APEM journal
Advances in Production Engineering & Management
Volume 20 | Number 1 | March 2025 | pp 99–115
https://doi.org/10.14743/apem2025.1.530
ISSN 1854-6250
Journal home: apem-journal.org
Original scientific paper

Privacy-preserving AI-based framework for container transportation demand forecasting in sea-rail intermodal systems

Huang, L.[a,*], **Jiang, D.Y.**[a], **Bai, T.Y.**[b]

[a]School of Economics and Management, Beijing Jiaotong University, Beijing, P.R. China
[b]China Waterborne Transport Research Institute, Beijing, P.R. China

ABSTRACT

In response to the growing demand for accurate freight forecasting in sea-rail intermodal transportation, particularly under the constraints of stringent data protection regulations, we introduce a privacy-preserving, AI-based framework that focuses on the micro-level identification of container transport potential. The framework combines Vertical Federated Learning (VFL) with advanced feature and sample selection techniques. It leverages privacy-preserving methods, such as homomorphic encryption and random noise, enabling secure collaboration between ports and railways while safeguarding commercially sensitive data. Through extensive experiments, our framework demonstrates superior performance in predicting container transport demand, significantly improving the accuracy of resource allocation and scheduling decisions for rail operators. The framework not only ensures compliance with data protection regulations but also provides valuable insights into intermodal transportation planning, optimizing both railway operations and customer service quality. This approach offers a practical solution for improving strategic decision-making in the sea-rail intermodal sector amid increasing privacy demands and complex logistical challenges.

ARTICLE INFO

Keywords:
Freight demand forecasting;
Vertical federated learning;
Privacy-preserving methods;
Sample and feature selection;
Machine learning;
Homomorphic encryption;
Resource allocation and scheduling

**Corresponding author:*
lhuang@bjtu.edu.cn
(Huang, L.)

Article history:
Received 20 October 2024
Revised 19 February 2025
Accepted 3 March 2025

1. Introduction

With the continuous expansion of global trade, sea-rail intermodal transportation has emerged as a crucial part of modern logistics networks, effectively combining the advantages of different transportation modes and optimizing the utilization of diverse transportation resources[1]. Every day, major ports and railway systems worldwide process vast amounts of transport data, covering multiple stages from ship docking and cargo handling to final rail transportation [2]. Although this data is crucial for improving transport efficiency and optimizing logistics management, its fragmented storage across various organizations (such as customs, ports, and railway companies) presents significant challenges for efficient utilization and integrated analysis [3]. This data fragmentation not only limits information sharing and flow but also increases uncertainty throughout the transport chain. For example, delays in data exchange between port and rail departments can lead to prolonged container dwell times, ultimately reducing overall transport efficiency.

Due to the fragmented nature of the data, each party has a very limited view of the complete dataset, which not only affects information flow and collaboration efficiency but also increases operational complexity. More importantly, the growing demand for data privacy protection from data holders has made it increasingly difficult to integrate and analyse these datasets together, raising concerns about potential data leaks [4]. Therefore, developing an approach that allows for the efficient integration of fragmented data without violating privacy is now a critical challenge in sea-rail intermodal data analysis.

Federated learning offers an innovative solution to these challenges by enabling multi-party data collaboration without sharing raw data [5]. The core principle of federated learning is that participants can collaboratively train models while keeping their respective data private. Vertical Federated Learning (VFL), in particular, is well-suited for scenarios where different organizations hold different data features but share the same samples [6]. The VFL framework ensures data privacy for all parties while utilizing the complementary data from multiple sources to improve model accuracy.

Securing high-quality training datasets has always been a central challenge in machine learning and AI applications. The representativeness and quality of training data directly impact model performance. However, collecting and labelling sufficient high-quality data is costly. In a federated learning system, the selection of training samples and features plays a significant role in model performance. For instance, in horizontal federated learning, low-quality data—such as incorrect labels or skewed class distributions—can result in low and unstable model accuracy [7]. In vertical federated learning, where data features are distributed across different organizations and label access is limited, the challenge of selecting effective training samples and features becomes even more complex [8].

