umore. Na zgrajenem omrežju preizkusimo različne metode za napovedovanje povezav in preverimo njihovo uspešnost. Poleg tega iz družbenega omrežja likov pridobimo dodatne značilke in jih uporabimo za napovedovanje umorov s tehnikami podatkovnega rudarjenja. Ugotovimo, da zaradi majhne velikosti podatkovne množice in nestrukturiranosti umorov, z osnovnimi metodami napovedovanja povezav umorov ne moremo napovedati. Dodajanje novih značilk in uporaba tehnik podatkovnega rudarjenja izboljša dosežene rezultate. Pokažemo, da obstaja način, ki na zgrajenem omrežju umorov doseže najboljše rezultate, vendar ni praktično uporaben. Ta način doseže najboljšo površino pod ROC krivuljo, t.j. 0.875.
Ključne besede: ekstrakcija značilk, Igra prestolov, napovedovanje povezav, analiza omrežij.

Abstract
TV series such as HBO's Game of Thrones have a high number of dedicated followers, mostly due to the dramatic murders of the most important characters. In our work, we try to predict killer and victim pairs using data on previous kills as well as additional metadata. We construct a network where two character nodes are linked if one killed the other, then use a link prediction framework to evaluate different techniques for kill predictions. Lastly, we compute various network properties on a social network of characters and use them as features in conjunction with classic data mining techniques. Due to the small size of the dataset and the somewhat random kill distribution, we cannot make accurate predictions with standard indices alone, although using them in conjunction with additional rules based on degrees has yielded results that are more reliable. The features we compute on the social network help the classic machine learning approaches; however, they do not yield very accurate predictions. The best results overall are achieved using indices that use simple degree information, the best of which result in the Area Under the ROC Curve of 0.875.
Keywords: Feature extraction, Game of Thrones, link prediction, network analysis.

1 INTRODUCTION
With the ever increasing popularity of the TV series Game of Thrones, coming mostly from it's incredible plot twists and deaths of main characters, the question arises whether we can predict those deaths from a network analysis point of view. If we are able to predict the deaths from the data, collected from previous episodes, that means that the author is very predictable, which might not be the best thing in terms of the show being entertaining. To predict the deaths, we construct a network illustrated in Figure 1, where nodes are characters in the show (along with other entities that are able to kill another character, such as a horse or a dragon) and connect two if one has murdered the other. We then try to predict whether a certain link between two nodes in the network happened (removing the link from the network beforehand). Because we are dealing with temporal data,

we also remove links that appear in the network after the link currently being predicted. We use different approaches to assign scores to the pairs, for example different link prediction indices such as the preferential attachment index (Liben-Nowell & Kleinberg, 2007) or the Adamic-Adar index

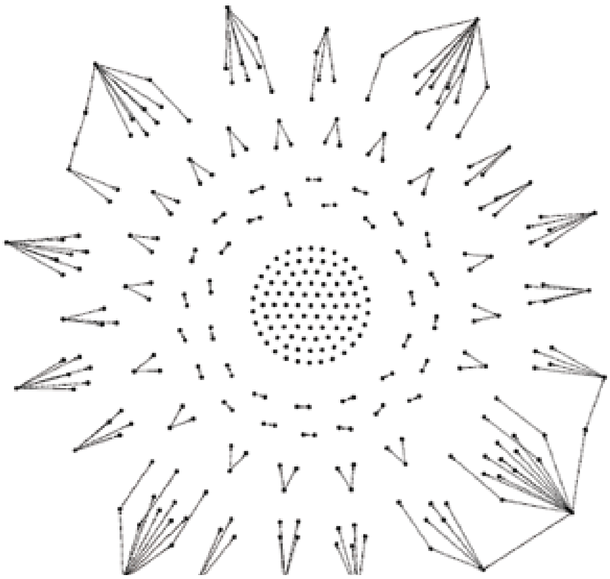


Figure 1: **Game of Thrones kills network.**

(Adamic & Adar, 2003). We also construct additional features based on various network properties and additional metadata and see how this influences the results. The network properties for the features used in machine learning approaches are then computed on a different social network of the Game of Thrones characters, where nodes representing characters are connected if the characters appear somewhat close in the original books' story.

The rest of the paper is structured as follows. In Section 2 we provide an overview of existing literature on our studied topic. In Section 3 we describe the methods that we use in our work. In Section 4 we provide the results of our work, which we then discuss in Section 5. In Section 6 we summarize the work that was done and provide some possible future improvements.

2 RELATED WORK

We use link prediction as a way to infer new links between nodes in a graph using different network properties. The field of link prediction research started evolving around trying to either generate synthetic networks or infer missing data for network-like data structures. Later on, that evolved into more of a recommendation type of approach (for example, trying to recommend friendships on social networks), based on the same principles as we use to calculate probabilities of new links. One of the most important articles on the network properties that we can use to infer new links is written by Barabási & Albert and features exploring the phenomenon of preferential attachment (Barabási & Albert, 1999). The work on this property and other founding principles for link prediction is then briefly described inside the article by the same authors (Albert & Barabási, 2002). The next explored thing is missing data (Kossinets, 2006) along with further studies on missing links and spurious networks (Guimerà & Sales-Pardo, 2009). An extensive survey on link prediction is written by Lü & Zhou (Lü & Zhou, 2011) where the whole field along with all the better-known indices to date is presented and the authors show how the classic problems in link prediction didn't use enough network properties or community structure, which was explored by Girvan & Newman (Girvan & Newman, 2002). We use some of their findings in our research. One of the most well known indices to calculate the likelihood of new links appearing in a network is the common neighbors index (Newman, 2001) (Kossinets, 2006), implying that the more common neighbors nodes have, the more likely they are to form a link. Some other popular choices include the cosine distance index (also named the Salton index) (McGill & Salton, 1983), the Jaccard index (Jaccard, 1901), the preferential attachment index (Barabási & Albert, 1999) and the Adamic-Adar index (Adamic & Adar, 2003). For our experiments, we pick some of the most used indices.

For Game of Thrones specifically, some research has already been done in terms of finding which aspects of the show resonate with the viewer count the most and how real the characters' interactions are done by modeling the show's houses with a network and exploring structural balance (Liu & Albergante, 2017). Further studies determined who has the best strategic position in the show's world (Beveridge & Shan, 2016), but the only article touching on death prediction studies (Angraal et al., 2018) for the show used Cox's proportional hazard model (Cox, 1972), which didn't explore any network structure properties, but rather used a regression approach to determine which factors introduce a higher risk of mortality through time. But the killers were not taken into consideration here, we just know how likely a character is to die, thus we cannot really compare this approach with ours. This gives us a unique opportunity to try and use network properties along with metadata to better predict kills in the mythical world of the series, but instead of only predicting how likely someone is to die, we can also predict both the killers and victims, on which there has not yet been much research.

3 METHODS AND NETWORK PROPERTIES

In this section, we describe the framework used to test our link prediction methods. We also provide descriptions of some classic link prediction methods based on well-known properties of networks. Then, we introduce a new social network, described in Subsection 3.2.5, which connects characters if they appear close in the original story from the books. We compute various network properties from that network and use an automatic node embedding technique to compute features for use in traditional machine learning approach for classification.

3.1 Link prediction using the kills network

The network of kills is constructed of directed links between pairs of nodes i and j, where node i represents a character that killed the character represented by node j. The network is very small and also very sparse. It has 353 nodes and 194 links. By looking at the visualization of the network in Figure 1, we can see that most kills appear outside somewhat bigger connected components and do not seem to be attributed to hubs (i.e. high degree nodes). We can still observe a few hubs and connected components around them, where a lot of kills seem to occur, but in the majority of cases there are just two nodes involved in the kill, which on their own form a small two-node connected component. That gives us a clue as to which methods might predict kills better than others. We can construct an index that predicts links based on nodes' out degrees. Since the in-degrees in the networks could be either zero or one (meaning alive or dead) we cannot get any additional information from that besides whether a person can still kill someone or not (when they are already dead). The out-degree of a node can potentially be of use, since people that kill a lot of people might also tend to kill more people, and those who never killed anyone might not be inclined to murder or kill. A plot of the out-degree distribution (using a logarithmic scale for the fractions of nodes) is shown in Figure 2.

In this subsection, we propose an evaluation framework and different indices that we use for the link prediction.

3.1.1 Evaluation framework for link prediction

To test how well our link prediction techniques work on our network, we construct a framework and provide a brief description of it here. It takes our prediction index function and the network as the input and outputs the Area Under the ROC Curve (AUC) value. The prediction index function assigns a score sij to every link between two nodes i and j that is being tested. A high score implies a high likelihood that the link exists in the network and a low score implies a small likelihood of the link's existence.

The core of the testing framework is the logic, which removes links from the network and then tries to predict how likely the links we have removed are to form as the network evolves through time. To use as much information as we can, we choose an episode and remove links from that episode (e.g. episode 30) and onwards from the graph. Then, we predict the links (i.e. kills) at that time using the information about kills from the previous episodes. We can then clone the original graph, remove links from the next episode in the chronology (i.e. episode 31) and predict links for that episode using the information from all the episodes before (including information from episode 30 in this example). By predicting links in this way, we are not using the data from the future to predict past links and we are using a lot more information than if we were to remove links from a certain episode onwards and just try to predict links for multiple episodes in one single iteration.

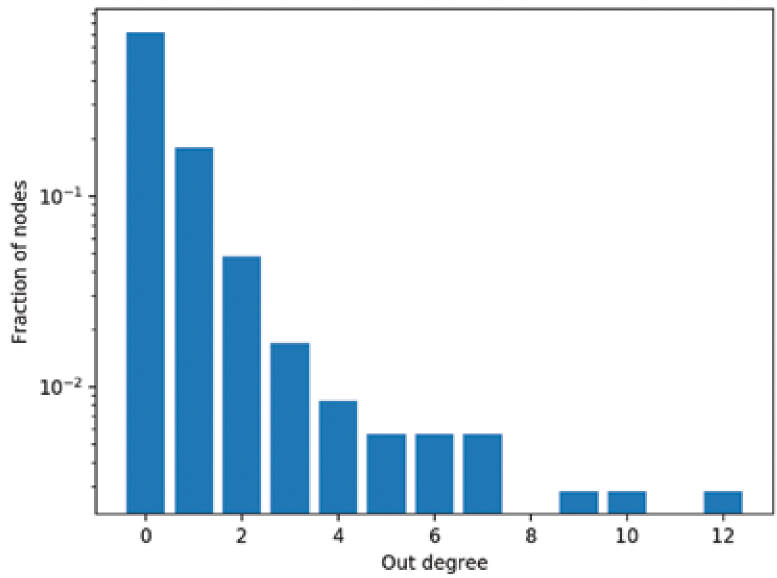


Figure 2: **Out-degree distribution for the kills network.**

The idea of the framework's core implementation is as follows:

1. Iterate through every episode in the range from 30 to 60, and at each step do:
   (a) $L_P \leftarrow$ remove node links $(i, j) \in L$ after the current episode.
   (b) Compute sij for $(i, j) \in L_P$ at the current episode time.
   (c) Save the scores to S_P.
2. $L_N \leftarrow$ randomly sample $|L_P|$ unlinked nodes $(i, j) \notin L$.
3. Compute sij for $(i, j) \in L_N$ and save scores to S_N.
4. Compute AUC by comparing the scores assigned to positive samples S_P and scores assigned to negative samples S_N. When comparing scores, we assign to b the amount of times when the score from S_P is bigger than the one from S_N, and assign to e the number of times when the two scores are equal. This is counted across a random sample of $|L_P|$ pairs. Then, we compute the AUC as $\frac{b + e/2}{|L_P|}$.

3.1.2 Alive index

We create a type of a baseline index by looking at the network's high level properties. We check if the killer has an in-degree of zero and the target has an in-degree of zero (i.e. killer is alive and target has not been killed yet). If the endpoints of a link satisfy these conditions, the value of the index is 1, otherwise it is 0.

3.1.3 Preferential attachment index

Real world networks tend to have a scale-free degree distribution due to a phenomenon known as preferential attachment (Barabási & Albert, 1999). The preferential attachment index (Liben-Nowell & Kleinberg, 2007) is defined as $s_{ij} = k_i k_j$, where k_i is the degree of node i. For our problem, we use out-degrees only, as only they provide useful information. The idea behind the index is in the preferential attachment process — nodes are more likely to connect with nodes that have a high degree, thus a link between two nodes with high degrees should be assigned a high score s_{ij}. We modify this definition slightly and define a modified preferential attachment index as $s_{ij} = k_i^{(out)} k_j^{(out)}$, where $k_i^{(out)}$ is the out-degree of node i. The logic behind that is that the more kills one has to their record, the more they are inclined to murder and vice-versa. Additionally, we include the in-degree information in a modified version of the preferential attachment index: if the source node or the target node has an in-degree larger than zero (either the killer or the victim is already dead), a very negative score ($-\infty$) is returned. Otherwise, the regular version of the index gets computed.

3.1.4 Adamic-Adar similarity index

In real world networks, links tend to appear between nodes that have a lot of common neighbors (Watts & Strogatz, 1998). But due to preferential attachment, nodes tend to connect to higher degree nodes more likely, thus making that neighbor less useful for predicting new links between two nodes. The Adamic--Adar similarity index (Adamic & Adar, 2003) takes the high degree neighbors into account and is defined as

$$s_{ij} = \sum x \in \Gamma_i \cap \Gamma_j \frac{1}{\log k_x},$$

where Γ_i is the neighborhood of node i and k_x is the degree of node x. Similarly, as for the preferential attachment index, we also include a modified version of the Adamic-Adar index, where we take into consideration whether the killer or the victim is already dead.

3.1.5 Community index

New links in e.g. social networks tend to appear between members that are inside of the same community and only rarely between members of different communities (Girvan & Newman, 2002). We can find densely linked communities where people are killing each other the most and treat the new links in those communities as more likely to occur than the ones outside communities. So that we do not ignore links between communities, we can also count the amount of links that occur between two communities and model the inter-community densities. Let {*C*} be the set of communities output by the Leiden modularity optimization algorithm (Traag, Waltman, & van Eck, 2019), and ci the community of node *i*. Then,

$$s_{ij} = \begin{Bmatrix} m_i/\binom{n_i}{2}, & when\ c_i = c_j \\ m_{ij}/(n_i \cdot n_j), & when\ c_i \neq c_j \end{Bmatrix},$$

where ni is the number of nodes in the community of node *i*, mi the number of links within community of node *i* and mij is the number of links between communities of nodes *i* and *j*. As is the case for previous two indices, we include a modified version of the community index as well, returning a very negative score when the killer or victim are already dead.

3.2 Feature extraction for machine learning approaches

Besides using the classic index-based link prediction techniques, we also make use of a machine learning-based approach, in which we construct features from network properties and additional metadata, and use them to train a classifier. The data used for training the classifier includes properties of the kills network such as PageRank scores for all characters and the basic kills from the original dataset. Computing a score like PageRank (Brin & Page, 1998) on the kills network would be pointless as the nodes have an in--degree of at most one, so the scores would be high for those who already died. We can take a different network of characters into account for this particular case. An online repository of sample networks, found at https://github.com/melaniewalsh/sample--social-network-datasets, contains a sample of the Game of Thrones social network, which creates an edge between two character nodes if they appear within a 15-word distance in the original books. Since the shows are vastly influenced by the books (at least very strongly up to episode 60 to which our kills data is collected) we can use properties from that network as well to gather some additional information. This network has 107 nodes and 352 edges, and is shown in Figure 3.

3.2.1 Standard machine learning framework

The framework we use for predicting links using machine learning is similar to the framework that is described in Section 3.1.1. We split the procedure for getting scores for test links into two parts — first we obtain the scores for positive examples and then the negative examples.

To obtain the scores for positive examples, we first choose some episode, whose kills we are currently trying to predict. These links make up our current test set. Links that appear in the episodes after the selected one are ignored, as we must not predict the past based on future events. The kills that happened prior to the selected episode make up our current training set. In addition to that, we sample the same amount of negative examples in order to make the training set balanced. The classifier is then trained on the training set and used to predict scores for chosen test examples. This is repeated for multiple chosen episodes and at the end we obtain scores for P positive examples, where P contains all the kills that have happened in the chosen episodes.

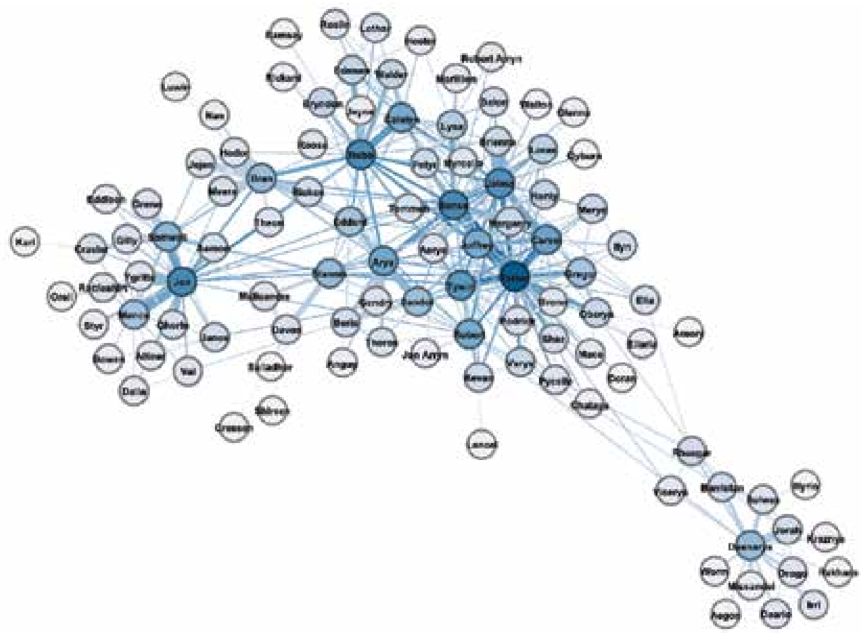


Figure 3: **Game of Thrones social network.**

For negative examples, we take all the kills from our dataset and sample the same amount of non-kills to form a training set. The test set consists of P randomly sampled non-kills from the entire dataset. These test examples are sampled in a way that they do not overlap with negative examples in the training set.
For obtaining the scores, we use three different models: K-nearest neighbors (KNN), logistic regression and support vector machine (SVM) (Hastie, Tibshirani, & Friedman, 2009). The rest of the framework remains the same as the one described in Section 3.1.1.

3.2.2 PageRank

The first feature we use is the PageRank score (Brin & Page, 1998). We can calculate it to find the most important characters as they will hopefully have the highest scores. Being more important could mean two things — either you are very important and thus have a lot of security by guards and other helpers, so you are very unlikely to die, or the exact opposite. The opposite would imply that since this series is oriented around taking the power from others, you are more likely to die if you are very important, which seems more plausible since the show thrives on the sudden deaths of the more popular characters. A very low PageRank score of some character would then also mean that they are very unlikely to have their death portrayed, since the viewers and readers don't care or have forgotten about the least important people in the story. We use two PageRank-based features, the first being *killer_pagerank* and the second *victim_pagerank*, since we are trying to predict killer and victim pairs. The top 5 scoring characters after PageRank calculations are Tyrion (0.055), Jon (0.045), Daenerys (0.041), Jaime (0.037) and Sansa (0.036). From our knowledge of the show, we can safely claim that these are some of the most, if not the most important characters, so we can then be sure that these scores make sense in terms of importance. We assign a mean of all the PageRank scores for each character that is found in the kills dataset, but is not present in the social network.

3.2.3 Betweenness centrality

Another measure that we extract from the social network as a feature is the betweenness centrality (Freeman, 1977). This score measures how many shortest paths from two different nodes go through a certain other node. That means that the higher the score, the more control over a big portion of the network a node has (i.e. is one of the nodes that can make the network split quickly). That should also yield a measure of importance — if one character wants to reduce the power of a part of a network, they can disconnect it from the biggest component by eliminating nodes with high betweenness centrality. Again, we observe that the five top scoring people are also the most important characters in the story: Jon (0.230), Robert (0.209), Tyrion (0.198), Daenerys (0.157) and Robb (0.127). For every character from the kills dataset that is not in the social network, we use a mean of betweenness centrality scores, similarly to the PageRank scores.

3.2.4 Community detection

Simply using the house that a character belongs to as a feature could perhaps help. The intuition behind this is that there might be more kills occuring between members of different houses than between members of the same house. Since our dataset has a lot of undefined values for the houses, we can find communities in the social network using an algorithm such as the Leiden modularity optimization (Traag et al., 2019). We can assign a label of the community to each character and see if that helps us with the prediction by creating some sort of indicator between which groups the kills occur. After running the community detection algorithm on the social network, we can see that we obtain meaningful results (meaningful to people who watch the show) in terms of alliances. The network gets partitioned into five big communities. For example, we can see that Khal Drogo, Daenerys Targaryen, Aegon Targaryen and Jorah Mormont all fall into the same community, even though they do not originate from the same houses, but as we know from watching the show, they are allies. Every character from the kills dataset that is not included in the social network gets assigned to a dummy community.

3.2.5 Automatic feature extraction for machine learning approaches

In addition to handcrafting features we also try an approach using automated feature extraction, specifically node2vec (Grover & Leskovec, 2016) to obtain node and link features. We use these in place of previously handcrafted features.

For a given node, node2vec constructs a feature representation (embedding) that aims to preserve the network neighbourhood properties in a vector space of fixed dimensionality. Depending on parameters p (return parameter) and q (in-out parameter), the algorithm performs different types of biased random walks in order to represent the node's neighbourhood. Setting p to a low value encourages a search that is local to the given node, while setting q to a low value encourages a more explorative search. The sampled neighbourhoods are then used to estimate a feature representation that maximizes the probability of observing these neighbourhoods for the given node. We obtain a link embedding by concatenating together the embeddings of source and target node. If a node has no embedding, a generic embedding for an unknown node is assigned to it. We train the embeddings on the social network, capturing character co- ocurrences. Because we are dealing with a very small dataset, we cannot afford to tune the hyperparameters reliably. Following the findings of authors of the method, we set the parameters to $p = 2$ and $q = 0.5$ since these settings are shown to bias the embeddings to capture homophily. We fix the node embedding size to 16.

4 RESULTS

For our experiments we select and remove links that are associated with kills that happened in seasons four to six. There are 114 of those, to which we add 114 randomly selected unlinked nodes and compute the AUC based on these examples. We repeat this process five times to account for the randomness in selection of negative examples and provide the mean AUC and its standard deviation. We support the AUC scores with precision and recall scores as well, since AUC only measures how well a randomly selected positive example can be distinguished from a randomly selected negative example. For the classic link prediction techniques, we classify examples with scores strictly higher than zero as positive and the others as negative. For the machine learning approach, every example with a score greater than or equal to 0.5 is classified as positive and the others as negative. We then calculate the precision as

$$precision = \frac{\#true\ positives}{\#true\ positives + \#false\ positives}$$

and recall as

$$recall = \frac{\#true\ positives}{\#true\ positives + \#false\ positives}.$$

The obtained results are shown in Table 1.

Results show that the community index achieves the best result in terms of AUC, 0.875. The best precision and recall scores are obtained using the alive index, which are 0.822 and 0.930. Other indices using the death info achieve similar results, however their precision and recall scores are different.

Tabela 1: **Results for four link prediction methods: preferential attachment index, Adamic-Adar index, community index and alive index. The table shows the mean AUC, precision and recall and their standard deviation over five runs. The symbol marks the versions of indices where a check is first performed if the killer or the victim is already dead.**

Method	AUC	Precision	Recall
preferential attachment	0,503 (0,020)	0,522 (0,113)	0,087 (0,000)
preferential attachment †	0,862 (0,015)	0,654 (0,123)	0,070 (0,000)
Adamic-Adar	0,500 (0,000)	0,000 (0,000)	0,000 (0,000)
Adamic-Adar †	0,872 (0,028)	0,000 (0,000)	0,000 (0,000)
community index	0,500 (0,000)	0,000 (0,000)	0,000 (0,000)
community index †	**0,875** (0,018)	0,000 (0,000)	0,000 (0,000)
alyve index	0,863 (0,032)	**0,822** (0,000)	**0,930** (0,000)

Table 2: **Results for three classic machine learning methods (using the basic dataset): K-nearest neighbors, logistic regression and support vector machine (SVM). The table shows the mean AUC, precision and recall and their standard deviation over five runs.**

Method	AUC	Precision	Recall
KNN	**0,632** (0,048)	0,641 (0,024)	0,453 (0,020)
Logistic regression	0,596 (0,036)	**0,797** (0,010)	0,400 (0,013)
SVM	0,556 (0,066)	0,688 (0,024)	**0,523** (0,007)

The Adamic-Adar and community index achieve 0 precision and recall because no links get classified as positive. For the community index, this is due to the network components being very disconnected, while for the Adamic-Adar this is due to the fact that nodes cannot have common neighbors as kills are only attributed to one person, meaning two killers cannot kill the same victim.