To address these challenges, this paper introduces a VFL framework based on Gradient Upper Bound Norms and Feature Joint Information Gain. This framework aims to optimize the analysis process for identifying potential import container sources in sea-rail intermodal transportation while ensuring data privacy protection. The key concept is to assess the importance of each participant's features to the overall model by calculating information gain, considering feature interactions via joint information gain. Sample importance is measured using Gradient Upper Bound Norms, determining which samples are best suited for model training. Additionally, the training and feature selection process incorporates homomorphic encryption and random noise to ensure privacy protection during model training. This innovative framework provides new insights and practical solutions for data analysis and decision-making in the sea-rail intermodal transportation system.

The structure of this article is as follows: Section 2 presents a thorough literature review on Freight Demand Forecasting, Vertical Federated Learning (VFL), and Sample and Feature Selection methods. Section 3 discusses the critical issue of data privacy protection within the scope of identifying potential containers for sea-rail intermodal transportation and formulates the core research problem. Section 4 provides a detailed explanation of the proposed framework, including the methodology and algorithms used to address the challenges of privacy-preserving container identification. Section 5 outlines the experimental setup and presents the results, comparing the performance of our method with existing state-of-the-art algorithms. Finally, Section 6 concludes by evaluating the effectiveness of the framework and discussing the business implications for railway container identification and intermodal transportation planning.

2. Literature review

This study stands at the intersection of the research streams on Freight Demand Forecasting, Vertical Federated Learning, Sample and Feature Selection methods. We comprehensively review the previous literature in each research stream as follows.

2.1 Freight demand forecasting

In the field of freight demand forecasting, methodologies have evolved from traditional statistical approaches to more advanced models that combine multiple techniques for improved accuracy and adaptability.

Early approaches, such as time series models like ARIMA, primarily focused on leveraging historical data trends to make future predictions. Regression models followed, incorporating external variables such as economic indicators. For instance, Khan and Khan [9] applied multivariate time series methods, including the Johansen co-integration and error correction model, to capture both short- and long-run dynamics of rail freight demand. Over time, these models have been supplemented with more sophisticated machine learning techniques, such as LSTM networks, which have proven effective in handling sequential data and capturing long-term dependencies [10].

As the complexity of freight data increased, machine learning models like Random Forests and Neural Networks gained prominence. Salais-Fierro and Martínez [11] demonstrated the superior accuracy of Artificial Neural Networks (ANNs) over traditional statistical models, particularly in forecasting freight demand using historical transportation management system (TMS) data. More recently, hybrid approaches that blend machine learning with other techniques have emerged as powerful tools for freight demand forecasting. Hassan *et al*. [12] introduced a reinforcement learning framework that combines time series models and machine learning algorithms in a rolling horizon to improve prediction accuracy over various time periods.

Other hybrid models have sought to improve interpretability and predictive power by incorporating domain-specific insights. For instance, Liu *et al*. [13] combined Grey Relational Analysis (GRA) with Deep Autoencoder Neural Networks (DNN) to enhance railway freight demand prediction. Ling *et al*. [14] introduced the Spatio-Temporal Heterogeneous Graph Attention Network (STHAN), which captures both spatial and temporal relationships within freight transportation data, demonstrating the growing complexity of models designed to account for multiple data dimensions. Econometric models also remain a staple in freight demand forecasting. Lu *et al*. [15] used input-output models to examine the effects of economic growth and structural changes on freight demand, emphasizing the continued relevance of economic indicators in freight modelling.

To date, most studies have focused on macro-level freight demand forecasting, often overlooking micro-level analysis that could optimize logistics at the individual container level. This gap in the literature is particularly important for sea-rail intermodal transportation, where predicting the transport potential of individual containers is critical for optimizing resource allocation and planning. The current study addresses this gap, offering new insights into identifying freight demand for sea-rail intermodal carriers at the micro level.

2.2 Vertical federated learning

Vertical Federated Learning (VFL) is designed for scenarios where different organizations hold disjoint sets of features for the same users or entities [16]. VFL enables organizations to jointly train machine learning models while keeping their raw data private, which is essential for privacy protection.

VFL operates primarily through two architectures: Aggregation-based VFL (aggVFL) and Split-based VFL (splitVFL) [17]. In aggVFL, each party trains its local model, and the server aggregates the results to produce a global model. Tree-based models such as SecureBoost [18] and SecureGBM [19] often operate in this framework, utilizing techniques like homomorphic encryption to ensure privacy. Meanwhile, splitVFL uses a more dynamic approach where a trainable global model is split across parties, with neural network-based models being common [20]. This allows the parties to collaborate on training without exchanging sensitive label information, with only the server retaining access to the global model [21]. Neural network-based approach has proven effective across various applications, from financial systems [22, 23] to healthcare [24], ensuring data privacy while maximizing the utility of distributed datasets.

In VFL, both sample and feature selection play crucial roles in improving communication efficiency and ensuring model performance. However, traditional methods face challenges due to privacy constraints and the large communication overhead involved. In feature selection, approaches

like SFFS [25] struggle with contextual dependencies and heavy parameter transmission. To address this, methods such as FedSDG-FS [26], LESS-VFL [27] focus on reducing the impact of noisy features through advanced filtering mechanisms, maintaining privacy while ensuring feature importance, though it lacks consideration of feature correlations. For sample selection, VF-PS [28] focuses on selecting a subset of important participants. The LEARN framework [29] proposes a solution by selecting representative samples without requiring full-sample training.

2.3 Sample selection and feature selection

In machine learning, sample selection and feature selection are essential for improving model performance, reducing computational costs, and avoiding overfitting. In centralized learning, feature selection methods are usually divided into three categories: filter, wrapper, and embedded methods [30]. Filter methods calculate statistical relationships between features and the target variable, e.g., Gini impurity [31], mutual information. Wrapper methods evaluate different feature subsets by iteratively training models, though this can be computationally expensive [32]. Embedded methods, such as Lasso regression, integrate feature selection within the training process [33], offering a more balanced approach between accuracy and efficiency. Mlinarič *et al.* [34] compared various classifiers (Decision Tree, Random Forest, Bagging, and Gradient Boosting) for feature selection in automated end-of-line quality inspection of electric motors.

Sample selection focuses on selecting the most representative or important data samples for training, especially useful in scenarios with large datasets or limited labels [35]. It is particularly critical in situations where computational resources are constrained or data labelling is expensive. Traditional sample selection methods include uncertainty-based selection, where the model selects samples the most uncertain about for further labelling [36], and representativeness-based selection, where clusters or core sets are used to select samples that represent the overall data distribution [37].

However, in VFL scenarios, the lack of global data visibility adds significant complexity to both sample and feature selection. Each party holds a portion of the data (either features or samples) and cannot directly share raw data due to privacy constraints, rendering centralized selection approaches impractical. While emerging methods like LESS-VFL [27] and LEARN [29] address these challenges by introducing secure and efficient communication protocols that enable local computations and selective data sharing, there is still considerable room for further research.

3. Problem formulation

3.1 Data sharing status and problems

Current data sharing in container sea-rail intermodal transportation heavily relies on the point-to-point exchange model, particularly through Electronic Data Interchange (EDI) between ports and railway stations. While this model provides a standardized and streamlined approach, it imposes strict requirements for data transmission based on specific message standards for different data types. As a result, the format of data exchanges is highly regulated, and participants must closely adhere to these standards when transmitting key data fields through interface protocols.

The data exchange process in sea-rail intermodal transport involves multiple key stakeholders, such as ports, customs, freight forwarders, shipping companies, railway operators, and final cargo recipients. As containers transition between modes of transport, such as from sea to rail, real-time information sharing becomes increasingly crucial. However, the current lack of a unified data-sharing infrastructure, combined with delays in exchanging critical information, can lead to several challenges. These include prolonged container dwell times, miscommunication, and operational inefficiencies, all of which may result in shipment delays, information asymmetry, or even cargo loss.