Other indices that don't use information about deaths achieve AUC scores around 0.5. This implies that they perform no better than if links were classified randomly.

The standard machine learning approaches came out to be a little bit better than random when using the base kills dataset using only the out-degrees of characters as a feature, with AUC scores ranging from 0.556 to 0.650 as shown in Table 2.

By adding network features (PageRank, betweenness and community identifier for each character) we improve the general performance of all the classifiers and obtain a better top score of 0.686 by using SVM and the handcrafted features. The features created by node2vec yield worse AUC and precision scores and higher recall scores than the handcrafted features. All the final results for the machine learning approaches are shown in Table 3.

Table 3: **Results for three classic machine learning methods (using the augmented dataset): K-Nearest- Neighbors, Logistic regression and Support Vector Machine (SVM). The table shows the mean AUC, precision and recall and their standard deviation over five runs.**

Method	AUC	Precision	Recall
KNN	0,659 (0,035)	0,725 (0,016)	0,453 (0,004)
Logistic regression	0,658 (0,033)	**0,816** (0,016)	0,418 (0,013)
SVM	**0,686** (0,058)	0,719 (0,058)	0,456 (0,056)
KNN (node2vec)	0,650 (0,044)	0,595 (0,052)	**0,702** (0,036)
Logistic regression (node2vec)	0,640 (0,025)	0,600 (0,051)	0,684 (0,028)
SVM (node2vec)	0,605 (0,091)	0,642 (0,062)	0,632 (0,022)

5 DISCUSSION

We see that the sparsity of the network and its size (less that 400 nodes) make link prediction on such a small network very inaccurate in most cases.

Since the original network of kills does not seem to have any community-like structure, it is very hard to predict kills based on community-based link prediction methods, such as the community index. The modularity optimization algorithm finds more than 100 communities with no links between them, which means that the modeling of densities between communities does not give us any information, since the probability of a link occurring between communities is zero. That is a solid foundation for the claim that the authors have done a good job by not creating a very obvious structure of the kills, where someone would kill a lot of people from e.g. their opposing house, making it easy to predict that there will be another similar kill occurring in the following episodes which the viewers have not yet seen.

We cannot achieve good results by constructing indices that try to predict kills using out-degrees only, since the majority of nodes have a very low out-degree (as can be observed on Figure 2).

When we add the death information, we observe that the modified Adamic-Adar index performs as good as the alive index, since it represents the same idea. We know that two nodes cannot have a common successor, since a character can only die once. They can only have one common predecessor, however that implies that both the killer and victim are already dead. By taking the death information into consideration, we automatically decide that when one character is already dead, there will be no link. That makes the common neighborhood factor in the index irrelevant, making it decide the outcome based only on whether a character is alive or not.

The indices that use the death information achieve the best results because of the way the network is constructed. Nodes either have an in-degree of zero or one (depending on whether they were already killed or not). When sampling positive examples in our framework, we remove the edge for that positive example, decreasing the in-degree of the target node from one to zero. When sampling negative examples, we do not remove any edges. However, because our original network is constructed from deaths in the series, most of the characters in our network have died at some point. Therefore, the target node of a negative example is quite likely to have an in-degree of one. These death information indices make use of the fact that when we sample a positive example, we are going to decrease the in-degree of the target to zero, and when we sample a negative example, the in-degree of the target is likely to still be one (i.e. that the target was killed by somebody else at some point). We try to account for this by adding some additional isolated nodes (corresponding to characters that did not kill anyone and did not die), but the index still performs best on the expanded network. So although the baseline index and the other indices which use death information achieve good results, they do not do so by using any structural properties of the network but rather just by abusing the way our network is constructed. The key takeaway here is that achieving an AUC score around

0.85 is not that hard. The hardest part is achieving a noticeable increase in performance over the baseline classifier. The results from the standard machine learning approaches are bad, since we only use the out degree as the basic feature from the original dataset to predict kills. When we augment it with different centrality measures and community identifiers, the performance is improved, mostly because of the fact that we do not only have one feature anymore and because these features are not equally weighted anymore. However, there should be some added value to the features due to what the features represent, which is explained for each index in Section 3.

Among the approaches using node2vec features, the approach using KNN achieves the best results. Visualization of the link embeddings reveals why the approach performs well. Figure 4 shows embedded positive and negative links, projected onto a two-dimensional plane using t-distributed Stochastic Neighbor Embedding (t-SNE) (Maaten & Hinton, 2008). We can observe that the links are often surrounded by links of the same class in their neighbourhood, allowing KNN to correctly predict many links. The visualization also reveals one of the reasons the approaches using node2vec do not perform better: the links where at least one of the nodes does not have a "proper" embedding are embedded closely. The circle-shaped cluster in the visualization represents the links where at least one of the nodes were assigned a generic embedding for unknown nodes (due to some character being present in the social network but not in the kills network). The cluster contains very mixed classes, rendering KNN less useful. In additional experiments using node2vec features, we found that using a link embedding technique where node embeddings are averaged instead of concatenated together (i.e. ignoring direction of the link) results in a significant performance drop. The modified embeddings in combination with KNN result in a mean AUC reduced by 0.068, a mean precision reduced by 0.028 and mean recall reduced by 0.127. This aligns well with intuition since, for example, a notorious killer is more likely to kill an innocent victim than vice versa.

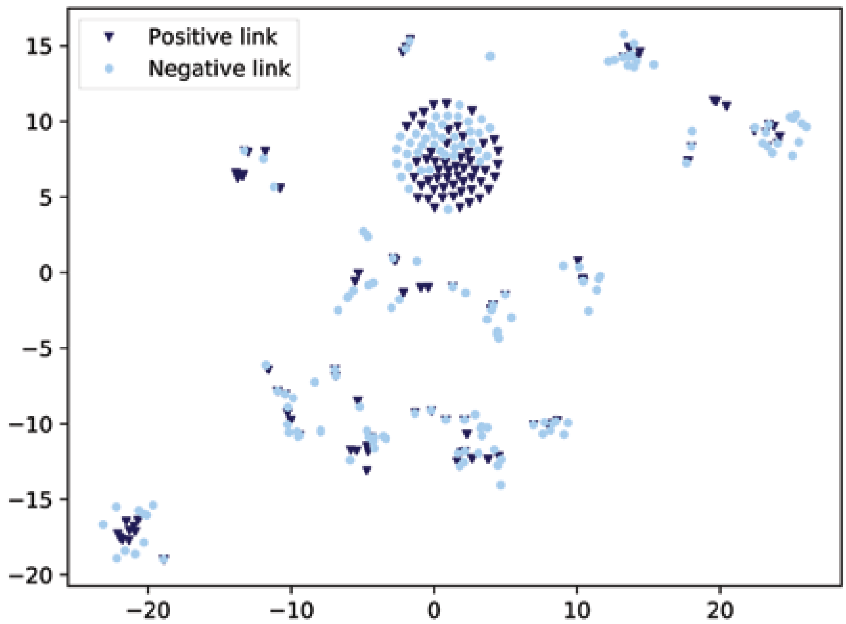


Figure 4: **Node2vec embeddings for positive and negative links in the kills network, projected onto two dimensions using t-SNE.**

6 CONCLUSION

Throughout our work we have acknowledged that the Game of Thrones kills do not have a particularly detectable pattern, since all the kills appear between two nodes that have not yet killed anyone before in most cases. But since our network is small, our test samples are even smaller and that can give deceivingly high AUC scores for indices that would potentially fail on bigger networks and in different scenarios. By using classic machine learning and link prediction techniques, we have found that, on this dataset, no index or feature works better than a simple baseline index (the alive index), which does not model some useful property, but rather abuses the way the network is formed. Most of the techniques used to predict kills in Game of Thrones gave us a fairly good AUC score, but the predictions that our approaches get right do not have a high "shock factor". For example, our classifiers might be able to predict that Jon Snow will kill a wight, but likely fails on less obvious kills. For future work, our approaches could be tested on different TV shows, books or even movies to see how predictable the kills are. Our death information indices could potentially fail on bigger networks with a bit more diverse structure (e.g. having more bigger connected components) and that would give other more basic indices higher accuracy. We could also construct edges using some other information besides kills, e.g. who had a relationship with whom in some show or who scammed whom in a criminal series.

Source code

The source code to reproduce results presented in this paper is available at
https://github.com/matejklemen/got-link-prediction.

REFERENCES

[1] Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3), 211–230. Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. Reviews of modern physics, 74(1), 47.

[2] Angraal, S., Bhatnagar, A., Verma, S., Shergill, S., Gupta, A., & Khera, R. (2018). Risk Factors Associated with Mortality in Game of Thrones: A Longitudinal Cohort Study. *arXiv preprint arXiv:1802.04161*.

[3] Barabási, A.-L., & Albert, R. (1999). Emergence of Scaling in Random Networks. *Science, 286(5439)*, 509–512.

[4] Beveridge, A., & Shan, J. (2016). Network of thrones. *Math Horizons*, 23(4), 18–22.

[5] Brin, S., & Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Seventh international world-wide web conference*.

[6] Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2), 187–202.

[7] Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 35–41.

[8] Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12), 7821–7826.

[9] Grover, A., & Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 855–864).

[10] Guimerà, R., & Sales-Pardo, M. (2009). Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52), 22073–22078.

[11] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

[12] Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat* , 37 , 547–579.

[13] Kossinets, G. (2006). Effects of missing data in social networks. *Social networks*, 28(3), 247–268.

[14] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for Information Science and Technology* , 58(7), 1019–1031.

[15] Liu, D., & Albergante, L. (2017). Balance of thrones: a network study on “Game of Thrones” that unveils predictable popularity of the story. *arXiv preprint arXiv:1707.05213*.

[16] Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6), 1150–1170.

[17] Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t--sne. *Journal of machine learning research*, 9(Nov), 2579–2605.

[18] McGill, M., & Salton, G. (1983). Introduction to modern information retrieval. 1983. *McGraw-Hil, New York* . Newman, M. E. (2001). Clustering and preferential attachment in growing networks. *Physical review E*, 64(2), 025102.

[19] Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1), 5233. doi: 10.1038/s41598-019-41695-z

[20] Watts, D. J., & Strogatz, S. H. (1998, June 04). Collective dynamics of ’small-world’ networks. *Nature, 393*(6684), 440–442.

▪

Jaka Stavanja je študent drugega letnika magistrskega programa Računalništvo in informatika na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Zanimajo ga različne metode strojnega učenja, umetne inteligence, uporaba analize omrežij na še na tak način neraziskanih problemih, primarno pa se ukvarja z razvojem različnih spletnih rešitev.

▪

Matej Klemen je študent drugega letnika magistrskega programa Računalništvo in informatika na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Zanimajo ga različna področja umetne inteligence, predvsem uporaba strojnega učenja v povezavi z obdelavo jezika. Na tem področju trenutno raziskuje uporabo vložitev v povezavi z različnimi nalogami obdelave jezika. V prostem času pomaga pri razvoju odprtokodnih knjižnic za podporo strojnemu učenju, kot je na primer Orange.

▪

Lovro Šubelj je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Diplomiral je leta 2008 na Fakulteti za matematiko in fiziko in Fakulteti za računalništvo in informatiko ter doktoriral leta 2013 na temo analize velikih omrežij. Je avtor ali soavtor več kot petdeset znanstvenih prispevkov in patentov ter urednik prestižnih mednarodnih znanstvenih revij. Njegovo preteklo delo je bilo izbrano kot izjemen znanstveni dosežek v Sloveniji ter predstavljeno na uglednih mednarodnih univerzah kot sta Stanford in UCSD. Sodeloval je že pri številnih uspešno zaključenih raziskovalnih in razvojnih projektih v sodelovanju s podjetji Petrol, Celtra, Optilab, Iskratel in drugimi.

Ne gre le za melodijo: kako Evropa glasuje za svoje najljubše skladbe

Anej Svete, Jakob Hostnik, Lovro Šubelj
Univerza v Ljubljani,Fakulteta za računalništvo in informatiko, Večna pot 113, SI-1000 Ljubljana
as5108@student.uni-lj.si, jh9934@student.uni-lj.si, lovro.subelj@fri.uni-lj.si

Izvleček

Tekmovanje za pesem Evrovizije je priljubljeno mednarodno pevsko tekmovanje, ki ga organizira Evropska radiodifuzna zveza. Zmagovalca določi glasovanje publike in strokovne žirije vsake sodelujoče države, kar pomeni, da analiza glasovalnega omrežja tekmovanja ponuja dragocen vpogled v faktorje, ki, poleg kvalitete nastopov, vplivajo na odločitve za glasovanje.

V članku predstavimo rezultate analize glasovalnega omrežja in opišejo rezultate napovednega sistema rezultatov, ki temelji na dosedanjih uvrstitvah. Opišemo metodologijo in predstavimo podatke, na katerih je bila izvedena analiza. Rezultati med drugim vsebujejo nekaj splošnih lastnosti omrežja, najdene skupine držav, ki si med seboj izmenjajo znatno več točk, kot bi pričakovali, in razpravo o tem, kaj so vzroki tega trenda. Samo na podlagi algoritmične obdelave izpostavimo znana razmerja med državami, kot je favoriziran odnos med Ciprom in Grčijo, med skandinavskimi državami in med državami nekdanje Jugoslavije. Na splošno se izkaže, da sosednje države pogosto sodelujejo, sploh če so kulturno ali geografsko ločene od ostalih. Očitno je tudi, da gradnja takih odnosov izboljša uspeh v tekmovanju in da se trend vključevanja v take skupine povečuje. Hkrati pa je opazen tudi negativen vpliv vpletenosti v odnose zanemarjanja drugih držav na uvrstitev v tekmovanju. Preizkusimo tudi model, ki bi na podlagi zgodovinskega glasovanja in preferenc posameznih držav napovedal prihodnje rezultate, vendar se model izkaže za slabšega v primerjavi z napovedmi, ki jih omogočajo stavnice tekmovanja.

Ključne besede: Tekmovanje za pesem Evrovizije, Pristranskost glasovanja, Struktura skupnosti, Predvidevanje izidov glasovanja

Abstract

The Eurovision Song Contest is a popular annual international song competition organized by the European Broadcasting Union. The winner is decided by the audience and expert juries from each participating nation, which is why the analysis of its voting network offers a great insight into what factors, beside the quality of the performances, influence the voting decisions.

In this paper, we present the findings of the analysis of the voting network together with the results of a predictive model based on the collected data. We touch upon the methodology used and describe the dataset that we are analyzing. The results include a number of general features of the voting networks, the exposed communities of countries that award significantly more points to each other than would be expected and predictions on what the biggest factors that lead to this phenomenon are. Multiple known favourable relationships are exposed, such as the one between Greece and Cyprus, between Scandinavian countries and between the countries of the former Yugoslavia. A general trend of neighbouring countries forming alliances is observed, which is especially apparent if they are culturally or geographically separated from others. Furthermore, it is also observed how membership in communities of common point exchange helps achieve better results in the competition and that this trend is on the rise. At the same time, countries involved in relationships of neglect often achieve poorer results. We also try out a model in order to predict the votes based on the network structure of both previous votes and song preferences of nations, which was found not to offer significant improvement of predictions compared to betting tables alone.

Keywords: Eurovision Song Contest, voting bias, community structure, voting inference.

1 INTRODUCTION

The Eurovision Song Contest (ESC) has been held every year since 1956. Its initial purpose was to unite the European nations after the Second World War and has since evolved into an annual entertainment spectacle followed by millions of people. Every rendition of the contest except for the first featured one song entry by each participating country. The countries involved are mostly European, with the recent addition of some nations from outside the continent, e.g. Israel and Australia.

Although some rules have changed throughout the years, the main principles of the competition remain the same. Each participating nation awards some number of points to the performances chosen by the public and jury of that country. Countries cannot give points to themselves. The song with the highest number of points wins. The current system is in place since 2004 and consists of two semi-finals and a final. Each country gets the same number of points to distribute, and they are equally split between the jury and televoting votes. Both are converted on a scale of points ranging from 1 to 12, with the exception of 9 and 11 points, which are not awarded. This means that every country awards points to 10 performances (EBU, 2019), (EBU, 2019).

The nature of voting offers an intriguing opportunity to explore what European countries base their voting decisions on. Some of the commonly attributed factors include geographical proximity, language similarity (Dekker, 2007), ethnic structure (Spierdijk & Vellekoop, 2006), common history, political preference and cultural similarity (Ginsburgh & Noury, 2008). We try to extract as much information on the deciding factors as possible from the network structure and the nation's attributes to pinpoint the most important influences and to leverage that information to infer future voting choices. We have not found any previous work that tried to use the network information for future predictions. It a challenging task, partly because of the challenge of obtaining enough quality data and partly because of the changing format of the competition. We take these historical differences into account during the analysis. However, since one of our main goals is making future predictions, the most recent results are the most important, and these are enough to expose some main trends.

Besides making predictions, we are also interested in how different similarities between nations correlate to their voting patterns. Just by paying some attention to points distribution, it can be seen that there is some correlation between the points awarded and geographic proximity. We try to leverage the network features (e.g. the strength of bias shown throughout the years and the community structure this implies) to extract useful information more precisely and then reason about the biggest deciding factors on the voting. To achieve that, we perform community detection on the network of shown bias to infer the influences.

There have also been some suggestions that building these friendships allows the participants to achieve better scores and rank higher in the competition. One part of the paper thus also focuses on finding out if this is really the case by finding a correlation between the community structure of specific nations and their success in the competition.

To prevent too much biased voting, some measures have already been taken by the ESC committee. For example, they try to minimize the number of neighboring countries competing in the same semi-final and since only the countries performing that night can vote, neighboring countries have less of a chance to help each other get into the final (EBU, 2019), (EBU, 2019). This measure can of course not be taken in the final.

Another thing that we investigate is the notion of neglect between countries. By this we mean the behavior when countries which are somehow linked to one another seldom award each other a significant number of points. In other words, neighbors that do not exchange points could be regarded as neglecting each other.

When the major influences on the voting behavior are exposed, we turn our attention to the actual predictor of future voting behavior and use the information about the communities in the second part of the project together with some additional data that we learn through song and artist features to try to infer the number of points countries will award in future competitions. The model is used together with betting predictions since they are considered to be the best existing way to predict the outcome of the competition. As described in Data collection and presentation, we gathered data from various sources in hope of making some confident predictions.

The rest of the paper is structured as follows. We review some of the previous work on the topic, ranging from specific analysis of the ESC voting network to the more general methods of examining the data. Then we present the dataset we worked with and how it was obtained. We also present some of its main characteristics. Then we describe in more detail the methodology used. Since some of the needed data turned out to be very difficult to get hold of, some compromises had to be made and these are discussed as well. In the last part we present the results of the project and we end by brainstorming some future work ideas and concluding the paper in a summary.

2 RELATED WORK

Since we mainly deal with analysis of the ESC voting network, we discuss some previous papers covering the topic in terms of applying network techniques on the problem, introducing some ideas and techniques that will prove useful for our project. Although they all deal with the competition as a network problem to some extent, none of them use more advanced network analysis tools such as community detection and link prediction on the graphs, which we implement. Both these methods can then be used for future projections, which is also not dealt with in any of the papers.

(Mantzaris, Rein, & Hopkins, Preference and neglect amongst countries in the Eurovision Song Contest, 2018) investigate different possible explanations for the voting patterns which deviate significantly from a uniform distribution, specifically focusing on the notion that nations try to build reciprocal voting connections that lead to them receiving more points from their "partners", and thus ranking higher. Therefore, they try to find correlation between the number of collusive edges a nation has and their success in the competition. They build on previous work in (Mantzaris, Rein, & Hopkins, 2017), (Gatherer, 2006), analyzing the voting behavior by simulation voting, since analytical identification of statistically significant trends in the competition would be mathematically too complex because of its changing nature. Capturing the different voting systems in place throughout the years mathematically is untraceable, therefore simulation provides a good compromise.

The authors extend the algorithm presented in (Gatherer, 2006) for finding significant exchange of points awarded between participants. The original paper focused on a limited interval of competitions when the voting rules were mostly homogeneous, therefore *Mantzaris et al.* provide a more general sampling technique. To be able to do that, they identify the three principles of voting used by ECS since its start in 1956. These can be grouped as *allocated*, *sequential* and *rated*. The algorithm samples the uniform distribution of points throughout a time period, based on the rules in place at the time and then extracts the highest-weighted edges. Network is formed based on those colluding edges between countries, showing patterns of biased voting. They then perform community detection on obtained structures and base their results on those. They consider both one-way and two-way relationships and thus lay groundwork for thorough network inspection in terms of both motif and community detection.

They find significant patterns of both preference and neglect spanning throughout the participating nations, showing that voting is geographically influenced, linking it to mutual history, similar ethnic features and the feeling of "brotherhood" of neighboring countries. They also conclude that the participants with higher number of colluding edges achieve better success in the competition, showing it does pay off to build partnerships. This, together with the changing nature of the network that is more and more concentrated around the colluding edges, implies that nations are actively trying to build these relationships.

(Dekker, 2007) provides a different take on the analysis of the voting network. The techniques the authors demonstrate have a more general applicability, spanning away from the ESC, and can also be used for analyzing other types of friendship networks. They focus on the votes from the 2005 rendition of the contest and come up with ways to adjust votes for song quality. With that, they produce a friendship network with valued links (the value of the link being the strength of the friendship). They find that friendships are often not returned, which reveals their asymmetric nature, especially visible in countries with a large number of immigrants.

They run a more statistical analysis by removing the influence of song quality or popularity and it shows that friendship between countries is determined in a big part by geographical proximity. Another factor they find are large immigrant groups voting for their home country. Other factors, such as population size, language similarity and economy were found to be insignificant. They expose a visible five-bloc structure, the blocs being the Eastern (former USSR countries, together with Romania, Hungary and Poland), Nordic (Norway, Sweden, Denmark, Finland

and Iceland), Balkan (former Yugoslavia and Albania), Eastern Mediterranean (Greece, Cyprus, Malta, Bulgaria and Turkey) and Western (Portugal, Spain, Ireland, Andorra, Israel, the UK, France, Monaco, Germany, Belgium and the Netherlands). Preferences among the different blocs are also analyzed, finding that some blocs are more connected than others. Grouping countries computationally by exposing the strongly connected components, they find three different blocs. Using taxonomic trees proved to be ineffective and only finding one bloc.

(Ginsburgh & Noury, 2008) analyze 29 years of the Eurovision Song Contest, specifically the competitions held between 1975 and 2003. Its main goal is to find any correlations between the points awarded and country similarity, performance type, etc. The authors find some meaningful properties impacting the scores and extract some clusters that exchange votes regularly. They propose what could lead to this behavior, stating that there exist cliques of countries that award points among themselves and even trading with votes. But these blocs are found not to base on politics, but rather on language and cultural similarities. To measure the language impact, they rely on the Morris-Swadesh method for analyzing linguistic differences.

To infer the influence of each factor, *Ginsburgh et al.* formulate a weighted expression, for which weights are assigned based on the voting behavior. The major takeaway of it is that the biggest factor influencing the voting decision is still the music quality. As with the previously discussed work, they also notice an important role of immigrants that vote for their country of origin. These observations are, however, not algorithmic but rather the results of looking at the formed communities and discussing the prevailing similarities in them.

3 METHODS

3.1 Bias detection

In the first major goal of the paper is determining the community structure of the voting networks. The most important step is the formation of edges that reflect a consistent bias between nations (both in terms of positive and negative relationships) and we approach that in two ways, described in this section. Both methods are used to detect bias over all the selected time periods. Altogether, this gives us more than 4400 different networks which are later used to present some statistical facts about the distribution of points.

```
function Gatherer (start_year, end_year)
  conf_high = bias threshold  // 1 % in our case
  conf_low = neglect threshold  //  90 % in our case
  avg_simulation = []
  // simulate voting enough times to obtain
  // a reliable expectation (100000 times in our case)
  for selected number of iterations:
    simulation = []
    for year in start_year..end_year:
      score = expected number of votes
              received by a contestant
          // depends on the voting scheme
      append(simulation, score)
    avg_sim = mean(simulation)
    append(avg_simulation, avg_sim)
  sort_ascending(avg_simulation)
  positive_bias = percentile(avg_simulation, conf_high)
     // more than bias number of points reflect biased voting
  negative_bias = percentile(avg_simulation, conf_low)
     // more than bias number of points reflect neglect
```

Figure 3.1 **The Gatherer algorithm**

Firstly, we follow the methodology described in (Mantzaris, Rein, & Hopkins, 2017), (Mantzaris, Rein, & Hopkins, 2018), (Gatherer, 2006) and use the Gatherer algorithm. This turned out to be the most effective method and very important for our analysis, thus, we describe the pseudo code in Figure 3.1 and Figure 3.2. Its main idea is to estimate the number of points that a participant is expected to receive in a certain time period, based on the rules in place at that time. It then uses these estimates to find bias, i.e. behavior where nations exchange more than the expected number of points in a time period.

```
function Determine_Bias_Gatherer (start_year, end_year)
  period_length = start_year - end_year + 1
  participants = nations that took part in the ESC
  in the period
  for c1 in participants:
    for c2 in participants:
      if participated_together > period_length / 5
        // only take into account the participants that took
        // part in 20 % of all competitions in that period
        points_awarded_1, points_awarded_2 =
           number of points awarded by c1 to c2
           (and by c2 to c2) in the period
        threshold_high = the threshold number of points
          for that period showing bias
          calculated by the Gatherer algorithm
        threshold_low = the threshold number of points
          for that period showing neglect
          calculated by the Gatherer algorithm
        if points_awarded_1 > threshold_high and
          threshold_high > points_awarded_2:
          // one-way bias
          add edge (c1, c2) with weight
             (points_awarded_1 - points_awarded_2)
             to the directed bias network
        if points_awarded_1 > threshold_high and
          points_awarded_2 > threshold_high:
          // two-way bias
          add edge {c1, c2} with weight
            ((points_awarded_1 + points_awarded_2) / 2 -
            threshold_high)
            to the undirected bias network
        if c1 less than 3 hops away from c2:
          // only consider countries that are close
          // geographically to avoid noise
          if points_awarded_1 < threshold_low
            and points_awarded_2 < threshold_low:
            // two-way neglect
            add edge {c1, c2} with weight
              (threshold_low - (points_awarded_1
               + points_awarded_2) / 2)
               to the undirected neglect network
```

Figure 3.2: **Method of forming a bias voting network based on statistics calculated by the Gatherer algorithm**

The other method of obtaining the structure was developed by us and it accounts for the number of points a country has received each year in the selected period. We create a directed edge from country 1 to country 2 if the first one awarded the second one more than the average number of points received by the second one in more than 75 % of the competitions in that period. If the bias is shown both ways, we add the edge to the undirected network. The pseudo code is described in Figure 3.3.

```
function Determine_Bias_Average (start_year, end_year)
  overshot = defaultdict(int)
  period_length = start_year - end_year + 1
  threshold = 0.75
  // threshold of how many times more than the average
  // number of points need to be awarded for bias to occur
  participants = nations that competed in the period
  for year in start_year..end_year:
    determine the average number of points
      for each participant in the time period
    for c1 in participants:
      for c2 in participants:
        points_awarded_1, points_awarded_2 =
          number of points awarded by c1 to c2
          (and by c2 to c1) in the period
        determine how many times each country has awarded any
          other more than the average number of points
          received by the second in the time period
        appearances_1, appearances_2 =
          number of times c1 (and c2)
          participated in the ESC in the period
        overshot_1, overshot_2 =
           number of times c1 gave more than the average
           number of points received by c2 to c2 in the period
        if overshot_1 > threshold * appearances_2
           and
           overshot_2 < threshold * appearances_1:
           // one-way bias
           add edge (c1, c2) with weight
             (overshot_1 - overshot_2)
             to the directed bias network
           if points_awarded_1 > threshold * appearances_2 and
             points_awarded_2 > threshold * appearances_1:
             // two-way bias
             add edge {c1, c2} with weight
                ((overshot_1 + overshot_2 -
                threshold * appearances_1 -
                threshold * appearances_2) / 2)
                to the undirected bias network
```

Figure 3.3: **Method of forming a bias voting network based on the average number of points received by the countries in the time period**

We have generally found that the simulated voting implemented by the Gatherer algorithm gives clearer and less noisy results. It proves much more useful for detecting neglect, since the average points method generates too much noise. Even the graphs generated by the Gatherer algorithm were tricky to work with, which is why we added an additional criterion to detect a neglect between 2 countries. Since geographic proximity proved to be very important, we also demand that two neglecting countries lie no more than 3 hops (borders) away on the map thus reducing the amount of random edges.

Since the Gatherer algorithm is also used in (Mantzaris, Rein, & Hopkins, 2017), (Mantzaris, Rein, & Hopkins, 2018), (Gatherer, 2006), it is well tested and reliable. Therefore, we focus mainly on its results for graph formation from here on. Although we were able to extract some valuable information with the second method and it performed very similarly to the Gatherer algorithm for the longer periods, it behaves inconsistently on the shorter time spans, picking up too much randomness, while the Gatherer algorithm performs consistently no matter the period length, which led us to this decision.

We form two types of graphs: undirected, showing mutual affinity between contestants (i.e. an undirected edge between two nodes is added if both show bias towards each other), and directed, showing only one-way bias. Here, we are more interested in actual one-way relationships - an edge was therefore added only if one country shows bias towards the other, but the other does not show any bias for the first. Both graphs use weighted edges, the weight denoting the difference between the actual and expected (simulated / average) number of votes.

Despite the consistent performance by the Gatherer algorithm, it still needs some tuning. Besides the noise picked up in neglect detection, it also struggles on the directed networks, adding insignificant edges. This is why we also post-process the directed graphs and remove the edges whose weights were below the average in the network, giving us much more readable results.

3.2 Basic graph features and communities

In order to get the general oversight over the voting networks, we calculate some basic statistics about the voting and bias networks, such as the average number of nodes in a certain time period, the average number of edges and the average degree. All the methods are already implemented in the *networkx* library (Hagberg, Swart, & Chult, 2008).

After extracting the biased voting trends, we extract the communities using the Louvain (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008) community detection algorithm in the undirected and the Newman's leading eigenvector method (Newman, 2006) in the directed ones. Both are implemented in the CDLib library (Rossetti, Milli, & Cazabet, 2019). The extracted communities in the undirected network depict blocs of countries "collaborating" in the competition. The directed networks are analyzed somewhat differently, since the actual communities do not play such a vital role here, as there is no mutual point exchange. However, they still expose some interesting behavior that would be missed if we only focused on the undirected networks. The results of both types are presented in Results.

The number of extracted graphs also allows us to find the most commonly co-occurring nations in communities. Those were extracted with the *apriori* algorithm, implemented in MLxtend library (Raschka, 2018).

The plots throughout the paper were generated with the Gephi visualization tool (Bastian, Heymann, & Jacomy, 2009).

3.3 Correlation between the community structure and success in the competition

One of the main goals of bias edge construction and community detection was determining the affect the bias behavior has on the final score of participants. In other words, we wanted to find out whether being in a large community or having many friends in the network pays off. Thus, based on the communities a node (country) belongs to, we gather some voting data (points, points from community, percentage of points from community and final place) and aggregate it for each community type and period length.

We find the most interesting aggregations to be points per degree in community, portion of points received from communities and total number of points received from communities. Therefore, we decided to interpret those more carefully in Results.

3.4 Preference detection and future projections

One of the hypotheses we set was that users and jury vote based on three main factors: the song popularity and features, the country of the performance and the artist features. We split those categories to subcategories and obtain as much data as possible about them. We build a knowledge graph which connects all possible properties that can be considered together into relations of different types. With this graph we perform similarity scoring and link prediction where we try to predict the "voting relations" based on other connections. We use different link prediction algorithms which have to consider the rich structure of the formed graph. In addition to network analysis techniques, the dataset was also examined with the Orange package (Demšar, et al., 2013).

3.5 Prediction performance evaluation

In the second part of the analysis, we focus on the predictor of the success in the competition. The performance is measured against the performance of the betting tables. We use two different scores to calculate the success of the model. The first score is the mean absolute error (MAE) of the ranks inferred by the predictor based on the actual results and the second score the recall at n (Recall@n) score for n = 3, 5 and 10. MAE, too, is measured at distinct intervals: for the whole set of performing nations and just for the top 10 performances each year.

4 DATA COLLECTION AND PRESENTATION

4.1 Collection

The data set used was obtained by scraping various web pages. The voting data was collected from (European Broadcasting Union) and the information about specific countries, songs and performers was downloaded from (Flecht, 2019) (Wikimedia Foundation, Inc, 2019). The available voting data includes all points awarded by every country to every other participant, both for the final and the semi-finals (when both were held), with the exception of the first ever competition in 1956, since the data is not available. For the period between the years 2016 and 2019 we even got separated votes from jury and audience, since this is when the EBU started sharing these figures.

For most countries we obtained their names, Wikipedia category entries, languages, the currency, calling code, ethnic groups, religions, neighborhood. For the participants we have their country of origin, how old they were when the represented their nation, name, Wikipedia categories, music genres, instruments and occupations. Data for songs was scrapped from Wikipedia. We have among other things the genres, categories, languages and released date. To analyze songs even better we scrapped lyrics, chords and scores from (Fandom, Inc., 2019), (Musixmatch , 2019), (Musescore BVBA, 2019), (Naidenov, 2019). The biggest challenge presented the data about songs, performers and performances themselves. We have tried to obtain as much as we could from Wikipedia, at least for the latest entries, which were better represented. We therefore focus mainly on those. We also try to extract some other important properties (the tones, harmony, metrum, melody...) about song quality from the chords and scores and the prevailing themes and motifs with text mining. Those features will be useful to pinpoint the preferences of specific countries and the factors that contribute most to success.

We have also obtained the betting tables for each competition between the years 2004 and 2019 (Eurovision World Betting Odds). These allow us to combine our models with the expected outcomes based on the betting odds. The data is stored in structured JSOG format (extended JSON format which can work with references and is therefore better for graphs). The scraping was done in Java, but data analysis is done in Python because of its numerous robust libraries for data management.

4.2 The inferred networks

Based on the collected voting information, we are able to form a large number of graphs, showing the voting behavior throughout the history of the competition.

Firstly, we just create the voting network for each contest separately and for all of them together (the all-time voting network). The networks for each competition are directed and any edge between two countries depict the number of points awarded by one to another in that year. The in-strength of any node therefore shows its total score that year. Similarly, the all-time directed network depicts the total number of points awarded in the competition history.

These networks are then used to form the bias networks as described in Methods. To observe the changes throughout the competition history, we opt to form networks that represent biases and neglect in certain periods - those were chosen to be 1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 63 years. For each period, both directed and undirected networks are created. This gives us more than 4400 networks altogether, but we do not need to analyze all thoroughly. The main focus are the networks that show the all-time preferences (period length 63 years), the ones that depict different 10 year periods, since this can show any changing nature of the voting, and the ones that depict the last 20, 25 and 30 year periods, showing long-term but still recent trends.

The all-time voting network has 52 nodes, one for each country that has ever competed. The 10 most successful (the nations with the highest number of points collected throughout the history) were Sweden, Norway, the UK, Germany, France, Spain, Denmark, Greece, the Netherlands, and Ireland. The countries with the lowest number of points obtained so far have been Monaco, Bulgaria, Australia, San Marino, Montenegro, Czech Republic, Slovakia, Andorra and Morocco. However, these scores should not be too surprising and taken too seriously, since the most successful nations are also the ones that have participated in the competition the longest and many of the least successful ones have only taken part a few times. On the other hand, Australia has only participated 5 times so far and has achieved great success each time, which cannot be captured with this kind of analysis.

4.3 Betting tables accuracy

The baseline for measuring the performance of our prediction model was using the betting tables as the only means for predicting the outcome of the competition. Thus, this baseline needed to be determined. The results were obtained using the performance metrics described in Preference detection and future projections, averaged over the whole period for which we have obtained the betting tables. To limit how much was known about the outcome of the competition when the tables were updated, we only included data from 20-35 days before the final rounds of the competition. They are presented in Table 4.1. We can see the MAE of the tables improves for the higher part of the table, while the recall does not seem to be affected much by the range. These figures are the baseline for our model, which provided results described in Prediction performance evaluation.

Tabela 4.1: **Performance evaluation of betting tables as predictors.**

Performance measure used	Results
Mean averaged error over the whole set	4,3391
Mean averaged error over the top 10	4,0421
Recall@3	0.4386
Recall@5	0,54737
Recall@10	0,56842

5 RESULTS

The results present the communities of positive bias, countries showing neglect, the correlation between the community structure of a country and its success, what we think causes this behavior, the inferred preferences of specific countries and the prediction results.

5.1 Communities

The number of generated networks makes it possible to reason about the different trends and influences on the voting. Although we could have focused on any period in the competition history, we chose to further inspect the most recent results and mostly summarize the older.

Figure 5.1 shows the communities formed if we consider the results from the start of the competition in 1956. There are 10 communities in total and they are strongly geographically influenced, forming the following blocs: Northern (Sweden, Denmark, Norway, Iceland), Western (Ireland, United Kingdom, Germany, Luxembourg), Southern (Italy, Malta, Spain, Portugal), Central (Netherlands, Hungary, Belgium, Austria), Baltic (Lithuania, Estonia, Latvia, Finland), Eastern (Poland, Ukraine, Russia, Belarus), the Balkan (Greece, Cyprus, Albania), South-Western (Moldova, Romania, Turkey), Yugoslavian (Croatia, Slovenia) and Cross-Continental (Israel, France). Especially prominent are the connections between Cyprus and Greece, Greece and Albania, Romania and Moldova, Italy and Malta and the former USSR countries.

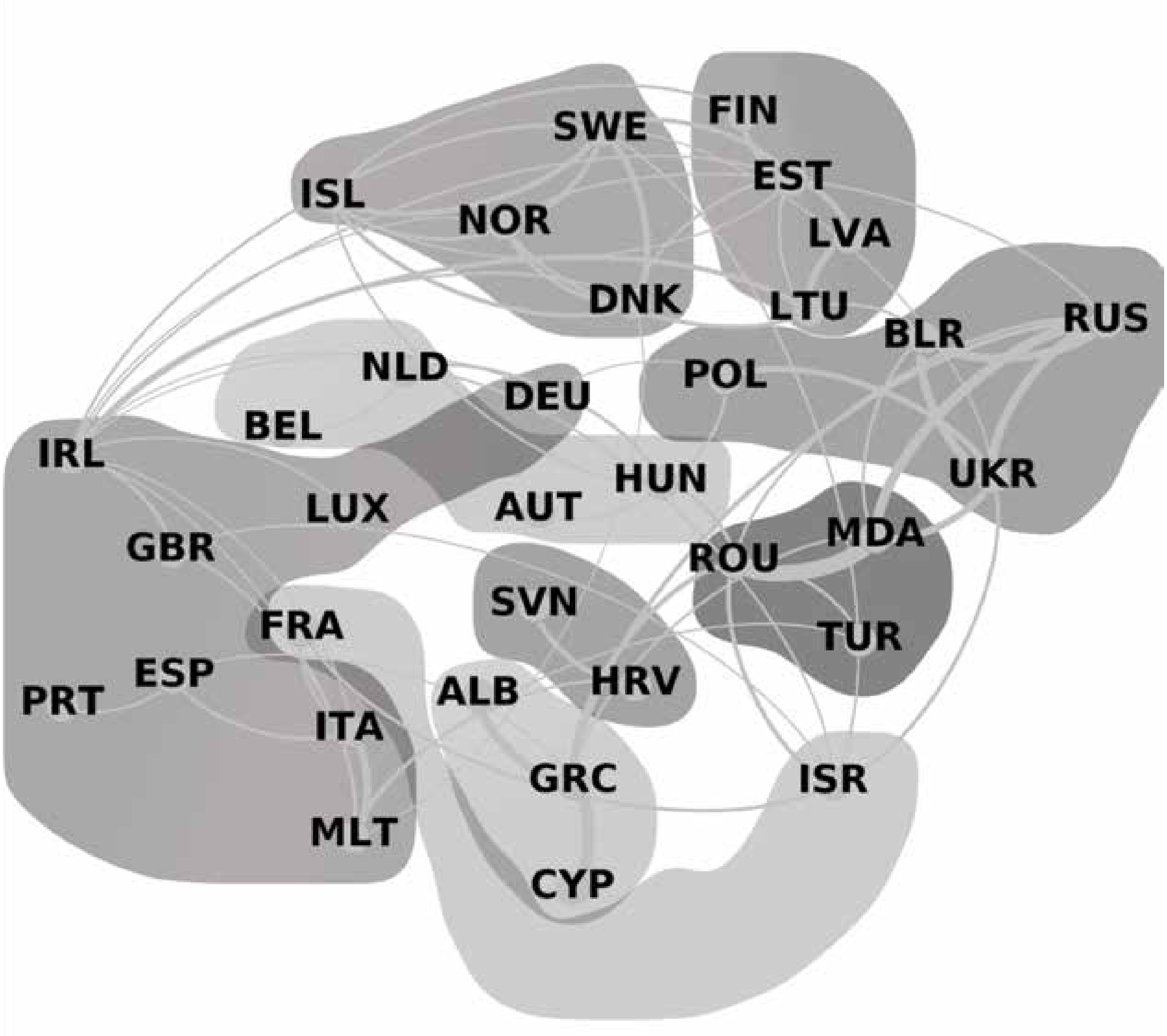


Figure 5.1: **Bidirectional bias from the start of the competition.**

Although this is the network that includes the most data and is thus seemingly the most important, we only mention it here for the sake of completeness. We are more interested in the networks depicted in Figure 5.2, Figure 5.3 and Figure 5.4, and for the later parts of the analysis as they speak of the more recent trends.

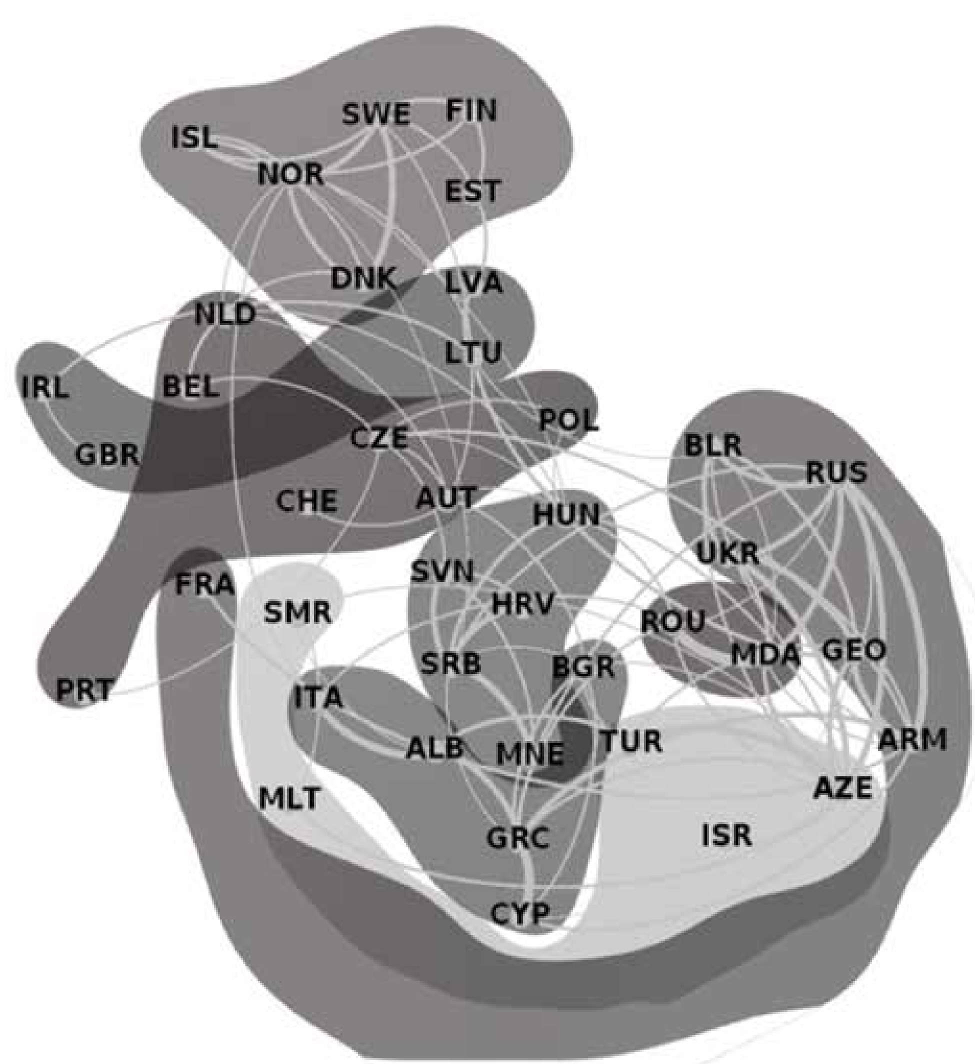


Figure 5.2: **Bidirectional bias from the last 20 years (1999-2019).**

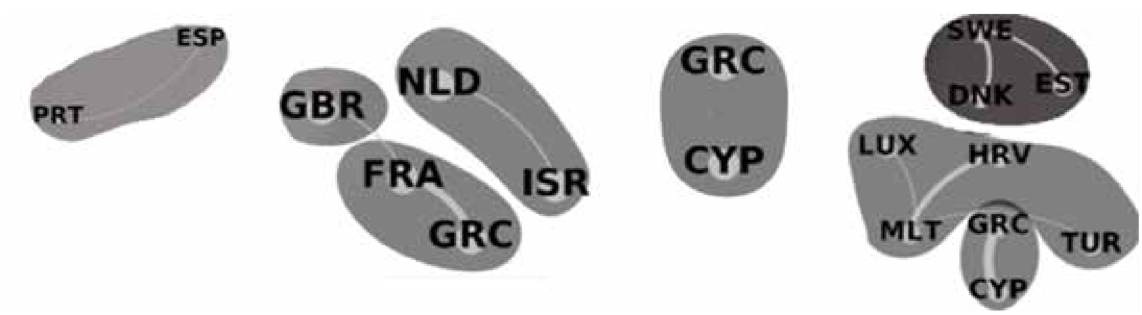


Figure 5.3: **Bidirectional bias in 10-year periods (1959-1969, 1969-1979, 1979-1989, 1989-1999).**

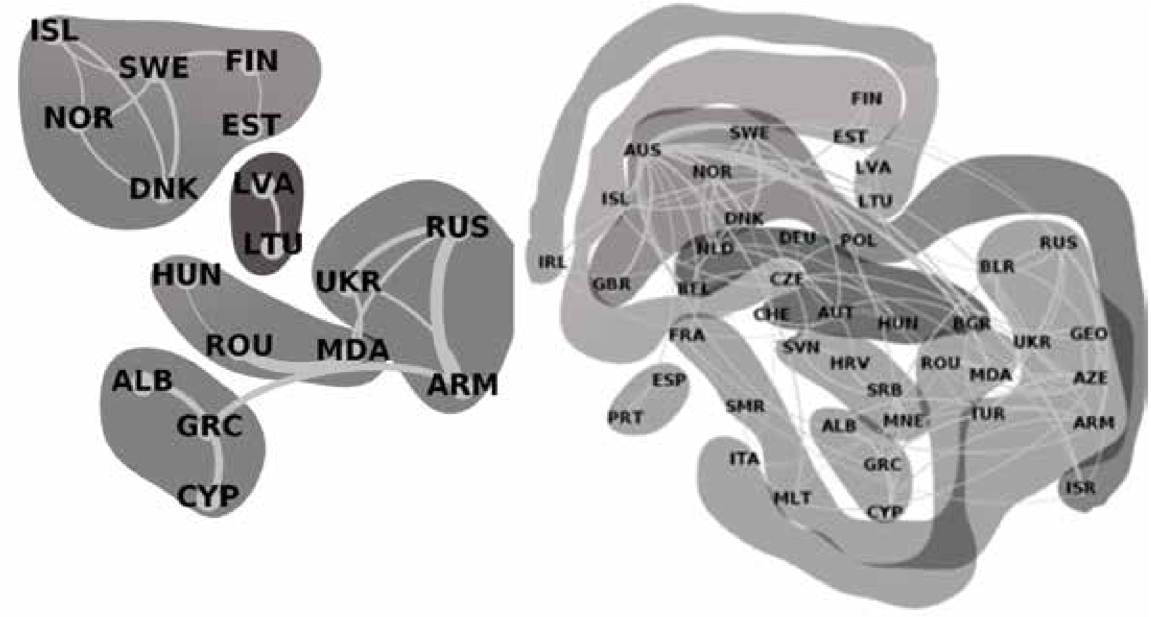


Figure 5.4: **Bidirectional bias in 10-year periods (1999-2009, 2009-2019).**

Figure 5.3 and Figure 5.4 show how the bias networks have evolved and grown, although the main communities remain the same. Clearly, there is more and more biased voting, but it remains concentrated in the same blocs in all periods. It is interesting to see how Australia got mixed into the Northern bloc in the last 10 years. This may be one of the reasons for their reasonable success so far. During the first five time they have taken part in the competition, they showed a very focused voting behavior and at the same time managed to collect many points from until then a very closed bloc.