In summary, current data sharing in sea-rail intermodal transportation provides essential support for the basic operations of various stakeholders and plays a crucial role in ensuring coordination between operations and organizations. However, due to the need to protect commercial secrets, comply with data privacy regulations, and ensure data security, the scope and effectiveness of

existing data-sharing mechanisms are limited. Participants in the sea-rail intermodal chain are unable to share all their data unconditionally.

This selective data sharing, while safeguarding commercial interests, customer privacy, and data security, greatly restricts the potential for data mining. It limits the ability to fully leverage data for improving the overall efficiency of the transportation system, forecasting logistics demand, and optimizing resource allocation. Furthermore, the current reliance on message exchanges and data interfaces poses additional security risks, such as potential interception or tampering during transmission. The delays in information transfer prevent real-time updates on the transport process, hindering timely decision-making.

Additionally, challenges such as non-uniform data formats, inconsistent data quality, and compatibility issues between different information systems further complicate the data-sharing process. These problems often require extensive data cleaning and validation to ensure accuracy, increasing both the cost and complexity of data sharing.

3.2 Potential container identification scenario

In the sea-rail intermodal import process, the railway transportation workflow includes several key stages: freight forwarder application, daily train requests, railway acceptance, scheduling approval, plan preparation, and departure confirmation. Before these steps, the railway freight marketing department typically conducts freight demand mining, which is crucial for efficient resource allocation, maximizing transport efficiency, and minimizing costs.

Currently, railway freight departments conduct market research-based freight demand mining. This process involves identifying transport demand across various regions and industries. The departments engage directly with shippers, offering freight rate subsidies to encourage them to choose rail transport for container shipments from ports. However, this method is time-consuming and labour-intensive, resulting in slow progress in increasing the sea-rail intermodal ratio and facing bottlenecks. Additionally, the current approach does not utilize big data and related technologies for data analysis, limiting the accuracy and effectiveness of freight demand mining.

As illustrated in Fig. 1, the traditional railway transport process is optimized by analysing data such as port schedules, container storage, documentation, operational records, and customs information. Based on the results of potential container identification, preliminary railway transport plans—such as block trains and direct services—are formulated. The railway freight marketing department then uses these plans and transport products to conduct targeted marketing to customers. Once the freight sources are secured, the pre-compiled plans are seamlessly integrated into the existing workflow for final plan preparation. However, identifying potential container demand in real-time and ensuring privacy requires more advanced data-sharing solutions, which are discussed in the following section.