We think the most interesting and current network is the one in Figure 5.2, since it shows the recent trends, while still taking into account a longer time period. The communities are very similar to the ones implied by the all-time bias network, showing the persistence of these relationships.

Also interesting is the network shown in Figure 5.5, showing the all-time network of one-way relationships. The edges depict relationships where only one country awards more than average number of points and the other do not. Communities are not that prominent in this network, but still visible. One reason why the community structure is limited in the fact that historically very successful countries such as Sweden receive a high number of points from others very often and they cannot "return" the votes to all of them. Therefore, they have a very high in-degree and this does not infer any preference, just the fact that they were successful. However, some relationships in the all-time network are still quite interesting, like the strong edge from Croatia to Bosnia and Herzegovina and the edges from the former USSR nations to Russia.

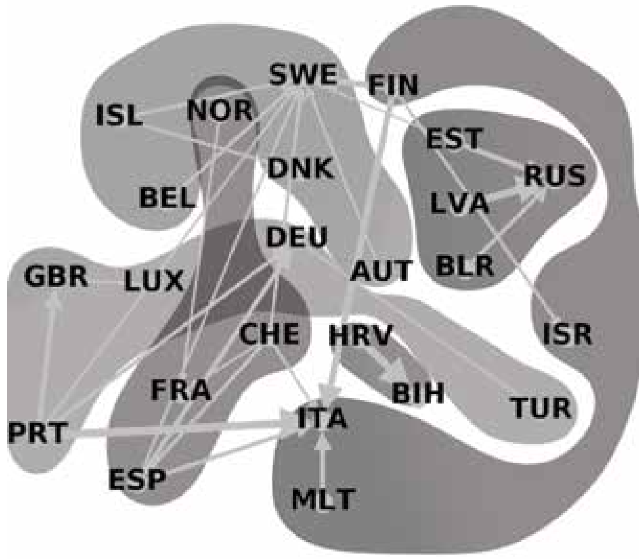


Figure 5.5: **Unidirectional bias from the start of the competition.**

The data also allows us to find sets of countries that end up in the same communities most often. The results are presented in Table 5.1. As the table shows, the countries that co-occur in a community most often are Cyprus and Greece, which are a part of more than 10 % of the formed communities. They are followed by some Scandinavian countries and the most regular participants in the competitions, such as the UK, Ireland, and Switzerland. The most common set of size three contains Denmark, Sweden, and Norway. We also notice a strong relationship between Portugal and Spain, Romania and Moldova, Slovenia and Croatia and, interestingly, France and Israel. All the relationships are also visible in the figures below.

The graphs in Figure 5.3 and Figure 5.4 provide a different view as to how the bias has evolved throughout the history and it is clear that there are more and more biased connections. This can also be seen if we look at the average degree of the bias undirected network throughout the history, depicted in Figure 5.6. The degree has been rising consistently, which means that the countries are actively forming more and more friendship communities and concentrating their votes among specific "partners".

Tabela 5.1: **Countries that ended up in the same community most often and the relative number of times.**

Rank	Countries	Relative support
1	Cyprus, Greece	0,109
2	Denmark, Sweden	0,081
3	Sweden, Noreway	0,069
4	Switzerland, United Kingdom	0,066
5	Denmark, Norway	0,062
6	United Kingdom, Ireland	0,062
7	Denmark, Sweden, Norway	0,055
8	Spain, Portugal	0,053
9	Sweden, Iceland	0,043
10	Germany, United Kingdom	0,042
11	Denmark, Iceland	0,041
12	Romania, Moldova	0,037
13	Belgium, Netherlands	0,036
14	Slovenia, Croatia	0,035
15	Israel, France	0,034
16	Denmar, Sweden, Iceland	0,033
17	Norway, Iceland	0,033
18	Germany, Iceland	0,033
19	Finland, Sweden	0,029
20	Estonia, Litvia	0,028

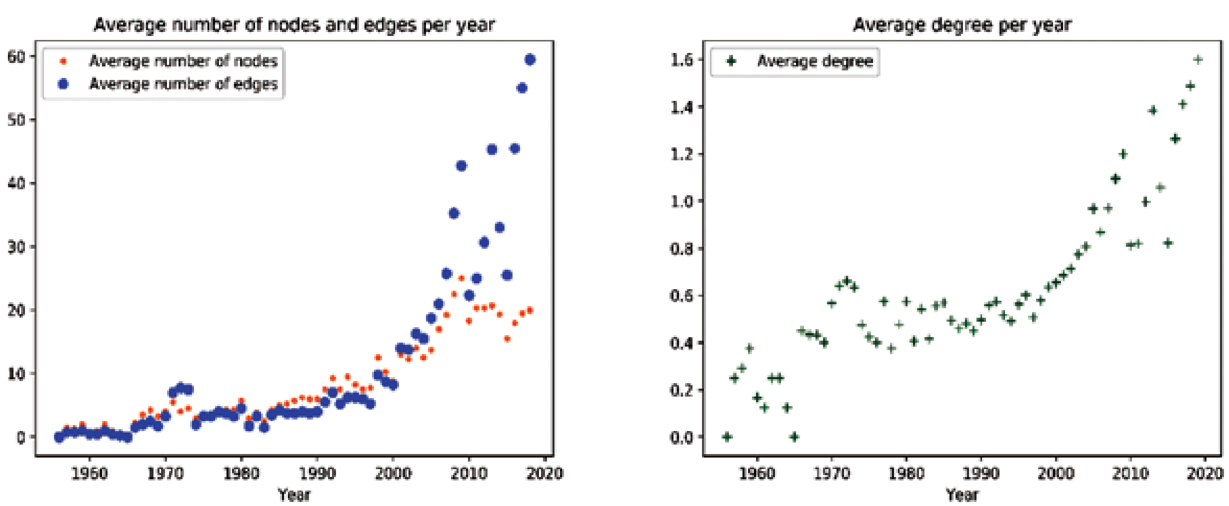


Figure 5.6: **Average number of nodes and edges, average degree and clustering coefficient in the bias networks throughout the years.**

5.2 Correlation between the community structure and success

Table 5.2 shows what percentage of points countries got from their communities. For example, averaged over 25 years, members of neglect communities only received 6.1 % of their points from that community, while members of communities in directed graphs get on average 17.6 % of their points from that cluster.

Table 5.3, Table 5.4 and Figure 5.7 show a recurring relationship between the success of a nation and its community structure. Being in a positive bias community pays off because countries get on average higher scores and achieve higher places, a trend clearly visible in the plots. We can also see that it is better to avoid neglect clusters, since membership in those usually means lower ranking.

Table 5.2: **Average percentage of points received from communities..**

	bidirectional bias		**bidiretional nelgct**	
Period	**Average**	**STD. deviation**	**Average**	**STD. deviation**
1	192.80	121.30	10.00	0.00
5	532.97	272.12	179.53	122.62
10	812.57	402.08	334.40	216.89
20	1160.40	466.33	656.59	485.73
63	2877.68	732.65	2151.29	871.49

Table 5.2: **Average percentage of points received from communities..**

	bidirectional bias		**bidiretional nelgct**	
Period	**Average**	**STD. deviation**	**Average**	**STD. deviation**
1	0.22	0.10	0.03	0.05
5	0.18	0.09	0.01	0.06
10	0.18	0.09	0.03	0.04
20	0.17	0.09	0.05	0.06
63	0.15	0.06	0.09	0.06

Table 5.3: **Average place.**

	bidirectional bias		**bidiretional nelgct**	
Period	**Average**	**STD. deviation**	**Average**	**STD. deviation**
1	4.02	8.85	1.00	0.00
5	11.28	12.19	11.24	7.41
10	14.76	11.74	19.94	10.59
20	15.29	10.93	23.03	11.22
63	18.47	10.45	28.42	11.56

Table 5.4: **Average number of points.**

	bidirectional bias		**bidiretional nelgct**	
Period	**Average**	**STD. deviation**	**Average**	**STD. deviation**
1	192.80	121.30	10.00	0.00
5	532.97	272.12	179.53	133.62
10	812.57	402.08	334.40	216.89
20	1160.40	466.33	656.59	458.73
63	2877.68	732.65	2151.29	871.49

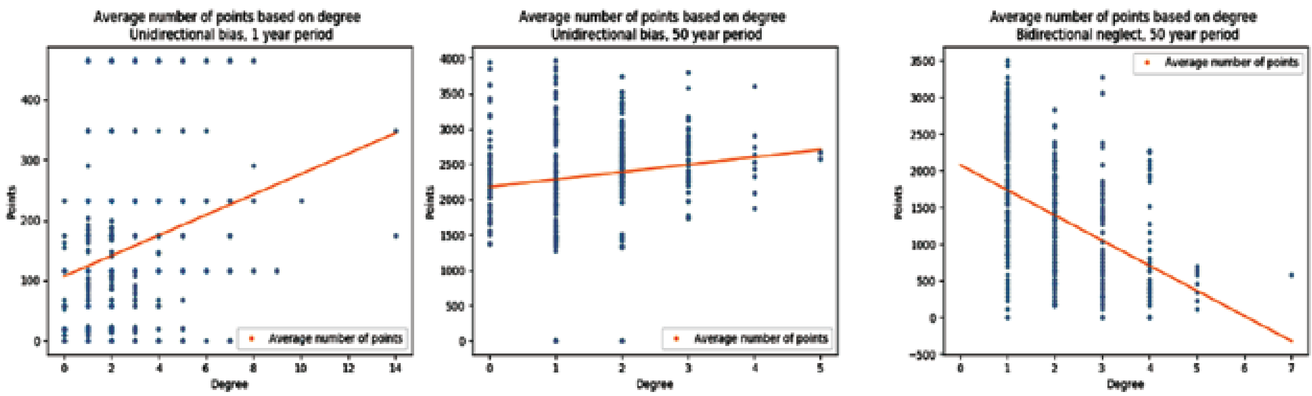


Figure 5.7: **Relationship between the degree in the bidirectional bias (left and center) and neglect (right) and the number of points received in the last 50 years.**

5.3 Neglect

We form the neglect networks in a similar manner to the positive bias ones and the results are shown in Figure 5.8, Figure 5.9 and Figure 5.10. As expected, some distinct neglect relationships are visible between nations, most notable between Macedonia and both Greece and Cyprus. Similar holds for the pair Cyprus and Turkey and more recently, for Azerbaijan and Armenia. Interestingly, there is also a strong evidence of neglect between Germany and the pair Belarus and Ukraine. As seen in the more recent networks, the trends persist.

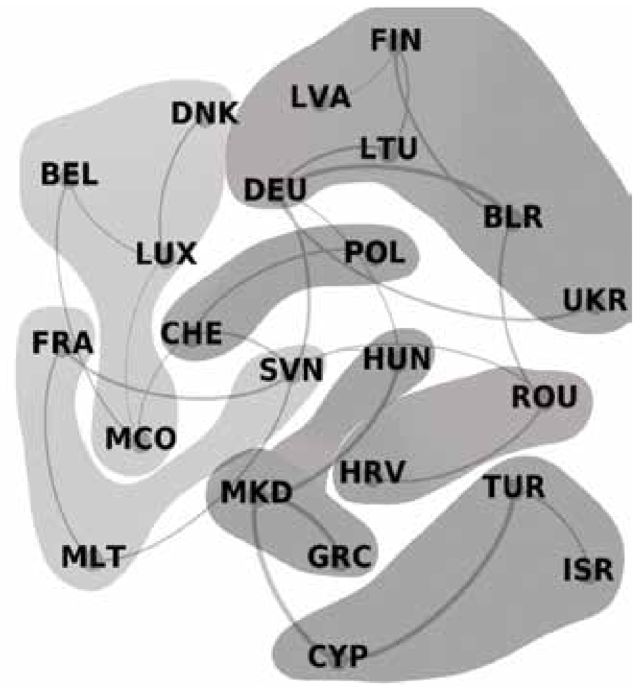


Figure 5.8: **Bidirectional neglect from the start of the competition.**

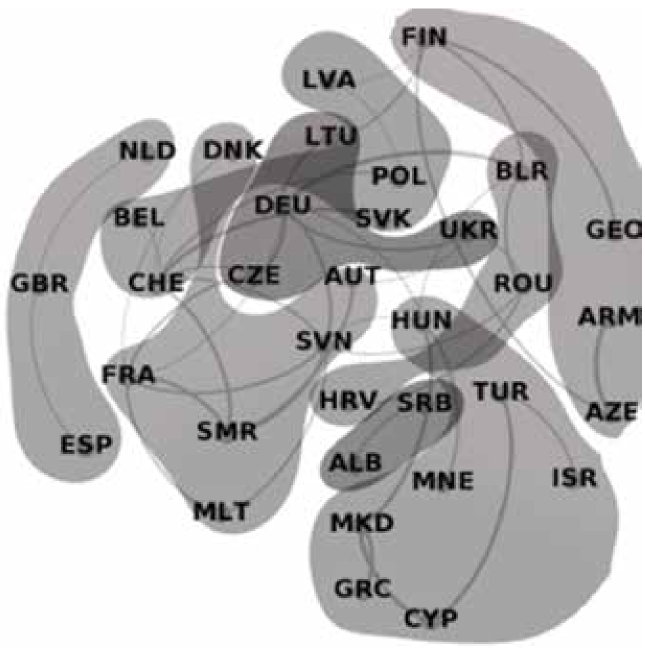


Figure 5.9: **Bidirectional neglect from the last 30 years (1989-2019).**

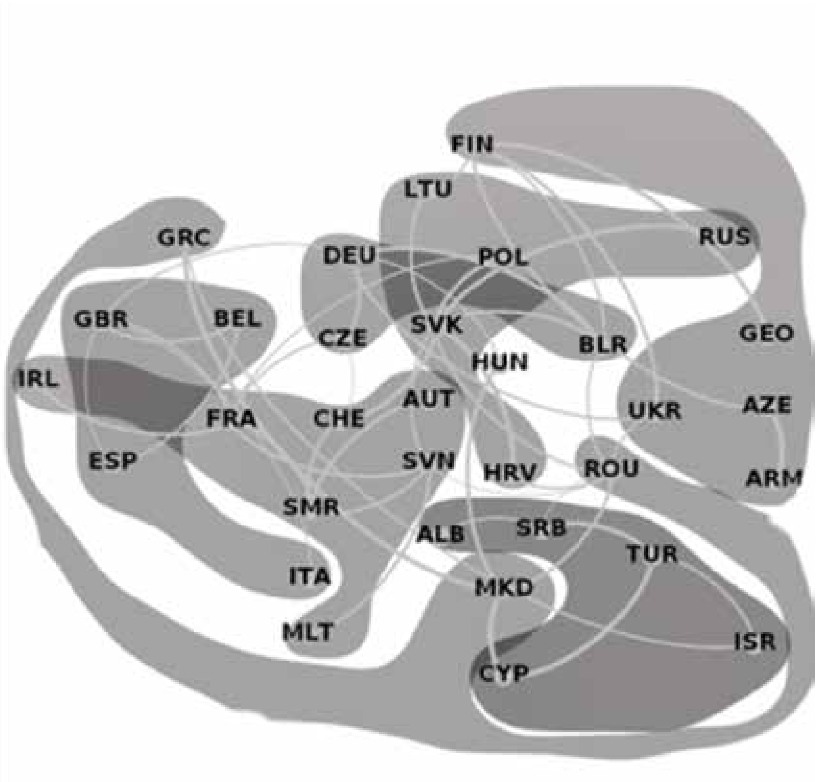


Figure 5.10: **Bidirectional neglect from the last 10 years (2009-2019)**

5.4 Possible influences and motivations

As found in the discussed literature, geographical proximity seems to influence the voting behavior most, as can be seen through the geographically local communities that form. Moreover, affinity between nations such as the UK and Malta stress that language similarity also plays a role. Common historical background could be attributed to the affinity between the former Yugoslavian and USSR nations, since the communities rarely extend beyond the bounds of the former unions.

The one-way relationships are trickier and less obvious. However, they can be explained to some extend by the number of immigrants (e.g. votes from Croatia to Bosnia and Herzegovina, Germany and France to Turkey and Switzerland to Serbia) and historical significance of one country to the other (e.g. the votes from the former USSR countries to Russia). Other reasoning is hard to ground since the highest in-degrees can be explained purely on the success in the competition.

It is worth noticing that the positive bias behavior is most strongly represented by pairs of nations that are more isolated, either geographically (e.g. Spain and Portugal, the UK and Ireland, Romania and Moldova, the Scandinavian countries), or culturally (e.g. Cyprus and Greece, Greece and Albania, the Baltic countries and, again Romania and Moldova).

The countries showing most neglect have notable reasons as well. Especially the historical relationship between Macedonia and Greece and more recently between Albania and Serbia can be explained by their non-friendly neighborhood relations.

5.5 Nation's music preferences

The constructed knowledge graph and Orange visualization tools offer a glimpse into what genres, music styles and other performance features caught the voters' attention. Unfortunately, due to the lack of data, we are only able to extract the crudest of relationships, thus, we do not discuss them here thoroughly. Some trends we observe, though, are the fondness of Slovenia towards Croatian songs (both in the form of the language and the origin), Australia towards songs in English and we again confirm the strong relationship between Greece and Cyprus.

5.6 Prediction performance evaluation

The performance measures indicate that the built model did not increase the accuracy of the betting tables. Much of this can be attributed to the fact that the data was often very sparse and not structured

very well. Even after preprocessing and filtering the whole dataset, we were still left with too many unreliable and altogether not very useful entries.
In Table 5.5 we report the performance measures when we also consider the predictor data together with the data from the betting tables in variable amounts. The hyperparameter β indicates how much the predictions made by our model are taken into account (β =0 means only the betting tables are used and β =1 means we rely only on our predictor). We can see, the predictor does not improve the betting tables performance.

Table 5.5: **Performance evalutation of our prediction model.**

Performance measure used	Results	Results	Results
Mean averaged error over the whole set	6.00	6.52	6.76
Mean averaged error over the top 10	6.48	7.46	7.80
Recall@3	0.14	0.09	0.07
Recall@5	0.23	0.20	0.12
Recall@10	0.42	0.35	0.31

6 DISCUSSION

This section presents some of the problems faced during the analysis and concludes the paper with our closing remarks.

6.1 Problems and compromises

The incompleteness of the data has turned out to be a problem very early on, as we were initially unable to construct graphs based on some similarities, namely the ethnic groups, immigrant numbers and economic exchange. We thus resorted to manual inspection of the probable causes of some trends. The inferred relationships are thus based only on our domain knowledge and presumptions.

As expected, the availability of the data about the performances, songs and authors is also limited, but we have managed to obtain a reasonably diverse and complete dataset and hoped we could make use of it, especially in the second part of the paper.

Another problem we encountered was the noise in the less robust networks such as the directed ones and the ones dealing with neglect. They needed a lot of tuning and some post-processing to present any usable information, but the final outcome is still quite non-deterministic and open to numerous interpretations.

Motif counting and detection was also found to be not as effective as we had hoped. The process of extracting the motif structure itself was not very straight-forward since the functionality is not as widely implemented as some other tools and at the same time the results were not as informative and interpretable as the community structure itself. For example, the notion of the reciprocal point exchange is summed up in the undirected positive bias networks. Thus, we think that a thorough inspection of the motif structure would not provide better enough understanding of the network. We therefore abandoned this idea and focused on other analysis tools.

If we were able to manage the dataset deficits in the first part of the paper, they really came forward in the second part, since the shortcomings disabled us to build a valid and useful model for prediction. We leave this feat for future work.

6.2 Future work

During the analysis, we came across a few possible applications to other fields. Firstly, the ideas and method discussed here do not necessarily apply only on the Eurovision voting network. Such analysis can be applied to any voting system, especially ones with a smaller number of voting entities, such as the participating countries discussed in this paper. We would find analysis of the voting behavior in sports where points are awarded by judges from different countries very interesting. Similarly, taking a closer look at the voting for awards would presumably reveal interesting trends. One of such awards is the Ballon d'Or prize in soccer, where journalists and players from around the world vote for the best footballer each year. Each nation is represented by its journalists and players, which is similar to the voting structure of the ESC.

A different field we would also be interested in is the voting a political environment such as the European Parliament. As representatives from the whole EU vote for propositions which come from different backgrounds, one might find some trends in the way the representatives from specific countries vote.

Lastly, we consider our own implementations and dataset. Some methods we implemented did not take into account all the specifics in the ESC dataset (e.g. the change of Macedonia to North Macedonia was handled manually) and could be extended to further increase the result reliability. One of the main objectives for future work would also be the aforementioned expansion of the dataset that could allow a better model of the behavior.

6.3 Summary and conclusions

In this paper, we analyzed the trends in the ESC voting network. The results show strong and recurring patterns of mutual point exchange between neighboring countries. We observed the most commonly recurring friendships and one-way relationships together with some persistent behavior of neglect. As discussed in the previous work, they can be explained by geographical proximity and language similarity, as well as ethnic structure and historical bonds. Having a large number of biased relationships positively correlates to the success in the competition and we observed more and more relationships the more recent years. Isolation of sets of countries seems to make bonds among the members of the set stronger.

We also described the methodology used in more detail and explained how the data was structured to obtain the information. The obtained data was then used to build predictor for future contests. To the extent possible, we leveraged the distinct music preferences of individual nations to extract which genres and music styles achieve the greatest success in different countries. This involves both the points given by the nation to other countries for their performances and also their representative artists. This data was combined with betting tables, since they are widely considered to be the best predictors about the success of participants. The resulting model did not outperform the betting tables alone with its main weakness being the lack of reliable data.

We look forward to future extensions of our work on similar fields or the same project with a more promising prediction model.

REFERENCES

[1] Bastian, M., Heymann, S., & Jacomy, M. (2009, 3). Gephi: An Open Source Software for Exploring and Manipulating Networks. doi:10.13140/2.1.1341.1520

[2] Blondel, V., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008, 4). Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics Theory and Experiment, 2008*. doi:10.1088/1742-5468/2008/10/P10008

[3] Dekker, A. (2007, 1). The Eurovision Song Contest as a ‚Friendship' Network 1. *Connections, 27*.

[4] Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., . . . Zupan, B. (2013). Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research, 14,* 2349-2353. Retrieved from http://jmlr.org/papers/v14/demsar13a.html

[5] EBU. (2019, October 20). *European Broadcasting Union*. Retrieved from European Broadcasting Union: https://www.ebu.ch/home

[6] EBU. (2019, October 20). *Eurovision Song Contest*. Retrieved from Eurovision Song Contest: https://eurovision.tv/

[7] Eurovisionworld. (2019, October 20). *Eurovision World Betting Odds*. Retrieved from Eurovision : https://eurovisionworld.com/odds/eurovision

[8] Fandom, Inc. (2019, October 20). *LyricWiki*. Retrieved from Lyrics Fandom: https://lyrics.fandom.com/wiki/LyricWikilecht, M. (2019, October 20). *ESCHome*. Retrieved from ESCHome: https://eschome.net

[9] Gatherer, D. (2006, 3). Comparison of Eurovision Song Contest Simulation with Actual Results Reveals Shifting Patterns of Collusive Voting Alliances. *Journal of Artificial Societies and Social Simulation, 9*.

[10] GeoDataSource. (2019, October 16). *Country Borders*. Retrieved from GitHub repository: https://github.com/geodatasource/country-borders

[11] Ginsburgh, V., & Noury, A. G. (2008). The Eurovision Song Contest. Is voting political or cultural? *European Journal of Political Economy, 24*, 41-52. doi:https://doi.org/10.1016/j.ejpoleco.2007.05.004

[12] Hagberg, A., Swart, P., & Chult, D. (2008, 1). Exploring Network Structure, Dynamics, and Function Using NetworkX.

[13] Lao, N., Mitchell, T., & Cohen, W. (2011, 1). Random Walk Inference and Learning in A Large Scale Knowledge Base., (pp. 529-539).

[14] Mantzaris, A. V., Rein, S. R., & Hopkins, A. D. (2017). Examining collusion and voting biases between countries during the Eurovision song contest since 1957.

[15] Mantzaris, A. V., Rein, S. R., & Hopkins, A. D. (2018, 9 01). Preference and neglect amongst countries in the Eurovision Song Contest. *Journal of Computational Social Science, 1*, 377-390. doi:10.1007/s42001-018-0020-2

[16] Musescore BVBA. (2019, October 20). *Musescore*. Retrieved from Musescore: https://musescore.com

[17] Musixmatch . (2019, October 20). *Musixmatch*. Retrieved from Musixmatch : https://www.musixmatch.com/

[18] Naidenov, E. (2019, October 20). *Ultimate Guitar*. Retrieved from Ultimate Guitar: https://www.ultimate-guitar.com/

[19] Newman, M. (2006, 10). Finding Community Structure in Networks Using the Eigenvectors of Matrices. *Physical review. E, Statistical, nonlinear, and soft matter physics, 74*, 036104. doi:10.1103/PhysRevE.74.036104

[20] Raschka, S. (2018, 4). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *The Journal of Open Source Software, 3*, 638. doi:10.21105/joss.00638

[21] Rossetti, G., Milli, L., & Cazabet, R. (2019, 12). CDLIB: a python library to extract, compare and evaluate communities from complex networks. *Applied Network Science, 4*. doi:10.1007/s41109-019-0165-9

[22] Spierdijk, L., & Vellekoop, M. H. (2006, 2). *Geography, culture, and religion: Explaining the bias in Eurovision song contest voting.* University of Twente, Department of Applied Mathematics.

[23] Wikimedia Foundation, Inc. (2019, October). *Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/

▪

Anej Svete končuje dodiplomski študij računalništva in matematike na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Zanima ga splošno področje podatkovnih ved, še posebej obdelava naravnega jezika, uporaba umetne inteligence v robotiki in spodbujevano učenje, kar je tema njegovega diplomskega dela. Med študijem izkušnje nabira tudi pri razvojnem podjetju XLAB d.o.o. Magistrski študij nadaljuje na področju podatkovnih znanosti na univerzi ETH.

▪

Jakob Hostnik je študent prve stopnje univerzitetnega študija računalništva in matematike na Univerzi v Ljubljani, je ustanovitelj podjetja Etilk d.o.o. in v podjetju HPE d.o.o. devops in tehnični vodja ekipe. Profesionalno in zasebno se najbolj ukvarja managementom, arhitekturo programske opreme, devops delom in računalništvom v oblaku.

▪

Lovro Šubelj je docent za računalništvo na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Diplomiral je leta 2008 na Fakulteti za matematiko in fiziko in Fakulteti za računalništvo in informatiko ter doktoriral leta 2013 na temo analize velikih omrežij. Je avtor ali soavtor več kot šestdeset znanstvenih prispevkov in patentov ter urednik prestižnih mednarodnih znanstvenih revij. Njegovo preteklo delo je bilo izbrano kot izjemen znanstveni dosežek v Sloveniji ter predstavljeno na uglednih mednarodnih univerzah kot sta Stanford in UCSD. Sodeloval je že pri številnih uspešno zaključenih raziskovalnih in razvojnih projektih v sodelovanju s podjetji Petrol, Celtra, Optilab, Iskratel in drugimi.

Tehnologije navidezne in obogatene resničnosti, kot orodje za predstavitev novih idej in produktov na sejmih: primer Mahepa

Simon Kolmanič[1], Maršenka Marksel[2], Domen Mogus[1], Borut Žalik[1]
[1]Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor
[2]Univerza v Mariboru, Fakulteta za gradbeništvo, prometno inženirstvo in arhitekturo, Smetanova ulica 17, 2000 Maribor
simon.kolmanic@um.si, marsenka.marksel@um.si, domen.mongus@um.si, borut.zalik@um.si

Izvleček
Navidezno in obogateno resničnost lahko uporabimo tudi na področju predstavitev novih izdelkov in idej širši javnosti. Z njima lahko uporabnike in potencialne kupce seznanimo z izdelkom že v času njegovega razvoja. V članku opisujemo naše izkušnje s predstavitvijo rezultatov projekta MAHEPA. Gre za evropski projekt v okviru programa Obzorje 2020, znotraj katerega se razvijata dva hibridna pogona za letala prihodnosti. Ker so leti s tema pogonoma predvideni šele v letu 2020, je bilo možno napredek v razvoju obeh pogonskih sistemov letal na letalskem sejmu AERO 2019 predstaviti zgolj s pomočjo virtualnih modelov. V članku predstavljamo naše izkušnje s pripravo predstavitve, temelječe na navidezni in obogateni resničnosti, ter pokažemo, da so takšne predstavitve konkurenčne klasičnim in predstavljajo cenovno ugodnejšo alternativo. Predstavitev smo razdelili v tri sklope, pri čemer smo uporabljali očala HoloLens, čelado HTC Vive, s kodami QR pa smo krmilili tudi predstavitev na mobilnih napravah. Vse tri predstavitve so bile izdelane z igralnim pogonom Unity.
Ključne besede: virtualni razstavni eksponati, razvoj virtualnih predstavitev, navidezna resničnost, obogatena resničnost.

Abstract
Virtual and augmented reality can also be used for the presentation of new products and ideas to the public. With their help, new products can be introduced to the users and potential buyers already during the product development stage. In this article, our experiences with the presentation of the research results of the MAHEPA project are presented. MAHEPA is a European Horizon 2020 project focusing on the development of two hybrid powertrains for airplanes of the future. Because the first flights with these propulsion systems were not planed until 2020, the presentation of the powertrains development progress during the AERO 2019 flight-expo was only possible with the use of virtual models. In the article, we present our experiences with the development of the presentation based on virtual and augmented reality and show that such presentation can constitute a competitive and cost-effective alternative to classical exhibitions. Our presentation was divided into three distinct parts, using HoloLens glasses and the HTC Vive headset, while QR codes were used for controlling the research results presentation on mobile devices. All three presentations were developed in the Unity game engine.
Keywords: Virtual exhibits, virtual exhibition development, virtual reality, augmented reality.

1 UVOD

Vedno večja ozaveščenost pri varovanju okolja izpostavlja potrebo po čistejših pogonskih tehnologijah tudi v letalski industriji. Za razliko od cestnih vozil, je razvoj tovrstnih pogonov v letalski industriji šele na začetku. V okviru projekta MAHEPA se tako na dveh različnih štirisedežnih letalih, Panthera in HY4, razvijata dva hibridna pogona. Pogona se razlikujeta po načinu generiranja električne energije. V primeru Panthere za to poskrbi običajen letalski motor, ki ga poganja kerozin, v HY4 pa se v ta namen uporabljajo vodikove gorilne celice. Z namenom seznanitve širše javnosti z dosedanjimi rezultati raziskav, je bilo treba rezultate predstaviti na pomembnejših letalskih sejmih. Eden najpomembnejših za splošno letalstvo v Evropi je sejem AERO s preko 33.000 obiskovalcev. Ker pa letala s pogoni, ki so predmet raziskav, v tem trenutku še ne letijo, je bilo treba poiskati alternativne atraktivne metode predstavitve, ki bi bile primerljive z razstavljenimi modeli letal in letalske tehnike. Pri tem smo se odločili za predstavitev hibridnega pogona letala Panthera z uporabo navidezne in obogatene resničnosti.

Čeprav sta navidezna in obogatena resničnost tehnologiji, ki sta prisotni preko 20 let [2,16], pa sta zaradi zahtevne in dokaj drage opreme, ki jo za to potrebujemo, še vedno relativno redki v našem vsakdanjiku in zato še vedno vzbujata precejšnjo pozornost. Tako navidezna kot obogatena resničnost se uspešno uporabljata na mnogih področjih, od izobraževanja, v muzejih, promociji turizma, zabavni industriji [1,5,6,10,11] in še kje. Izkušnje raziskovalcev kažejo, da je glavna prednost predstavitev s to tehnologijo ta, da lahko uporabnik interaktivno sodeluje pri predstavitvi. Zaradi tega imajo tovrstne predstavitve veliko skupnega z razvojem in igranjem iger, kar še posebej uspešno uporabljajo v muzejih in pri ohranjanju kulturne dediščine [15]. Ob tem pa smo bili soočeni še z dodatnimi zahtevami, in sicer so morali biti uporabljeni modeli letala Panthera in njegovih komponent kar se da realistični in točni, celotna predstavitev pa je morala biti konkurenčna razstavljenim resničnim letalom. Glede na to, da so bile dosedanje izkušnje raziskovalcev vezane na virtualne predstavitve, ki so se odvijale ločeno od realnih, je bilo treba rešitev za to šele poiskati. Ne glede na to, da tehnologija to omogoča, raziskovalci v svojih predstavitvah niso posvečali veliko pozornosti fotorealizmu objektov, pa tudi s tem, kako pritegniti pozornost obiskovalcev, se niso ukvarjali. Pri tem je še posebni izziv predstavljalo dejstvo, da je izkušnja navidezne ali obogatene resničnosti vezana na posameznega uporabnika in ne na skupine, kakršne srečamo na sejmih. Tako smo predstavitev razdelili v tri dele: predstavitev, krmiljeno s kodami QR, pritrjenimi na nos enega od maket letal, predstavitev z očali HoloLens[1] in predstavitev s pomočjo čelade za navidezno resničnost HTC Vive[2]. Vsebina prvih dveh predstavitev je bila dokaj podobna, medtem, ko smo v predstavitvi z navidezno resničnostjo prikazovali notranjost pilotske kabine letala, ki so jo uporabniki hkrati videli tudi na velikem zaslonu. Takšna zasnova se je izkazala za uspešno, saj je razstavni prostor MAHEPE obiskalo zelo veliko obiskovalcev. Samo predstavitev smo kasneje uspešno preizkusili tudi na razstavišču konference Creactivity[3] v Italiji, kjer smo testirali odzive obiskovalcev in beležili njihovo uporabniško izkušnjo.

2 ZASNOVA IN IZDEVAVA VIRTUALNE PREDSTAVITVE

Glede na to, da je uspešna virtualna predstavitev podobna igri, je tudi njeno snovanje in implementacija podobna načrtovanju in implementaciji igre. Tako smo za izdelavo uporabili grafični pogon Unity. Unity 3D[4] je zelo priljubljeno orodje za izdelavo mobilnih iger in aplikacij, temelječih na navidezni in obogateni resničnosti [4,8,9,18]. Da bi zagotovili čim večjo natančnost objektov v predstavitvi, smo izhajali iz CAD modelov Panthere in njenih pogonskih sklopov.

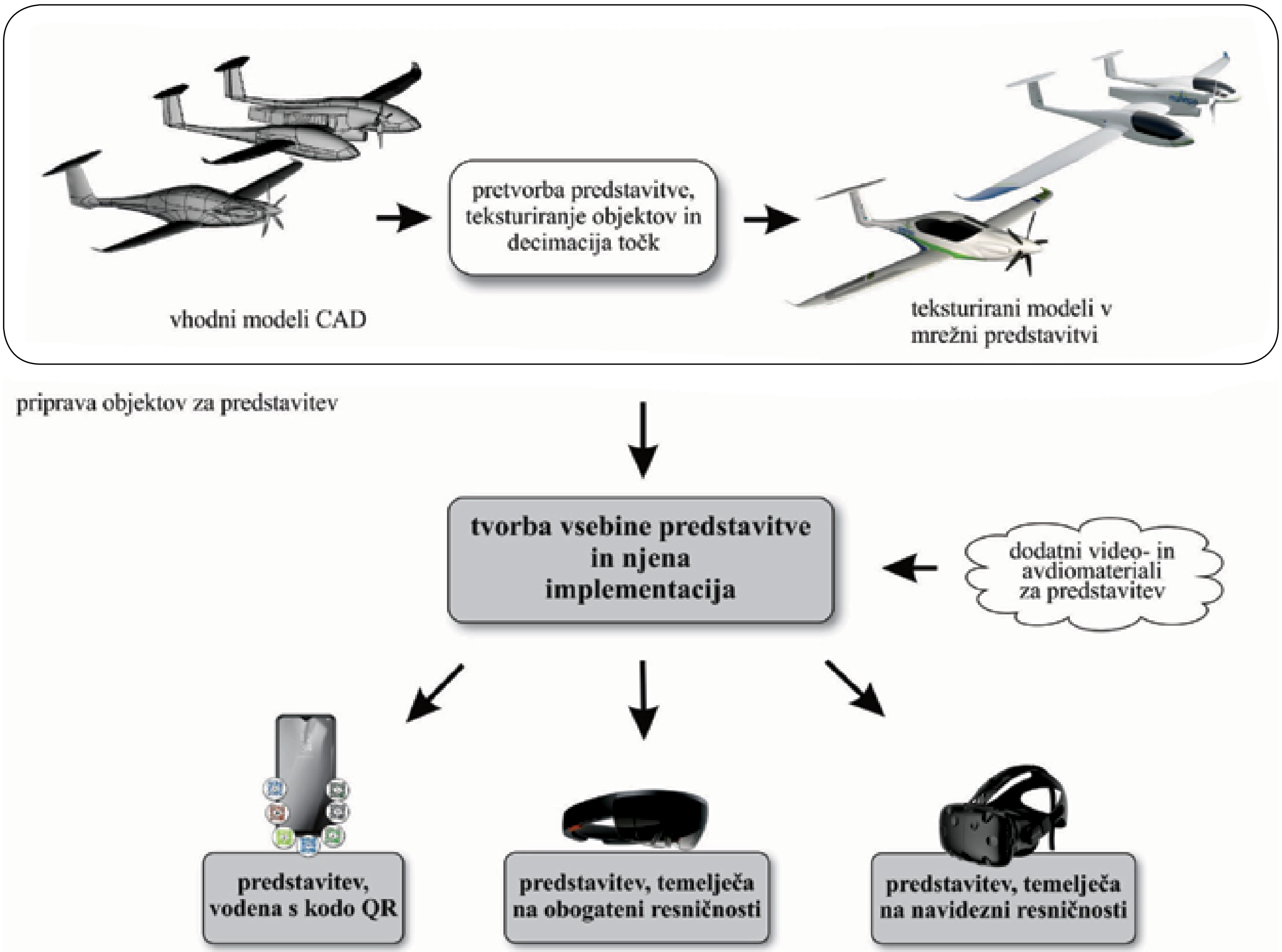
Čeprav so modeli v sistemih za industrijsko načrtovanje podobni tistim v animacijskih paketih in grafičnih pogonih, pa je med njimi vseeno velika razlika. Modeli v aplikacijah za industrijsko modeliranje in načrtovanje objektov imajo veliko natančnejšo predstavitev, medtem, ko so materiali in njihove lastnosti, na primer teksture in vizualni efekti drugotnega pomena. Tako je bilo potrebno najprej vhodne modele ustrezno pripraviti, da smo jih lahko uporabili v predstavitvi. Shematski postopek izdelave predstavitve je prikazan na sliki 1.

2.1 Priprava objektov za predstavitev

Ključna naloga je sprememba geometrijske predstavitve objektov. Pri modeliranju trdnih teles v aplikacijah CAD se namreč najpogosteje uporablja predstavitev z ovojnico [7], ki temelji na eksplicitnih funkcijskih modelih (najpogosteje NURBS [12]), medtem ko grafičnimi pogoni temeljijo na aproksimaciji površja z mnogokotniško mrežo [17]. Glede na to, da so bili v našem primeru modeli izjemno kompleksni (samo pogonski sklop je bil sestavljen iz 1800 delov, zdru-

[1] https://www.microsoft.com/en-us/hololens (zadnji obisk 5. 2. 2020)
[2] https://www.vive.com/eu/ (zadnji obisk 5. 2. 2020)
[3] https://www.progettocreactivity.com/creactivity19/ (zadnji obisk 5. 2. 2020)
[4] https://unity.com/ (zadnji obisk 5. 2. 2020)

[5] https://www.blender.org/ (zadnji obisk 5. 2. 2020)



Slika 1: **Shematski prikaz gradnje predstavitve rezultatov razvoja hibridnih pogonov letal.**

ženih v preko dvajsetih komponentah), je bilo delo časovno zelo zahtevno. Kot rezultat smo dobili množico objektov z gostimi mnogokotniškimi mrežami (pogonski sklop je sestavljalo $5{,}3 \cdot 10^6$ trikotnikov), kar je presegalo zmogljivosti prikaza večine mobilnih naprav, vključno z očali HoloLens. Tako je bilo nujno zmanjšati število točk, za kar smo uporabili animacijski paket Blender[5], s katerim smo dosegali zelo dobre stopnje decimacije točk, brez vidne izgube kvalitete, kar lahko vidimo v tabeli 1. Skozi celoten proces decimacije smo modele testirali s pomočjo testnih uporabnikov, ki so ocenjevali, ali je model vizualno ustrezen, ali pa na njem opazijo kakšne napake. V kolikor se je to zgodilo, smo model zavrgli in uporabili tistega z več točkami.

Ker je bila naša virtualna predstavitev postavljena ob bok atraktivnim realnim predstavitvam, smo veliko pozornosti polagali v zagotavljanje fotorealizma uporabljenih modelov. Tako smo se z veliko skrbnostjo posvetili izdelavi čimbolj vernemu izgledu materialov in tekstur. Kot rezultat te faze, smo dobili objekte, ki so se lahko primerjali z realnimi (slika 2).

Tabela 1:**Število trikotnikov v modelih pred in po decimaciji**

Ime modela	**pred decimacijo**	**po decimaciji**	**stopnja decimacije**
lupina letala	$340 \cdot 10^3$	$316 \cdot 10^3$	1,076
letalski motor	$2 \cdot 10^6$	$421 \cdot 10^3$	4,751
celotni hibridni pogonski sklop	$5{,}3 \cdot 10^6$	$1{,}01 \cdot 10^6$	4,636

[5] https://www.blender.org/ (zadnji obisk 5. 2. 2020)

a) b)

Slika 2: **Objekti, uporabljeni v predstavitvi, a) primerjava virtualnega modela (zgoraj) in realne makete letala (spodaj), b) končani hibridni pogonski sklop na razstavnem prostoru projekta MAHEPA v sklopu sejma AERO 2019.**

Tako slika 2a, kot slika 2b sta vizualizirani s pogonom Unity na očalih Microsoft HoloLens. Primernost objektov za predstavitev smo testirali še s pomočjo tehničnega osebja, ki je sodelovalo pri razvoju teh sistemov. S tem smo dobili dokončno potrditev, da so virtualni modeli dovolj dobri, da lahko prepričajo tudi domenske eksperte na sejmih.

2.2 Priprava vsebine predstavitev in implementacija

Pri pripravi vsebine predstavitve smo se naslanjali na izkušnje drugih raziskovalcev s tega področja [3,13,15] in vse tri predstavitve zasnovali po vzoru resnih iger [14] (angl. serious games), in uporabniku omogočili interakcijo z okoljem in predmeti predstavitve. Še najmanjša je bila interakcija med uporabnikom in predstavitvijo pri predstavitvi, krmiljeni s kodami QR, saj se je tam vsebina prikazala v odvisnosti od kode, ki jo je uporabnik prebral s pametnim telefonom ali tablico. Glede na razširjenost platforme Android smo se odločili za podporo zgolj njej, kar ni predstavljalo kakšne ovire za obiskovalce na sejmu, saj je bila tablica z aplikacijo uporabnikom ves čas na voljo. Znotraj te aplikacije smo predstavili vse pomembne komponente hibridnega pogona, skupaj z osnovnimi tehničnimi podatki, lokacijo baterij in rezervoarjev za gorivo. Aplikacija je delovala v kombinaciji z realno maketo letala, na katero smo pritrdili kode QR (slika 3b). Maketa drugega letala je bila dvignjena v zrak in tako od daleč vidna zain-

a) b)

Slika 3: **Uporaba treh različnih predstavitev na sejmu AERO 2019, a) ureditev razstavnega prostora, prirejena virtualnim predstavitvam, b) kode QR na nosu ene od maket letala.**

a) b) c)

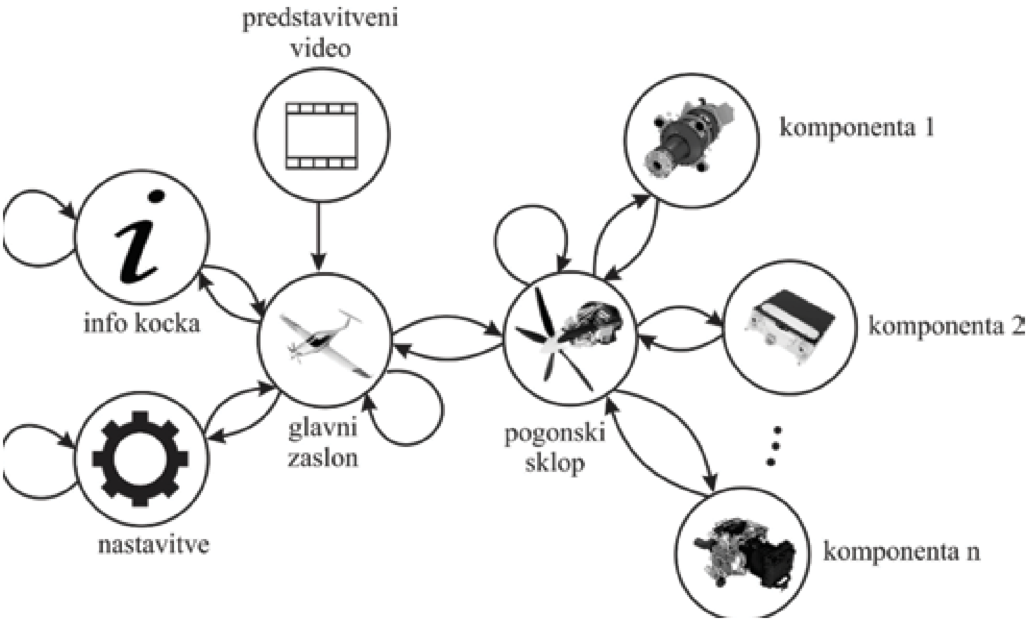
Slika 4: **Trije različni zasloni predstavitve, krmiljene s kodami QR, a) osrednji zaslon za branje kode QR, b) zaslon s podatki o projektu MAHEPA, c) tehnični podatki o uporabljenih elektromotorjih.**

teresiranim obiskovalcem, glej sliko 3a. Ker sta obe letali v tem segmentu dobro znani, je bilo to izredno pomembno za privabljanje pozornosti obiskovalcev, še posebej, ko predstavitvi, temelječi na obogateni in navidezni resničnosti nista bili v uporabi.

Na sliki 4 lahko vidimo primer glavnega zaslona v predstavitvi, krmiljeni z branjem kode QR (slika 4a) in dva zaslona, ki se prikažeta v odvisnosti od izbrane kode (sliki 4b in 4c).

Zaradi oddaljenosti mobilne naprave od ušesa obiskovalca v času delovanja aplikacije, smo vse podatke podali zgolj v obliki besedil in preglednic. To pa ni bilo potrebno ob uporabi očal HoloLens in predstavitvi, podprti z obogateno resničnostjo. Očala HoloLens so namreč zasnovana tako, da imajo vgrajene zvočnike, s katerimi lahko pričaramo občutek 3D zvoka, ki se odlično zlije z zvoki okolice. To nam je omogočilo, da smo predstavitev komponent opravili s predvajanjem zvočnega posnetka njihove predstavitve, kar je uporabniku veliko bolj prijazno, vključili pa smo tudi predstavitveni video. Zaradi vsega opisanega je bila predstavitev na HoloLensu postavljena za centralno predstavitev. Po sami informativnosti je popolnoma primerljiva s predstavitvijo, krmiljeno s kodami QR, je pa omogočala veliko več interakcije z objekti kot prva. Predstavitev smo zasnovali tako, da je imel uporabnik v vsakem trenutku nadzor nad tem, do katerih informacij bo dostopal in do katerih ne, oziroma, kdaj bo s predstavitvijo zaključil.

To smo dosegli tako, da smo predstavitev razdelili v različne sekcije, med katerimi se je uporabnik lahko premikal s klikom na posamezne objekte, kar lahko predstavimo z diagramom prehajanja stanj, prikazanim na sliki 5. Zanke v posameznih sekcijah pomenijo, da je znotraj te sekcije možnih več aktivnosti, ki pa se od sekcije do sekcije razlikujejo. V sekciji glavnega zaslona na primer lahko uporabnik pogleda letalo, mu odstrani pokrov motorja, in si ogleda postavitev pogonskega sklopa. Letalo lahko rotira, ga prime in poljubno pomika po prostoru. S klikom na pogonski sklop preide v isto imensko sekcijo, kjer si lahko pobliže ogleda najpomembnejše komponente pogona, s klikom na njih pa dostopa do tehničnih podatkov o posameznih komponentah, prikazanih v ločeni sekciji.



Slika 5: **Diagram prehajanja stanj med različnimi zasloni predstavitve na HoloLensu.**

Kljub skrbnemu načrtovanju pa je kot neznanka ostalo, ali bodo obiskovalci, ki se bodo prvič srečali z očali HoloLens, zaradi specifike rokovanja z napravo imeli težave z navigacijo in kako bo to vplivalo na njihovo uporabniško izkušnjo. Zato smo v predstavitev vključevali elemente uporabniških vmesnikov, ki jih poznajo iz osebnih računalnikov, to je ikone in sistem navigacije med posameznimi zasloni, ki je podoben navigaciji med posameznimi okni na namizju. Kot dodatno funkcionalnost pa smo dodali tudi možnost premikanja letala po prostoru, ki je predstavitev nekoliko bolj približala igri.

Kar se tiče predstavitve, temelječe na navidezni resničnosti, smo se odločili za manj zahtevno predstavitev. Prikazali smo samo notranjost pilotske kabine, ki jo dobimo z uporabo 360° slike, izdelane na podlagi pilotske kabine obstoječega klasičnega letala, ki je služil kot izhodišče pri razvoju novega hibridnega pogona. V kabino je dodanih še nekaj panelov, ki predstavljajo spremembe, ki jih bo v kabino prinesel novi pogon. S premikanjem glave se tako lahko razgledamo po kabini, lahko pa prisluhnemo tudi razliki v hrupu pri popolnoma električnem letu in letu z vključenim generatorjem za proizvodnjo električne energije. Vsebino te predstavitve lahko vidimo na sliki 6.

Vse tri predstavitve smo implementirali s pomočjo grafičnega pogona Unity 3D z uporabo knjižnic Mixed Reality Toolkit v2 (MRTK[6]) in paketa Steam VR[7]. Prva knjižnica je bila potrebna za razvoj predstavitve za HoloLens in s sabo prinaša funkcije za razpoznavanje gest, s katerimi krmilimo programe, funkcijami za razpoznavo in generiranje govora, ki omogoča, da lahko aplikacije krmilimo tudi glasovno in še mnogo koristnih funkcij, ki nam omogočajo hitro in učinkovito delo in hitro izdelavo atraktivnih aplikacij.

Paket Steam VR smo uporabili za izdelavo predstavitve, temelječe na navidezni resničnosti. Čeprav bi tudi to lahko izdelali s pomočjo knjižnice MRTK, pa smo se raje odločili za Steam VR, saj je bolj preizkušena in se ne spreminja tako pogosto.

3 REZULTATI IN RAZPRAVA

Opisana predstavitev raziskovalnih rezultatov je bila prvič uspešno uporabljena na sejmu AERO 2019, ki se vsako leto tradicionalno odvija v Friedrichshafnu

6 https://github.com/Microsoft/MixedRealityToolkit-Unity/releases (zadnji obisk 5. 2. 2020)
7 https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647 (zadnji obisk 5. 2. 2020)

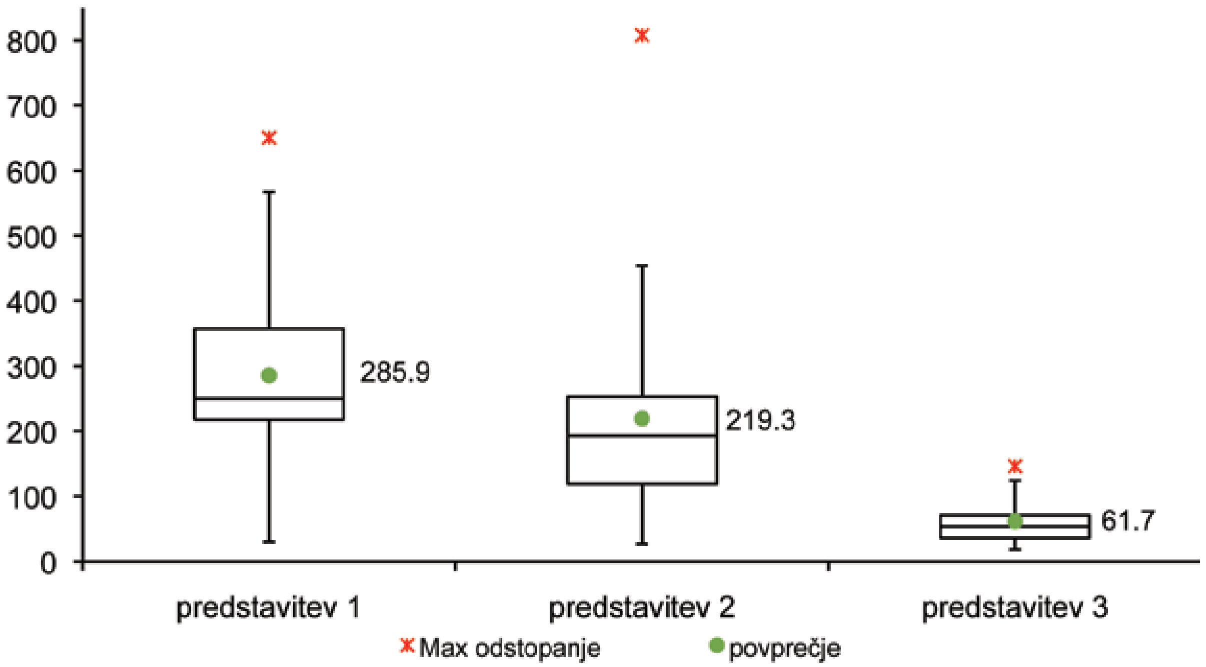
Slika 6: **Notranjost pilotske kabine enega od letal, ki je predmet predstavitve, temelječe na navidezni resničnosti.**

v Nemčiji. Razstavni prostor je bil organiziran tako, da je bilo največ prostora namenjenega predstavitvi z očali HoloLens, kjer se je v povprečju mudilo tudi največ obiskovalcev. Obiskovalce sta privabljali obe fizični maketi letal, kjer je bilo eno od letal dvignjeno v zrak, drugo je bilo postavljeno ob vhodu v razstavni prostor, kot lahko to vidimo tudi na sliki 3a. Pri ogledu notranjosti kabine letala je uporabnik sedel na pilotskem sedežu, kakršen bo vgrajen v letalo, kar je dodatno izboljšalo uporabniško izkušnjo. Glede na to, da je bil glavni cilj predstavitve promocija raziskovalnih rezultatov, nismo mogli kvantitativno ovrednotiti uspešnosti predstavitve z merjenjem časov, ki so jih obiskovalci preživeli na razstavnem prostoru MAHEPA, oziroma konkretno s katero od predstavitev. Po občutku pa lahko povemo, da so bile naše osnovne predpostavke pravilne. Predstavitvi na HoloLensu ali čeladi za navidezno resničnost sta pritegnili pozornost mimoidočih v času uporabe, saj smo vsebino predstavitve, temelječe na navidezni resničnosti prikazovali tudi na televiziji, uporaba HoloLensa pa je bila povezana z različnimi kretnjami uporabnikov, kar je bilo že samo po sebi zanimivo. Med premorom med posameznimi skupinami pa sta pozornost obiskovalcev pritegnili realni maketi letal na razstavnem prostoru. Zanimivo je bilo, da sta bili predstavitvi z uporabo navidezne in obogatene resničnosti enako dobro sprejeti ne glede na spol ali starost obiskovalcev. Kot dobra izbira se je pokazala funkcionalnost premikanja modela letala, saj so se s tem igrali predvsem otroci, prav tako pa je bila otrokom zelo zanimiva predstavitev pilotske kabine letala, kar je staršem omogočilo, da so tudi sami preizkusili katero izmed predstavitev. Poseben test predstavitve so predstavljali razni eksperti, ki jih je zanimal napredek pri razvoju hibridnih pogonov, saj so po ogledu predstavitve pogosto postavljali prisotnim ekspertom, ki so sodelovali pri projektu, konkretna tehnična vprašanja v zvezi z rešitvami, ki zaradi ohranjanja konkurenčne prednosti partnerjev v projektu MAHEPA niso bile v celoti vključene v predstavitev.

Konkretne meritve časov in ovrednotenje uporabniške izkušnje smo izvajali v okviru konference Ceractivity v Pontaderi v Italiji, kjer smo rezultate projekta MAHEPA predstavljali obiskovalcem konference. Za vsakega od obiskovalcev smo merili čas, ki ga je preživel s katero od predstavitev, po končanem ogledu pa smo jih še prosili, če lahko izpolnijo anketo o uporabniški izkušnji. Meritve so potrdile naša opažanja, da so obiskovalci največ časa uporabili pri predstavitvi, krmiljeni s kodami QR in sicer v povprečju 30 % več kot za predstavitev na očalih HoloLens (285,9 sekunde proti 219.3 sekunde), kar lahko vidimo tudi na sliki 7.

4 ZAKLJUČEK

V delu smo opisali naš poskus izdelave virtualne predstavitve novih hibridnih letalskih pogonov, ki se razvijajo v okviru evropskega projekta MAHEPA. Predstavitev je bila uporabljena na letalskem sejmu AERO 2019, kjer smo v praksi preizkusili, ali lahko virtualno predstavitev uspešno postavimo ob bok



Slika 7: **Porazdelitev časov, v sekundah, ki so jih obiskovalci preživeli s predstavitvijo, krmiljeno s kodo QR (predstavitev 1), predstavitvijo, temelječo na obogateni resničnosti (predstavitev 2) in predstavitvijo, temelječo na navidezni resničnosti (predstavitev 3).**

realnim eksponatom. Pri tem smo bili uspešni. Tako smo pokazali, da je lahko virtualna predstavitev cenovno ugodna alternativa realnim predstavitvam v primeru, ko predmeta predstavitve ni še niti v prototipni izvedbi; takrat je to tudi edina možnost za predstavitev novih idej in produktov širši javnosti. Naše ugotovitve sovpadajo s predhodnimi raziskavami na tem področju. Dodamo pa lahko, da je v primeru vzporedne predstavitve virtualnih in realnih eksponatov izredno pomembna realističnost in točnost modelov. Pri samem načrtovanju predstavitev je potrebno upoštevati tudi izsledke raziskav s področja igrifikacije in resnih iger, da uporabnika primerno motiviramo. Naše nadaljnje delo bo usmerjeno v iskanje ključnih faktorjev, ki vplivajo na uspešnost predstavitve, naše izsledke pa bomo testirali v praksi.

ZAHVALA

Avtorji izjavljamo, da je raziskavo finančno podprla Javna agencija za raziskovalno dejavnost Republike Slovenija v okviru programa P2-0041 in projekta BI-BA/19-20-003. Raziskava je prejela tudi sredstva raziskovalnega in inovacijskega programa Evropske unije Obzorje 2020 v skladu s sporazumom št. 723368.

LITERATURA

[1] Grishma Alshi, Mansi Dandiwala, Mikhail Cazi, and Renuka Pawar. 2018. Interactive Augmented Reality-Based System for Traditional Educational Media Using Marker-Derived Contextual Overlays. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 930–935.

[2] Ronald T Azuma. 1997. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments* 6, 4 (1997), 355–385.

[3] Johannes Breuer and Gary Bente. 2010. Why so serious? On the Relation of serious Games and Learning. *Journal for Computer Game Culture* 4, 1 (2010), 7–24.

[4] Chunming Cheng, Li Kang, Saihua Cai, Jiayao Li, and Ruizhi Sun. 2018. Virtual Display and Interactive Experience Platform of Farming Culture Based on Unity3D. *IFAC-PapersOnLine* 51, 17 (2018), 637–642.

[5] Yu-Chih Huang, Sheila J Backman, Kenneth F Backman, and DeWayne Moore. 2013. Exploring user acceptance of 3D virtual worlds in travel and tourism marketing. *Tourism Management* 36, (2013), 490–501.

[6] Filomena Izzo. 2017. Museum customer experience and virtual reality: H. BOSCH exhibition case study. *Modern Economy* 8, 4 (2017), 531–536.

[7] John Keyser, Shankar Krishnan, and Dinesh Manocha. 1999. Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic: I—representations. *Computer Aided Geometric Design* 16, 9 (1999), 841–859.

[8] Sung Lae Kim, Hae Jung Suk, Jeong Hwa Kang, Jun Mo Jung, Teemu H Laine, and Joonas Westlin. 2014. Using Unity 3D to facilitate mobile augmented reality game development. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, IEEE, 21–26.

[9] Xin Min, Wenqiao Zhang, Shouqian Sun, Nan Zhao, Siliang Tang, and Yueting Zhuang. 2019. VPModel: High-fidelity product simulation in a virtual-physical environment. *IEEE transactions on visualization and computer graphics* 25, 11 (2019), 3083–3093.

[10] Jane K Nielsen. 2017. Museum communication and storytelling: articulating understandings within the museum structure. *Museum management and curatorship* 32, 5 (2017), 440–455.

[11] Fuguo Peng and Jing Zhai. 2017. A mobile augmented reality system for exhibition hall based on Vuforia. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, IEEE, 1049–1052.

[12] L Piegl and W Tiller. 1995. The NURBS book springer. *New York* (1995).

[13] Werner Siegfried Ravyse, A Seugnet Blignaut, Verona Leendertz, and Alex Woolner. 2017. Success factors for serious games to enhance learning: a systematic review. *Virtual Reality* 21, 1 (2017), 31–58.

[14] Ganit Richter, Daphne R Raban, and Sheizaf Rafaeli. 2015. Studying gamification: The effect of rewards and incentives on motivation. In *Gamification in education and business*. Springer, 21–46.

[15] Selma Rizvic, Aida Sadzak, Vedad Hulusic, and Amela Karahasanovic. 2013. Interactive digital storytelling in the sarajevo survival tools virtual environment. In *Proceedings of the 28th spring conference on computer graphics*, ACM, 109–116.

[16] Jonathan Steuer. 1992. Defining virtual reality: Dimensions determining telepresence. *Journal of communication* 42, 4 (1992), 73–93.

[17] Rui Sun, Shuming Gao, and Wei Zhao. 2010. An approach to B-rep model simplification based on region suppression. *Computers & Graphics* 34, 5 (2010), 556–564.

[18] Sa Wang, Zhengli Mao, Changhai Zeng, Huili Gong, Shanshan Li, and Beibei Chen. 2010. A new method of virtual reality based on Unity3D. In *2010 18th international conference on Geoinformatics*, IEEE, 1–5.

▪

Simon Kolmanič je docent za področje računalništva na Univerzi v Mariboru, Fakulteti za Elektrotehniko, računalništvo in informatiko. Magistriral je leta 1999 iz računalništva in informatike, leta 2005 pa še doktoriral. Od leta 1996 do leta 2015 je bil član Laboratorija za računalniško grafiko in umetno inteligenco od leta 2015 pa je član Laboratorija za geometrijsko modeliranje in algoritme multimedije. Njegovo raziskovalno področje zaobsega računalniško grafiko in animacijo, algoritme in podatkovne strukture, modeliranje in simulacijo ter računalniško podprto poučevanje. Je avtor večih znanstvenih člankov v priznanih znanstvenih revijah.

▪

Maršenka Marksel je asistentka in raziskovalka na Univerzi v Mariboru, Fakulteti za gradbeništvo, prometno inženirstvo in arhitekturo. Vključena je bila v več mednarodnih raziskovalnih projektov med drugimi: FUTUREMED, HYPSTAIR, GRASPINNO, SULPITER, EGUTS itd. Trenutno aktivno sodeluje v projektih MELINDA, SMACKER, ION, E-CARRAGIES. Je izkušena vodja projektov in trenutno vodi projekt H2020 imenovan MAHEPA. Strokovno je sodelovala pri številnih študijah, kot so Ground infrastructure investments for hybrid-eletric aicraft, Understanding trends & scenarios of freight transport in FUAs, Living Lab approach, Impact analysis of new railway service.

▪

Domen Mongus je izredni profesor na Univerzi v Mariboru in podpredsednik programskega odbora nacionalnega Strateškega razvojno inovacijskega partnerstva v okviru slovenske strategije pametne specializacije (S4) na področju Pametnih mest in skupnosti. Njegovi raziskovalni interesi vključujejo obdelavo podatkov daljinskega zaznavanja, prostorsko-časovno analitiko in geooprostro inteligenco. Do nedavnega je bil tudi član izvršnega odbora krovne Evropske organizacije za geografske informacije (EUROGI) in slovenske sekcije ACM. Za svoje znanstveno in pedagoško delo je bil leta 2015 imenovan za mladega znanstvenika, leta 2018 pa je prejel najprestižnejšo institucionalno akademsko nagrado za izjeme prispevek k znanstvenemu in pedagoškemu ugledu ter odličnosti Univerze v Mariboru.

▪

Borut Žalik je redni profesor za predmetno področje računalništvo na Univerzi v Mariboru. Iz elektrotehnike je diplomiral leta 1985, magistriral je leta 1989 iz računalništva in informatike, leta 1993 pa je pridobil doktorat iz tehniških znanosti. Zaposlen je na Univerzi v Mariboru, Fakulteti za elektrotehniko, računalništvo in informatiko, kjer je bil od 2003 do 2011 prodekan za raziskovalno dejavnost, od 2011 do leta 2019 pa dekan. Vodi Laboratorij za geometrijsko modeliranje in algoritme multimedijev. Njegova raziskovalna področja so obdelava geometrijskih podatkov, stiskanje podatkov in računalniška multimedija. Objavil je več univerzitetnih učbenikov in preko 130 člankov v priznanih znanstvenih revijah. Ima tudi 11 patentov, od tega 3 s popolnim patentnim preizkusom.

Izboljšanje ozaveščenosti na področju informacijske varnosti z uporabo metod igrifikacije

Alenka Brezavšček[1], Maja Minič[2]
[1]Univerza v Mariborui, Fakulteta za organizacijske vede, Kidričeva cesta 55 A, 4000 Kranj
[2]Ministrstvo za obrambo Republike Slovenije, Vojkova cesta 55, 1000 Ljubljana
alenka.brezavscek@um.si

Izvleček
V prispevku obravnavamo uporabo metod igrifikacije (tudi poigritev) za potrebe izobraževanja. Na kratko smo opisali elemente igrifikacije, predstavili ključne značilnosti teh metod ter zaznane pozitivne učinke na samo učinkovitost izobraževanja. Na podlagi izsledkov iz literature ter izkušenj iz prakse smo prikazali možnosti in načine uporabe teh metod pri ozaveščanju uporabnikov na področju informacijske (kibernetske) varnosti. Tako izkušnje iz literature kakor tudi empirične izkušnje dokazujejo, da je uporaba igrifikacije pri informacijsko varnostnem ozaveščanju povezana s pozitivno uporabniško izkušnjo, naklonjenostjo, kakor tudi z dejanskim dvigom ozaveščenosti.
Ključne besede: igrifikacija, informacijska varnost, kibernetska varnost, ozaveščanje, uporabnik.

Abstract
TThe paper deals with the use of gamification methods for educational purposes. We briefly described the elements of gamification, outlined the main features of these methods and the acknowledged positive effects on the effectiveness of education. Based on the findings from the literature and practical experience, we presented the possibilities and ways in which these methods can be used in awareness-raising activities in the field of information (cyber) security. Indeed, both literary and empirical experiences show that the use of gamification in information security awareness activities leads to a positive user experience, inclination and actual increase in awareness levels.
Keywords: Gamification, information security, cyber security, awareness, user.

1 UVOD

Zagotavljanje informacijske varnosti je danes ključnega pomena tako z vidika organizacij kot tudi posameznikov. Temelj zagotavljanja informacijske varnosti sicer res predstavlja implementacija ustreznih tehničnih rešitev, vendar empirične izkušnje dokazujejo, da ne glede na višino investicij v tehnične zaščite, doseženi nivo informacijske varnosti ne bo zadovoljiv, v kolikor v procese zagotavljanja informacijske varnosti ne vključimo najšibkejšega člena –človeka (Weishäupl et al., 2018). Tudi organizacija NIST (National Institute of Standards and Technology) poudarja pomen postopnega oblikovanja kompetenc na področju informacijske varnosti s ciljem spremembe vedenja v smeri varnostno ozaveščenega delovanja (NIST, 2003). Eno izmed ključnih aktivnostih pri tem zagotovo predstavlja kontinuirano in sistematično ozaveščanje zaposlenih. Če je program ozaveščanja in izobraževanja vsebinsko premišljen in učinkovito izveden, lahko bistveno pripomore k zmanjšanju verjetnosti za realizacijo varnostnega incidenta v organizaciji.

Številne sodobne študije razkrivajo, da lahko k učinkovitosti izobraževanja v splošnem precej pripomore uporaba metod igrifikacije (tudi poigritev; glej npr. Giang, 2013; Kapp, 2012). Rezultati raziskav dokazujejo mnoge pozitivne učinke uvedbe tovrstnih metod v izobraževanje. Nekateri avtorji med drugim navajajo, da le-te povečujejo sposobnost osvajanja novih znanj tudi za 40% (povzeto po Gabe Zichermann, citirano v Giang, 2013).

V prispevku se bomo osredotočili na uporabo metod igrifikacije pri izobraževanju in ozaveščanju uporabnikov na področju informacijske varnosti. Prispevek je organiziran na naslednji način: Najprej bomo podali nekaj osnov same igrifikacije, izpostavili ključne značilnosti ter podali definicije temeljnih pojmov. Osrednji del prispevka bo namenjen uporabi metod igrifikacije za potrebe ozaveščanja zaposlenih na področju informacijske (tudi kibernetske) varnosti. Na podlagi izsledkov iz literature kakor tudi empiričnih spoznanj bomo prikazali možnosti in načine uporabe tovrstnih metod pri reševanju vsakodnevnih izzivov, kot npr.: Kako uspešno usposabljati uporabnike, ki jih vsebina pravzaprav ne zanima? Kako preseči način delovanja – vem, vendar ne delam tako? Po naših izkušnjah se s tovrstnimi vprašanji vsakodnevno soočajo skrbniki za informacijsko varnost v sleherni organizaciji.

2 TEORETIČNE OSNOVE S PODROČJA IGRIFIKACIJE

Osnovni koncepti igrifikacije temeljijo na lastnostih iger, ki so ena izmed najstarejših oblik socialne interakcije posameznika (Voorhies, 2012). Dandanes se z igrami srečujemo zelo pogosto tako v fizičnem kot virtualnem okolju. Čeprav je primarni namen iger sicer zabava, mnoge raziskave dokazujejo, da predstavljajo metode in principi, na katerih temeljijo igre, pomembno sodobno orodje, ki je uporabno na marsikaterem področju, od razvoja novih produktov, prilagajanja delovnih mest, marketinga ali oblikovanja življenjskega sloga (glej npr. Kim, 2013; Kim, 2013a). Tako se v zadnjem obdobju tako v teoriji kot v praksi vse bolj uveljavlja področje, znano pod imenom *igrifikacija* (ang. gamification).

Sam pojem *igrifikacija* lahko definiramo kot uporabo lastnosti iger in igralne mehanike v neigralnem okolju (Kapp, 2012). Osnovni namen metod igrifikacije je motiviranje udeležencev za njihovo aktivno vključevanje v določen proces, kar se izkaže kot uporabno na marsikaterem področju (Kim, 2013a). Zaradi široke aplikativnosti in dokazanih pozitivnih učinkov je problematika igrifikacije v zadnjem obdobju bogato zastopana tako v strokovni kot v znanstveni literaturi. Sistematičen pregled najdemo npr. v Hamari et al. (2014) ali Koivisto & Hamari (2019). V nadaljevanju bomo predstavili nekaj temeljnih pojmov in konceptov igrifikacije.

Igralno mehaniko (ang. game mechanics) predstavljajo elementi in pravila, ki narekujejo, kako igra poteka. Po svoj naravi so lahko taka pravila različna, skupno vsem pa je, da v sami igri igrajo pomembno vlogo. Igro lahko naredijo zahtevno, zabavno, tekmovalno, potek igre pa vztrajno usmerjajo k njenemu glavnemu cilju (Seppo, n.d.). Primeri elementov igralne mehanike so: točke (ang. points), značke (ang. badges), uvrstitev na lestvici (ang. leaderboard), nivoji (ang. levels), ipd. Primarni namen elementov igralne mehanike je zadovoljevanje določenih želja igralca. Povezavo med elementi igralne mehanike in igralčevimi željami ponazarja tabela 1.

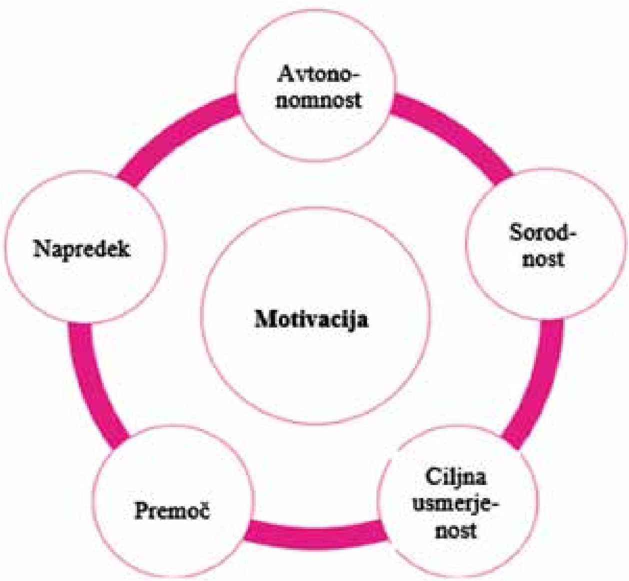
Podoben koncept kot igralna mehanika predstavlja *igralna dinamika* (ang. game dynamics). Razlika je v tem, da ima igralna dinamika nekoliko bolj motivacijski vzgib. S kombinacijo igralne dinamike in igralne mehanike skušajo avtorji igre pri igralcu vzbuditi njegove naravne potrebe po dokazovanju, kot npr. tekmovalnost, junaštvo, samopotrjevanje ali altruizem. Osnovni cilj je pri igralcu vzbuditi zanimanje in motivacijo za neko konkretno dejanje (Seppo, n.d.).

		Igralčeva potreba					
		Nagrada	Status	Dosežek	Samopotrditev	Tekma	Altruizem
Igralna mehanika	Točke	●	◐	◐		◐	◐
	Nivoji		●	◐		◐	
	Izzivi	◐	◐	●	◐	◐	◐
	Virtualne dobrine	◐	◐	◐	●	◐	
	Lestvica		◐	◐		●	◐
	Dobrodelnost		◐	◐		◐	

Tabela 1: **Povezava med elementi igralne mehanike in željami igralca (Kim, 2013a).**

Najvišji cilj je igralca pripeljati v t.i. *stanje toka* (ang. flow state), ki predstavlja stanje duha, ko je igralec visoko motiviran in zelo osredotočen na doseganje trenutnega cilja tako, da je sposoben iz samega procesa eliminirati vse ostale zunanje dejavnike (Seppo, n.d.). Namen je vzbuditi t.i. *notranjo motivacijo* (ang. intrinsic motivation), ki predstavlja človekovo notranjo željo nekaj narediti zaradi naloge same in ne zaradi zunanje nagrade za opravljeno nalogo. Notranja motivacija predstavlja temeljni pojem znane motivacijske teorije samodoločenosti (ang. selfdetermination theory) (Ryan & Deci, 2000).

Igre in igrifikacija vključujejo različne metode in tehnike, ki so se izkazale za učinkovite pri spodbujanju ključnih elementov notranje motivacije igralca (glej sliko 1). Zanimivo študijo podaja Karimi (2017), kjer analizira metode igrifikacije z vidika motivacijske teorije.



Slika 1: **Ključni elementi motivacije, ki jih metode igrifikacije spodbujajo (Seppo, n.d.).**

Strokovnjaki s področja igrifikacije se precej ukvarjajo tudi s proučevanjem psiholoških lastnosti samih igralcev, pri čemer igralce razvrščajo v različne kategorije. Znan je Bartlejev model (citirano v Kim, 2013), ki v osnovi opredeljuje štiri različne tipe igralcev (glej sliko 2): ubijalec (ang. killer), dosežkar (achiever), socialnež (ang. socializer) in raziskovalec (ang. explorer).

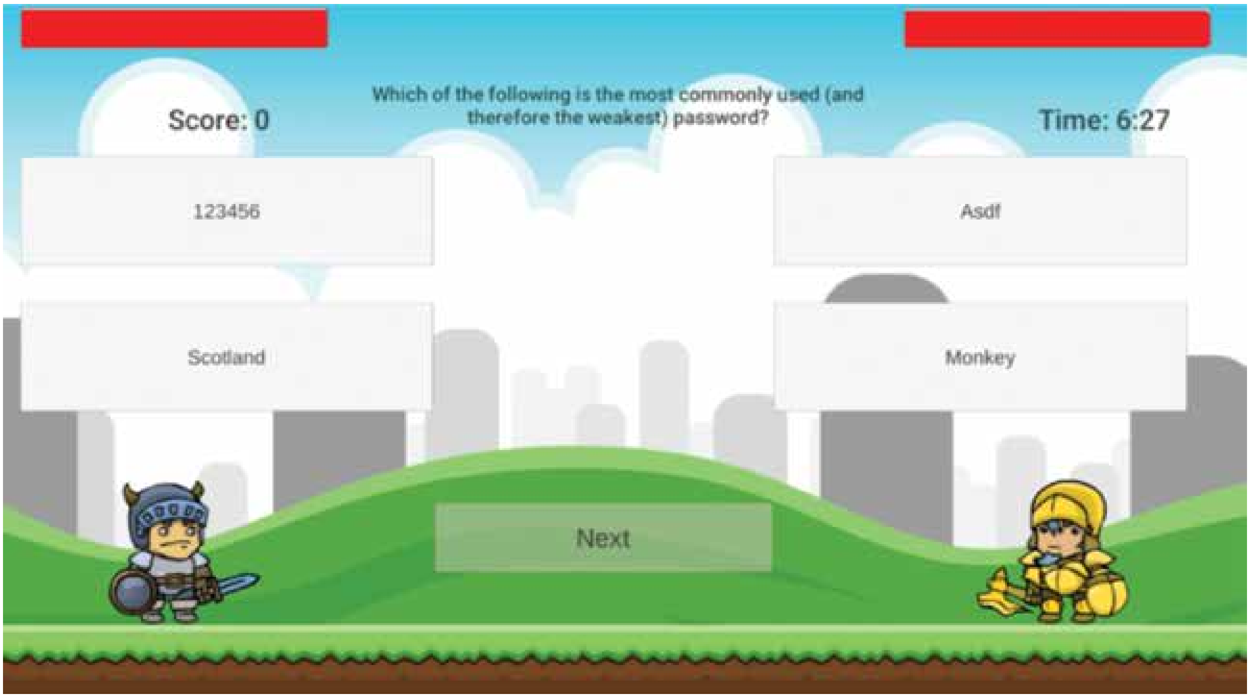
Slika 2: **Bartlejev model kategorizacije igralcev (prirejeno po Kumar et al., 2019).**

Poleg številnih aplikacij na različna področja so se metode igrifikacije izkazale za zelo učinkovite ravno na področju izobraževanja (Majuri et al, 2018). Cilj igrifikacije je narediti proces učenja zabaven z namenom vzbuditi interes za obravnavano tematiko. Tako tovrstne metode pozitivno vplivajo na motivacijo pri učenju ter posledično na učinkovitost samega učenja in učne rezultate (Damsa & Fromann, 2016). Poleg tega metode igrifikacije pomagajo razumeti povezave med abstraktnimi koncepti in vsakdanjim življenjem (Seppo, n.d.), kar »učencem« pomaga pri boljšem razumevanju proučevane tematike, kar se zopet kaže v boljših učnih rezultatih.

3 UPORABA IGRIFIKACIJE PRI OZAVEŠČANJU ZA IFORMACIJSKO VARNOST

3.1 Izkušnje iz literature

Da je uporaba metod igrifikacije za potrebe informacijsko varnostnega ozaveščanja aktualna tema, dokazuje število z obravnavano tematiko povezanih prispevkov, ki je v zadnjih letih precej naraslo. Avtorji poudarjajo pomen ozaveščanja na področju informacijske (kibernetske) varnosti, kakor tudi izpostavljajo prednosti, ki jih prinašajo metode igrifikacije v primerjavi s klasičnimi izobraževalnimi metodami. Nekateri avtorji so mnenja, da na področju informacijsko varnostnega ozaveščanja klasični pristopi izobraževanja ne zadoščajo več. Sklicujejo se na psihološke raziskave, ki priporočajo vpeljavo, sodobnih, sistemskih pristopov, kakršnega nudi tudi učenje, podprto z igrifikacijo (Scholl, 2018). Splošni vtis po pregledu literature je, da so se metode igrifikacije na področju



Slika 2: **Bartlejev model kategorizacije igralcev (prirejeno po Kumar et al., 2019).**

informacijsko varnostnega ozaveščanja izkazale kot uporabne, med uporabniki priljubljene, kakor tudi učinkovite, saj dejansko lahko pripomorejo k višjemu nivoju ozaveščenosti (Rieff, 2018).

Nekateri avtorji podajajo rezultate konkretnih empiričnih študij uvajanja metod igrifikacije pri informacijsko varnostnem ozaveščanju. Scholefield in Shepard (2019) na primer delita svoje izkušnje pri ozaveščanju uporabnikov za varno uporabo gesel. Le-to se izvaja z namensko Android aplikacijo (slika 3), ki temelji na kvizu z igranjem vlog. Rezultati dokazujejo pozitivno uporabniško izkušnjo, dejanski dvig ozaveščenosti med uporabniki, kakor tudi naklonjenost uvajanju metod igrifikacije. Podobnega mnenja so tudi avtorji raziskave Gjertsen et al. (2017), ki so uporabniško izkušnjo proučevali s pomočjo namenske prototipne interaktivne aplikacije, ki so jo testirali med zaposlenimi v dveh različnih organizacijah.

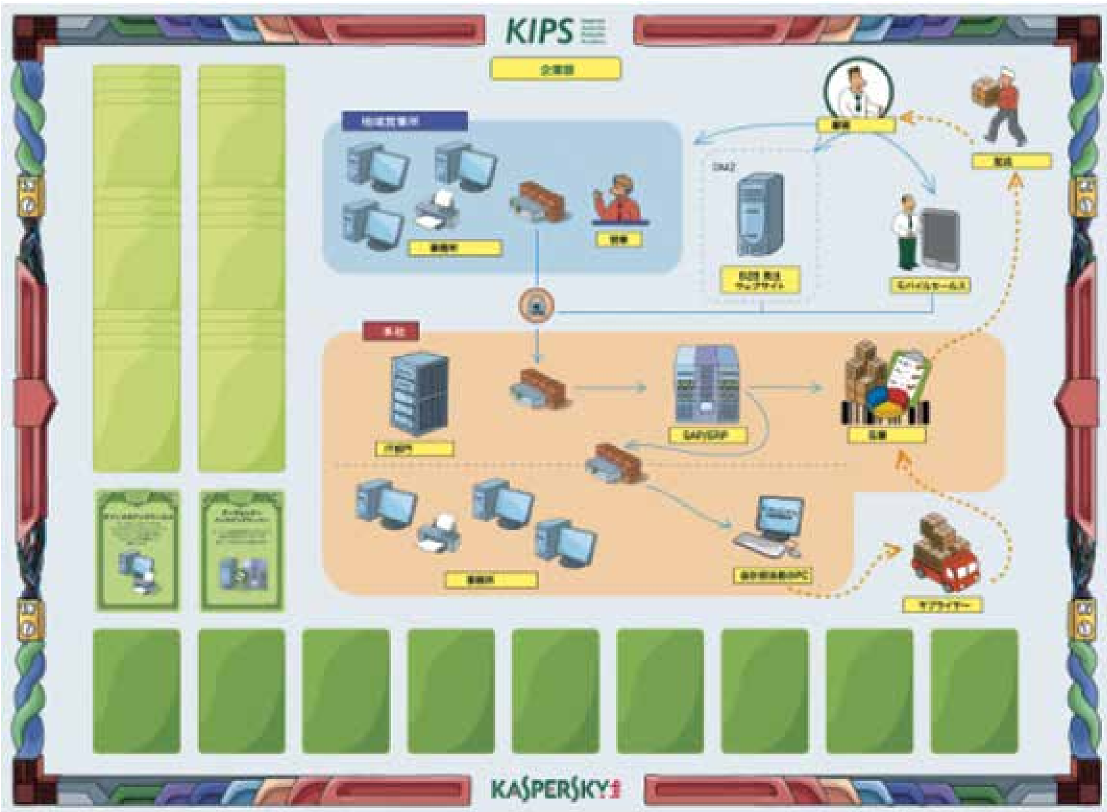
Nekateri avtorji proučujejo učinek t.i. *resnih iger*[1] (ang. serious games) in želijo raziskati, ali so tovrstne igre lahko učinkovito orodje pri ozaveščanju. Hendrix et al. (2016) in Alotaibi et al. (2016) podajajo izčrpen pregled literature in samih iger, uporabnih za namene informacijsko (kibernetsko) varnostnega ozaveščanja. Avtorji ugotavljajo, da so prvi učinki uporabe dejansko povečini pozitivni, vendar opozarjajo, da je za relevantnost rezultatov potrebno daljše obdobje opazovanja. Opozarjajo tudi na vrzel na tržišču, saj je večina tovrstnih izdelkov namenjenih povprečnim uporabnikom (splošni javnosti), pogrešajo pa igre, namenjene IT strokovnjakom.

To vrzel najbrž vsaj deloma zapolnjuje produkt Kaspersky Interactive Protection Simulation (KIPS)[2], ki omogoča skrbnikom informacijske varnosti in odločevalcem v organizacijah, da se preko simulacij različnih poslovnih okolij spopadejo z vrsto nepričakovanih kibernetskih groženj, jih skušajo obvladovati, hkrati pa povečevati dobiček in ohranjati zaupanje. Primer simulacije poslovnega okolja v sistemu KIPS prikazuje slika 4.

Rezultati eksperimentalne študije Yonemura et al. (2018) dokazujejo, da sočasno igranje več uporabnikov (ang. multiple playing) pozitivno vpliva tako na izobraževalne učinke kot na prenos veščin med uporabniki.

Antonaci et al. (2017) in Fuhrman et al. (2016) se

1 Resne igre so igre, katerih primarni namen je širši kot zgolj zabava.
2 https://media.kaspersky.com/en/business-security/enterprise/KL_SA_KIPS_overview_A4_Eng_web.pdf



Slika 4: **Primer simulacije poslovnega okolja v sistemu Kaspersky Interactive Protection Simulation (Yonemura Set al., 2018).**

ukvarjajo z uporabo igrifikacje za potrebe informacijsko varnostnega ozaveščanja mladostnikov (najstnikov in študentov). Poudarjajo, da se mladi pogosto ne zavedajo tveganj, povezanih z deljenjem zasebnih informacij v internet (npr. preko socialnih omrežij). Hkrati pa poudarjajo pomen ozaveščenosti mladih kot bodočih zaposlenih tudi z vidika organizacij in poslovnih okolij. Avtorji raziskave Antonaci et al. (2017) menijo, da igrajo ključno vlogo pri ozaveščenosti mladih njihovi učitelji, zato so razvili interaktivni spletni portal za izobraževanje učiteljev. Kot atraktiven, uporaben in učinkovit pristop za učenje mladih pa avtorji Li & Kulkarni (2016) navajajo tudi dogodke CTF (Capture-the-Flag).

3.2 Izkušnje iz prakse

Na podlagali dolgoletnih izkušenj na področju informacijsko-varnostnega ozaveščanja uporabnikov ugotavljamo naslednje: če želimo nivo ozaveščenosti uporabnikov dvigniti na želeno raven, to raven ohranjati in nenazadnje nadgrajevati, je potrebno izobraževanja izvajati sistematično in kontinuirano. Menimo, da je v večjih sistemih z velikim številom uporabnikov dobrodošel pripomoček za doseganje želenega cilja uporaba e-izobraževanja. Pri tem pa se postavlja vprašanje, na kakšen način podati izobraževalna gradiva, da bodo le-ta ne samo pritegnila uporabnika, temveč tudi zadržala njegovo pozornost skozi celoten proces učenja. Na podlagi dosedanje prakse ugotavljamo, da je eden od ključnih faktorjev, ki nam pri tem nedvomno pomaga, interaktivnost samih izobraževalnih gradiv, kar pa lahko učinkovito dosežemo z vključevanjem elementov igrifikacije.

Po našem mnenju se na področju informacijsko-varnostnega ozaveščanja uporabnikov v slovenskih poslovnih okoljih igrifikacija v pravem pomenu besede (z vsemi, v poglavju 2 predstavljenimi, elementi) v danem trenutku le redko uporablja. Najpogosteje manjka ravno element tekmovanja, vendar pa je viden porast implementacije simulacijskih iger. V takih primerih je uporabnik postavljen pred nek izziv, vsak naslednji korak pa je posledica njegove odločitve. Na tak način se posameznik sreča s problemom (npr. napad socialnega inženirja) v virtualnem okolju. Ne

Slika 5: **Šola internetne samoobrambe kot dober primer uporabe igrifikacije pri varnostnem ozaveščanju3.**

glede na to, ali je njegova posamezna izbira pravilna ali ne, se preko virtualnih izobraževalnih elementov nauči pravilnega odziva, ki ga v prihodnje lahko uporabi v realnem okolju. Uporaba igrifikacijskih metod tako preseže samo (pogosto suhoparno) teoretično učenje in uporabnika postavi bodisi v vlogo žrtve bodisi napadalca. Ključnega pomena je namreč vzpostavitev povezave med varnostnimi napadi in njihovimi potencialnimi vplivi na ljudi in podjetja. Z uporabo igrifikacije so uporabniki ves čas aktivno vključeni v usposabljanje, kar po mnenju nekaterih strokovnjakov poveča vztrajnost za učenje tudi za 75 odstotkov (Sedova, 2018). Z namenom ohranjanja pozornosti in koncentracije uporabnika je po naših izkušnjah priporočljivo tematiko razdeliti na krajše segmente, pri čemer priporočamo, da posamični segment ni daljši od 20 minut.

Odličen primer uporabe igrifikacije za ozaveščanje na področju informacijske varnosti v Sloveniji je primer izobraževanja za otroke, imenovano Šola internetne samoobrambe[3] (slika 5), pri razvoju katerega smo tudi aktivno sodelovali. Gre za interaktivni spletni portal, ki vključuje večino elementov igrifikacije, od simulacijskih iger, socialne interakcije, pa vse do pridobivanja točk (v tem primeru so otroci ob pridobivanju znanj pridobivali virtualne karateistične pasove). Rezultati omenjenega izobraževanja so bili spodbudni, saj so otroci na vseh področjih dosegli izboljšanje rezultatov pri preverjanju znanja na področju varne rabe interneta.

4 SKLEP

Metode igrifikacije kot sodobne metode izobraževanja prinašajo številne pozitivne učinke na samo učinkovitost izobraževanja in na učne rezultate. S pričujočim prispevkom smo skušali prikazati, kako lahko tovrstne metode izkoristimo tudi na področju informacijsko varnostnega ozaveščanja uporabnikov. Ker slednje predstavlja dokaj nov in v slovenskem prostoru še neraziskan koncept, menimo, da so izsledki prispevka koristni za vso strokovno javnost, predvsem pa za skrbnike informacijske varnosti v or-

[3] Šola internetne samoobrambe. https://otroci.e-ucenje.com/show.aspx?xid=WBT:X:Start.

ganizacijah, ki si vsakodnevno prizadevajo, da bi njihovi uporabniki ne le pridobili, temveč tudi ohranjali in nadgrajevali zanimanje in znanje na področju informacijske varnosti ter posledično prispevali k vse višjemu nivoju informacijske varnosti v organizaciji.

LITERATURA

[1] Alotaibi, F., Furnell, S., Stengel, I. & Papadaki, M. (2016). A Review of Using Gaming Technology for Cyber-Security Awareness. *International Journal for Information Security Research*, 6(2), 660-666. Pridobljeno iz: https://infonomics--society.org/wp-content/uploads/ijisr/published-papers/volume-6-2016/A-Review-of-Using-Gaming-Technology-for--Cyber-Security-Awareness.pdf

[2] Antonaci, A., Klemke, R., Stracke, C. & Specht, M. (2017, April). Gamification in MOOCs to enhance users' goal achievement. In *Proceedings of 2017 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1654-1662, IEEE Xplore. Pridobljeno iz: https://ieeexplore.ieee.org/document/7943070

[3] Damsa, A. & Fromann, R. (2016). Gamification and Gameful Approaches in Education, Business, and IT. *Informatika*, 8(1), 28-33. Pridobljeno iz: https://www.semanticscholar.org/paper/GAMIFICATION-AND-GAMEFUL-APPROACHES-IN--EDUCATION-%2C-Damsa-Fromann/fa6d5ab15a43dbd-706c5d44dffa6b4e47c8bc08d

[4] Fuhrmann, F., Scholl, M., Ehrlich, E.P., Edich, D., Leiner, B. & Scholl, L. (2016, November). Raising Awareness for Information Security in a Playful Way. In *Proceedings of the London International Conference on Education (LICE)*, pp. 190-191, London: Infonomics Society. Pridobljeno iz: https://publister.bib.th-wildau.de/publister/public/publication/2119

[5] Giang, V. (2013). Gamification Techniques Increase Your Employees' Ability To Learn By 40%, *Business Insider.* Pridobljeno iz: https://www.businessinsider.com/gamification-techniques--increase-your-employees-ability-to-learn-by-40-2013-9

[6] Gjertsen, E., Gjære, E. A., & Bartnes, M. & Flores, W. (2017, February). Gamification of Information Security Awareness and Training. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pp. 59-70. SciTePress. Pridobljeno iz: http://www.scitepress.org/DigitalLibrary/Link.aspx?-doi=10.5220/0006128500590070

[7] Hamari, J., Koivisto, J. & Sarsa, H. (2014, January). Does Gamification Work? — A Literature Review of Empirical Studies on Gamification. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, pp. 3025-3034, IEEE. Pridobljeno iz: https://ieeexplore.ieee.org/document/6758978

[8] Hendrix, M., Al-Sherbaz, A. & Bloom, V. (2016). Game Based Cyber Security Training: are Serious Games suitable for cyber security training?. *International Journal of Serious Games*, 3(1), 53-61. Pridobljeno iz: https://www.semanticscholar.org/paper/Game-Based-Cyber-Security-Training%3A-are-Serious-for-Hendrix-Al-Sherbaz/b5a5182a975504051ee241cff1e-02caec04a1082

[9] Karimi, K. (2017). Gamification from the viewpoint of motivational theory. *Italian Journal of Science & Engineering, 2017*, 1, 34-42. Pridobljeno iz: https://ijournalse.org/index.php/ESJ/article/view/19/4

[10] Kapp, K. M. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*, Pfeiffer, San Francisco, 2012.

[11] Kim, S. (2013). Fundamental Strategic Approach for Gamification : How to Start a Gamification in Your Organization. *International Journal of Digital Content Technology and its Applications*, 7(12). 48-55. Pridobljeno iz: https://www.researchgate.net/publication/310465839_Fundamental_Strategic_Approach_for_Gamification_How_to_Start_a_Gamification_in_Your_Organization

[12] Kim, S. (2013a). Recent Advances in Gamification Application. *Advances in information Sciences and Service Sciences,* 5(13), 93-99. Pridobljeno iz: https://www.researchgate.net/publication/310465687_Recent_Advances_in_Gamification_Application

[13] Koivisto, J. & Hamari, J. (2019). The rise of motivational information systems: A review of gamification research. *International Journal of Information Management*, 45, 191-210. Pridobljeno iz: https://www.sciencedirect.com/science/article/pii/S0268401217305169

[14] Kumar, J., Herger, M. & Dam, R. F. (2019). Bartle's Player Types for Gamification. *The Interaction Design Foundation.* Pridobljeno iz: https://www.interaction-design.org/literature/article/bartle-s-player-types-for-gamification.

[15] Li, C. & Kulkarni, R. (2016, June). Cybersecurity Education through Gamification – the CTF Approach. In *Proceedings of 2016 ASEE Annual Conference & Exposition*, American Society for Engineering Education. Pridobljeno iz: https://www.asee.org/public/conferences/64/papers/16728/view

[16] Majuri, J., Koivisto, J., & Hamari, J. (2018, May). Gamification of education and learning: A review of empirical literature. In *Proceedings of the 2nd International GamiFIN conference*, pp. 11-19, At Pori, Finland. Pridobljeno iz: http://ceur-ws.org/Vol-2186/paper2.pdf

[17] NIST (2003). *NIST. SP 800-50: Building an Information Technology Security Awareness and Training Program*, National Institute of Standards and Technology. Pridobljeno iz: https://csrc.nist.gov/publications/detail/sp/800-50/final

[18] Rieff, I. (2018). *Systematically Applying Gamification to Cyber Security Awareness Trainings: A framework and case study approach*, Master thesis, Faculty of TPM, Delft University of Technology, 2018. Pridobljeno iz: https://repository.tudelft.nl/islandora/object/uuid:bf832ca0-91d9-4be1-9a25--fe284c23d115/datastream/OBJ1/download

[19] Ryan. R. M. & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1), 68–78. Pridobljeno iz: https://selfdeterminationtheory.org/SDT/documents/2000_RyanDeci_SDT.pdf

[20] Scholefield, S. & Shepherd, L. (2019). Gamification techniques for raising cyber security awareness. In A. Moallem (Ed.), *HCI for cybersecurity,privacy and trust: HCII 2019*, 191-201. (Lecture Notes in Computer Science, 2019, Vol. 11594. Springer. Pridobljeno iz: https://www.researchgate.net/publication/331916014_Gamification_Techniques_for_Raising_Cyber_Security_Awareness

[21] Scholl, M. (2018). Play the Game! Analogue Gamification for Raising Information Security Awareness. *Systemics, Cybernetics and Informatics*, 16(3), 32-35. Pridobljeno iz: https://www.semanticscholar.org/paper/Play-the-Game-!-Analogue-Gamification-for-Raising-Scholl/6e23340c5ce46cfd1ae-36e6f2bdddb84f02a3991

[22] Sedova, M. (2018). How Gamification Is Changing the Way Employees Engage in Security Training, *Elevate Security*. Pridobljeno iz: https://elevatesecurity.com/blog/how-gamification-is-changing-the-way-employees-engage-in-security--training/

[23] Seppo (n.d.). *Motivation Through Gamification*, Seppo.IO. Pridobljeno iz: https://seppo.io/site/assets/files/2292/motivation-through-gamification-corporate.pdf
[24] Voorhies, B. (2012). Games ancient people played. *Archaeology*, 65(3), 48-51. Pridobljeno iz: https://archive.archaeology.org/1205/features/tlacuachero_chiapas_patolli_hualapai.html
[25] Weishäupl, E., Yasasin, E. & Schryen, G. (2018). Information security investments: An exploratory multiple case study on decision-making, evaluation and learning. *Computers & Security*, 77, 807-823. Pridobljeno iz: https://www.sciencedirect.com/science/article/pii/S0167404818300555
[26] Yonemura, K. et al. (2018). Effect of security education using KIPS and gamification theory at KOSEN. In *Proceedings of 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Penang, pp. 255-258. Pridobljeno iz: https://ieeexplore.ieee.org/document/8405480

▪

Alenka Brezavšček je izredni profesor na Fakulteti za organizacijske vede Univerze v Mariboru. Njeno habilitacijsko področje so kvantitativne metode v organizacijskih vedah. Ukvarja se z raziskavami na področju stohastičnih procesov, zanesljivosti in razpoložljivosti tehničnih sistemov ter varnosti informacijskih sistemov.

▪

Maja Minič je zaposlena na Ministrstvu za obrambo Republike Slovenije. Ima vrsto let izkušenj na področju analize in razvoja informacijskih sistemov, implementacije in razvoja e-izobraževanja, ter ozaveščanja na področju informacijske varnosti.

Spomini na prve korake v robotiko v Sloveniji

Saša Divjak
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, SI-1000 Ljubljana
sasa.divjak@fri.uni-lj.si

Spomini segajo v sedemdeseta leta prejšnjega stoletja in na ustvarjalno delo na Inštitutu Jožef Stefan (IJS). Naslednje vrstice odražajo le enega od pogledov, saj nas je bilo ustvarjalcev kar nekaj, večinoma pa je naše delo preveval veliko razvojno raziskovalno navdušenje. Prav bi bilo, da te vrstice dopolnijo še drugi »igralci«, saj so moji spomini le delni, pa tudi po tolikšnem času že nekoliko zamegljeni.

Z današnjega stališča so stvari morda videti preproste, a takrat smo orali ledino in imeli za današnje pojme na voljo res skromno tehnologijo. Ker so bili tudi prvi koraki v robotiko s tem pogojeni, je prav, da o teh romantičnih časih povem kaj več.

Osnova so tedaj bili mikroračunalniki z mikroprocesorji 8008, kasneje pa seveda tudi drugimi. Prvi tak razvojni mikroračunalnik smo kupili po komponentah in prvi program zanj ročno, dejansko binarno vpisali s stikali kar v strojni kodi in upali, da se pri tem ne bi kaj zmotili, saj je bil postopek zamuden. Kasneje pa smo prešli na lasten razvoj mikroračunalnikov. V našem odseku za avtomatiko, biokibernetiko in robotiko je bil temelj tako imenovani mikroračunalnik Darta 80, ki ga je leta 1980 modularno zasnoval Jurij Tasič. Zanimivost je še, da se je samo na IJS v različnih odsekih pojavilo več podobnih mikroračunalniških konceptov. To so bili časi Jugoslavije in hladne vojne med blokoma. Tako nismo smeli (z mikroračunalniki) avtomatiziranih sistemov, ki so imeli vgrajene ameriške mikroprocesorje, kljub siceršnjemu lastnemu razvoju, posredovati v dežele vzhodnega bloka. Mikroprocesorji so bili praktično prešteti. A o tem kasneje.

V sedemdesetih letih prejšnjega stoletja je bilo programiranje mikroračunalniških sistemov sprva zamudno, saj takrat nismo imeli zunanjih magnetnih medijev (diskov ali vsaj kaset). In urejevalniki besedila, torej izvorne programske kode, so bili tedaj zelo primitivni, enovrstični, kar je za današnje čase nepojmljivo. Samo za zgled, kako je potekal postopek od odkritja programerske napake do njenega popravka. Naslednjih nekaj stavkov lahko razume le računalnikar tiste generacije. Koraki so bili naslednji:

Slika 1: **Razvoj programske opreme je potekal s pomočjo teleprinerja**

1. V računalnik smo preko luknjanega traku prebrali urejevalnik besedila
2. Prebrali so nato izvorno programsko kodo z ugotovljeno napako.
3. Popravili smo napako v izvorni kodi
4. Sledilo je luknjanje popravljene izvorne kode na nov trak
5. Nato sledi branje tako imenovanega assemblerja (zbirnika)
6. In nato branje popravljene izvorne kode našega programa
7. Ahh, ponovno branje popravljene izvorne kode (tako je pač zbirnik deloval, v dveh pasovih)

8. In končno luknjanje tako pridobljene binarne (strojne) kode na luknjan trak
9. In branje traku z binarno kodo nazaj v ciljni računalnik. Ob takratni hitrosti teleprinterjev (teletype) je ves postopek trajal kar nekaj minut. Danes za to potrebujemo drobec sekunde. Ni čuda, če smo se temu hoteli izogniti. In ob odkritju manjših programskih hroščev smo pogosto raje napakico popravili kar na strojnem nivoju (če se je dalo) in »spekli« nov EPROM. To je vrsta bralnega pomnilnika, ki omogoča večkraten zapis podatkov. Pomnilnike pobrišemo z UV-svetlobo, ki preko odprtine na čipu toplotno ogreje pomnilne celice, ki zato pozabijo staro vsebino. Sledi programiranje z enakim programatorjem.

Posledica take slabe navade je lahko bila, da smo na koncu sicer imeli pravilno delujoč program, njegova izvorna koda pa temu ni več ustrezala. Za današnje razmere hudo napačen pristop.

Programiranje je v tistih časih tako praviloma potekalo na ravni zbirnega jezika. Razvoj v svetu je seveda tekel naprej in v drugi polovici sedemdesetih let smo dobili nekaj računalnikov PDP 11 znamke Digital, ki pa so že bili opremljeni z diskom. Za tak računalnik sem razvil tako imenovani križni zbirnik (crossassembler). Ta je omogočal bolj udobno in hitrejše razvijanje in popravljanje programov za naše avtomatizirane sisteme. Še vedno pa je programiranje potekalo na ravni zbirnega jezika, rezultat pa je še vedno bila binarna koda na luknjanem traku. A takrat so bili luknjači in bralci parirnega traku že precej hitrejši.

Za hip se tu ustavimo in se povrnimo na problematiko prepovedi izvoza računalniških avtomatizacij z ameriškimi čipi na vzhod. V odseku smo dobili projekt z Iskro, v okviru katerega naj bi razvili računalniško krmiljen spektrometer za znano firmo Carl Zeiss, ki pa je bilo v takratni Vzhodni Nemčiji. Kaj storiti?. Povrnili smo se h koreninam pojava mikroprocesorjev. Prvi mikroprocesor je bil 4 bitni Intel 4004, ki je v bistvu izhajal iz kombinacije CPE in pomnilnika. In prav tako pot smo ubrali z originalno kombinacijo klasičnega kalkulatorskega čipa in pomnilnika ROM. Tako smo dobili »svoj lastni« mikroračunalnik, ki ni bil podvržen embargu. Dopolnili smo ga s svojim naborom ukazov in s posebej zanj razvitim zbirnim jezikom in zbirnikom. Za aparaturno zasnovo je poskrbel Jurij Tasič, programski del pa je bil moj. Seveda so bili v razvoj vključeni tudi drugi sodelavci.

Spomnim se, kako smo s prototipom z avtom odšli v Vzhodno Nemčijo, za takratno železno zaveso. In v Carlu Zeissu so nam najprej pobrali potne liste in smo jih dobili nazaj šele po uspešni predaji prototipa. K sreči brez kakšnih zapletov. Spektrometer SPEKOL je bil nato uspešnica Iskre Horjul in je tudi dobil zlato nagrado v Leipzigu. Ekipa v Sloveniji pa je tudi dobila priznanje takratnega Sklada Borisa Kidriča.

Take in podobne avtomatizacije so bile dobre izkušnje za prve korake v robotiko. Prvi robot smo leta 1978 razvili v sodelovanju z inštitutom Mihajlo Pupin iz Beograda. Elektromehanski del so razvili oni, računalniški del pa pri nas. Spet smo vzeli kot osnovo mikroračunalniški sistem Darta. Zanimiv pa je bil razvoj programske opreme, ki sva jo razvila Pavle Oblak in jaz. Navdušilo me je teamsko delo in modularen pristop. S Pavletom sva predvideno programsko podporo razdelila v dva dela in vsak je delal svoje. Vendar en del ni mogel delovati brez drugega. Če se prav spomnim, je eden prevzel gonilnike periferije (torej bodočega robota), drugi pa samo programiranje bodočega robota. Ker pa je delo potekalo vzporedno in vsak od naju še ni mogel imeti dela drugega, je bilo pomembno, da sva jasno definirala stične točke, danes bi temu rekli API (Application Programming Interface). Za (začasno) manjkajoče dele pa sva sestavila programski simulator (morda bolje emulator) in bila tako med seboj neodvisna. Verjetno pa je kljub temu pomagalo, da sva delala v istem prostoru.

Ko je bil »najin« del razvit in v maksimalni možni meri preizkušen, je sledilo potovanje v Beograd. In v avtu sva imela naš računalnik. Po nekajurni vožnji je sledilo prvo srečanje z elektromehanskim delom robota. In skoraj neverjeten čudež. Praktično takoj sta se »naš« in Pupinov del v celoti razumela, čemur je

Slika 2: **Spektrometer Spekol**

seveda sledila vesela novica v Ljubljano. V dolgoletni razvojni praksi se kaj takega redko zgodi. Striček Murphy (dobro znani Murphyjev zakon, da če kaj gre lahko narobe, se bo to zanesljivo zgodilo) se je očitno med dolgo vožnjo proti Beogradu utrudil in zaspal.

Po prvem jugoslovanskem robotu so sledili projekti robotizacije v sodelovanju z Gorenjem. Takrat je na IJS nastala zelo močna ekipa, ki je pokrivala praktično vse. Če se ne motim, je bilo skupaj s sodelavci Gorenja v razvoj tako ali drugače vključenih okrog 50 ljudi. V računalniškem smislu je spet služil kot osnova mikroračunalnik Darta, ki pa je tedaj že bil dopolnjen z disketnim sistemom. Razvoj je bil popolnoma naš. Sam lahko nekaj besed več povem o programskem delu. Tudi za disketno enoto so moji sodelavci sami razvili krmilnik (aparaturni vmesnik), za katerega pa je bilo treba šele napisati programski gonilnik. Vse, kar smo imeli na voljo, so bile specifikacije integriranega vezja v krmilniku. To je bilo treba preštudirati in nato hoditi korak za korakom. Prvi uspeh je bil, da smo znali na disketo zapisati na absolutne naslove sledi s testnimi podatki in jih ponovno prebrati nazaj. To so seveda zelo nizkonivojske operacije. Pri popularnih računalnikih PC bi temu ustrezal BIOS. Sam sem pred tem že imel izkušnje z znanim Iskrinim računalnikom IskraData 1680. Tako je sledil razvoj višjenivojskih operacij in končno smo dobili svoj lasten diskovni operacijski sistem (neke vrste DOS). Luknjani trak je šel v zgodovino, programe smo lahko iz razvojnega računalnika (Digitalov PDP) prenašali na ciljnega preko disket. A lastni razvoj ima tudi svojo ceno. »Naš« format disket je bil seveda drugačen od formata, ki ga je razumel PDP-jev operacijski sistem (takrat še RT11). Potrebna je bila pretvorba. Z nekaj spretnosti smo tudi pri našem sistemu uvedli isti format zapisa in stvar poenostavili. In nauk iz tega? Dobro se je držati uveljavljenih standardov.

Programiranje programske opreme robotov je dokaj kompleksna zadeva. Ne pozabimo, da smo vse to še vedno delali na nivoju zbirnega jezika. Kompleksnost programske opreme seveda olajša modularno programiranje, ki predvsem omogoča testiranje posameznih, malo bolj enostavnih modulov. Nedvomno so pri tem pomagale izkušnje s prvim jugoslovanskim robotom. Naslednji korak je bil razvoj lastnega večopravilnega operacijskega sistema (multitasking). Po mojih zamislih in vodstvom ga je nato realiziral v okviru svoje diplome eden od mojih študentov. Še leta sem nato na fakulteti pri predmetu »Operacijski sistemi« predaval poglavje »Naredimo si sami lasten operacijski sistem«. Danes je to seveda zastarelo in operacijski sistemi so danes precej bolj dodelani. A je delovalo. Če bi danes korakal v tej smeri, bi uporabil popularni LINUX (a tega takrat še ni bilo).

Slika 3: **Prvi jugoslovanski računalniško krmiljeni robot UMS-2. Na sliki so tudi člani ekipe Jadran Lenarčič, Pavle Oblak, Uroš Stanič in Saša Divjak**

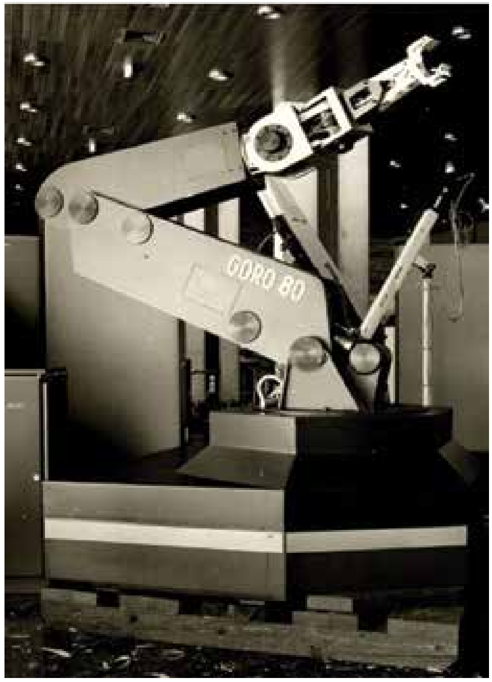
V bistvu je bila ta sistemska programska oprema jedro vseh takratnih robotov, ki smo jih razvili na Inštitutu Jožef Stefan. V letu 1979 sem sodeloval pri razvoju robota GORO 1 za Gorenje in v letu 1980 robota STEFAN 80, ki naj bi bil namenjen kovanju.

Za robot GORO 101 je v letu 1983 ekipa v sestavi Pavle Oblak, Uroš Stanič, Jadran Lenarčič, Saša Divjak, Danijel Šlebinger, Anton Ružić, Alojz Žnidaršič, Viktor Vavpot, Boris Krevzel, Miomir Vukobratović, Dragan Hristić in Miroslav Štrubelj prejela nagrado Sklada Borisa Kidriča za izume in tehnične izboljšave.

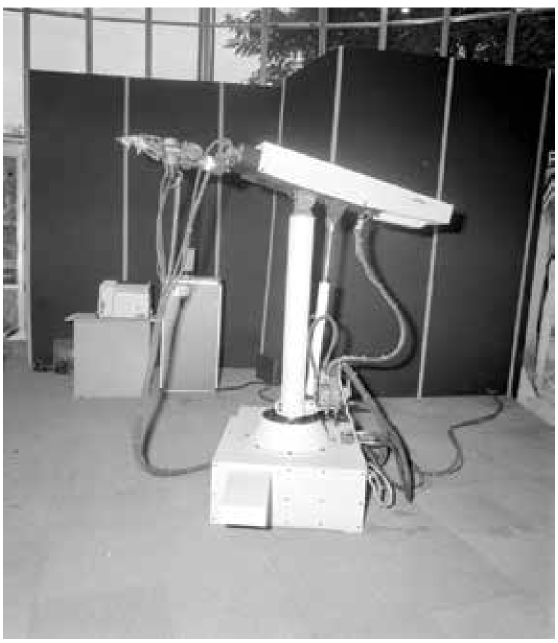
STEFAN 80 je bil zares impozanten robot, ki je s svojo roko imel delovni obseg nekaj metrov in je bil zmožen prenašati tudi malo težji tovor. Številka 80 v imenu robota pravzaprav pomeni, da je imel nosilnost kar 80 kg. Seveda so tudi pri razvoju programske opreme sodelovali tudi drugi. Spomnim se predvsem Antona Ružića, pa seveda Pavleta Oblaka.

Prav na robot STEFAN 80 je navezana še zanimiva anekdota. Robota naj bi razstavili na konferenci v Portorožu. In, ko je bil postavljen, ni delal. Kriza! Sledilo je ugotavljanje, zakaj robot miruje. Seveda človek, vsaj programer, najprej pomisli, da je napaka v njegovem programu. In sledilo je sistematično preverjanje modula za modulom. Ure so tekle in dan se je prevesil v pozno noč. In končno nenavadno odkritje. Napaka ni bila programska, pač pa začuda v mikroprocesorju, v katerem se je – zelo čudno – pokvaril en časovnik (timer), čeprav je sam mikroprocesor navidezno deloval povsem v redu. Z malo iznajdljivosti smo okvarjeni časovnik programsko simulirali in robot je spet, še pravočasno, zaživel.

Pa saj smo bili navajeni dolgotrajnega dela. Nič nenavadnega ni bilo, če smo na projektih delali tudi več kot 12 ur na dan. To so bili romantični časi, prepleteni z navdušenjem. V tistih časih se je ponekod že pojavil tudi termin inteligentne robotike. Vendar so tisti, ki so začeli s tako dejavnostjo, robote kupovali in jih nadgrajevali v smislu umetne inteligence. Mi pa smo robote razvijali in sami delali. Hiše ne moreš zgraditi, če nimaš prej temeljev. Tudi inteligentnih robotov bi ne bilo, če ne bi nekdo najprej ustvaril robotov.



Slika 5: **Industrijski robot STEFAN 80**



Slika 4: **Industrijski robot GORO1**

Izpitni centri ECDL

ECDL (European Computer Driving License), ki ga v Sloveniji imenujemo evropsko računalniško spričevalo, je standardni program usposabljanja uporabnikov, ki da zaposlenim potrebno znanje za delo s standardnimi računalniškimi programi na informatiziranem delovnem mestu, delodajalcem pa pomeni dokazilo o usposobljenosti. V Evropi je za uvajanje, usposabljanje in nadzor izvajanja ECDL pooblaščena ustanova ECDL Fundation, v Sloveniji pa je kot član CEPIS (Council of European Professional Informatics) to pravico pridobilo Slovensko društvo INFORMATIKA. V državah Evropske unije so pri uvajanju ECDL močno angažirane srednje in visoke šole, aktivni pa so tudi različni vladni resorji. Posebno pomembno je, da velja spričevalo v 148 državah, ki so vključene v program ECDL. Doslej je bilo v svetu izdanih že več kot 11,6 milijona indeksov, v Sloveniji več kot 17.000, in podeljenih več kot 11.000 spričeval. Za izpitne centre v Sloveniji je usposobljenih osem organizacij, katerih logotipe objavljamo.

Znanstveni prispevki

Jaka Stavanja, Matej Klemen, Lovro Šubelj:
NAPOVEDOVANJE UMOROV V SERIJI IGRA PRESTOLOV Z UPORABO ANALIZE OMREŽIJ

Anej Svete, Jakob Hostnik, Lovro Šubej:
NE GRE LE ZA MELODIJO

Simon Kolmanič, Maršenka Markset, Domen Mongus, Borut Žalik:
TEHNOLOGIJE NAVIDEZNE IN OBOGATENE RESNIČNOSTI KOT ORODJE ZA PREDSTAVITEV NOVIH IDEJ IN PRODUKTOV NA SEJMIH: PRIMER MAHEPA

Strokovni prispevki

Alenka Brezavšček, Maja Minič:
IZBOLJŠANJE OZAVEŠČENOSTI NA PODROČJU INFORMACIJSKE VARNOSTI Z UPORABO METOD IGRIFIKACIJE

Pogled v zgodovino

Saša Divjak:
SPOMINI NA PRVE KORAKE V ROBOTIKO V SLOVENIJI

ISSN 1318-1882
9 771318 188001