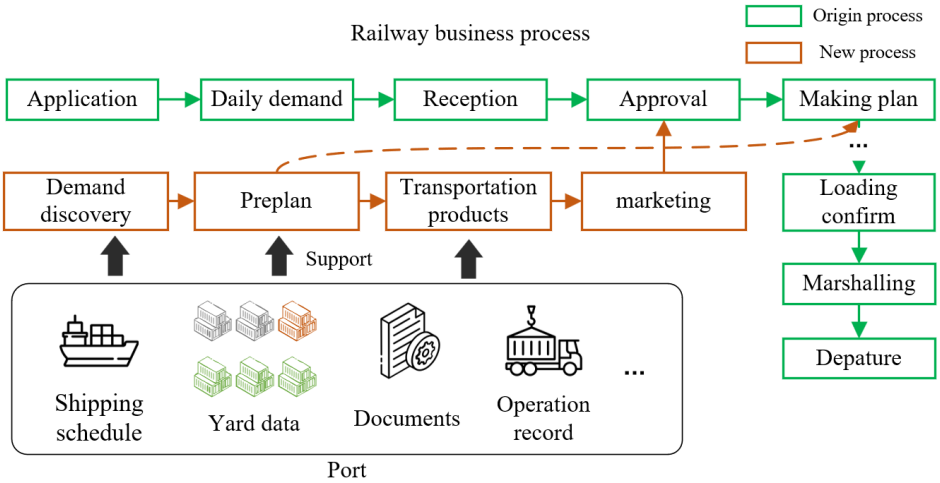

Fig. 1 Optimized railway transportation process

3.3 Privacy computing needs

The core of current data sharing in sea-rail intermodal transportation lies in supporting operational collaboration and process coordination between railway and ports, rather than indiscriminately sharing all data between both parties. The sharing mechanism focuses on improving joint operational efficiency, ensuring that data exchange enhances cargo transport efficiency, optimizes scheduling, and improves customer service. It primarily targets the exchange of essential operational data, such as train requests from ports, railway confirmations of train availability and estimated arrival times, loading confirmations, and intermediate stops.

However, the current data-sharing system is not suitable for deeper freight demand mining in sea-rail intermodal transport. Railway needs to dynamically analyse a broader range of data in real time, including ship schedules, operations, yard conditions, and destination flows of all containers. These types of raw data, however, are considered sensitive by organizations such as ports and customs and are large in volume. Under the current data-sharing framework, effective real-time sharing of this information is not possible.

The primary goal of freight demand mining in sea-rail intermodal transport is to enable railway freight marketing departments to accurately identify potential container freight demand while ensuring the security of data across multiple parties. Based on this demand, railway can dynamically optimize transport organization and train schedules. The existing data-sharing mechanism is inadequate for this purpose, requiring the development of a new solution. Techniques such as federated learning and privacy computing are necessary to allow efficient and secure information sharing, enabling the mining of potential freight demand while safeguarding sensitive data across all stakeholders in sea-rail intermodal transportation.

4. The GUBN-FJIG framework

Since the participants in sea-rail intermodal transportation, namely ports and railway, hold different features of the training samples, a vertical federated learning model is required. Existing feature selection methods typically need direct access to training data, the model training process, and labels, but this is not allowed in vertical federated learning due to privacy protection requirements. Additionally, the features held by the clients in vertical federated learning may interact with each other, and current methods tend to overlook these interactions and their joint impact on the target variable.

We proposed a vertical federated learning sample and feature selection framework based on Gradient Upper Bound Norm and Feature Joint Information Gain (GUBN-FJIG framework). It consists of three submodules: feature importance initialization, sample importance calculation, and important sample and feature selection. Fig. 2 illustrates the flowchart of this sample and feature selection framework.

In the vertical federated learning framework, the dataset of N samples is divided into M parts, denoted as $D = \{D_1, \cdots, D_M\}$, where each client holds a unique feature set $\{f_{m,1}, \cdots, f_{m,d_m}\}$ and the local sample $x_{n,m} \in \mathbb{R}^{d_m}, n \in [N]$. Typically, the server S holds the sample labels $y_n \in \mathbb{R}, n \in [c]$. With the server's coordination, all clients $m \in [M]$collaboratively contribute to the global model by sharing encrypted data to protect privacy. It allows clients to collaboratively train a global model by selecting important features and samples, while minimizing the global risk $R(\theta_s)$.

$$R(\theta_s) = \mathbb{E}_{x,y} L\left(h\left(\theta_{z_1}, z_1, \ldots, z_m, y_n\right)\right) \tag{1}$$

Each client m trains a local parameter h_m, representing the local dataset $x_n^m \in \mathbb{R}^{d^m}$, which is mapped to a lower-dimensional space $z_n^m := h_m(\theta_m, x_n^m \odot s_m) \in \mathbb{R}^{d_f^m}$, where $d_f^m \ll d^m$, and $s_m = \{0,1\}^{d_m}$ indicates the selected features. The server coordinates the process by optimizing a joint model, with parameters $\theta_0 := \{w_1, \cdots, w_M, \alpha_0\}$, where $w_m \in \mathbb{R}^{d_m'}$ are the parameters of the interaction layer. These parameters are combined with the lower-dimensional embeddings z_n^m sent by each client. α_0 represents the parameters other than interaction layer.

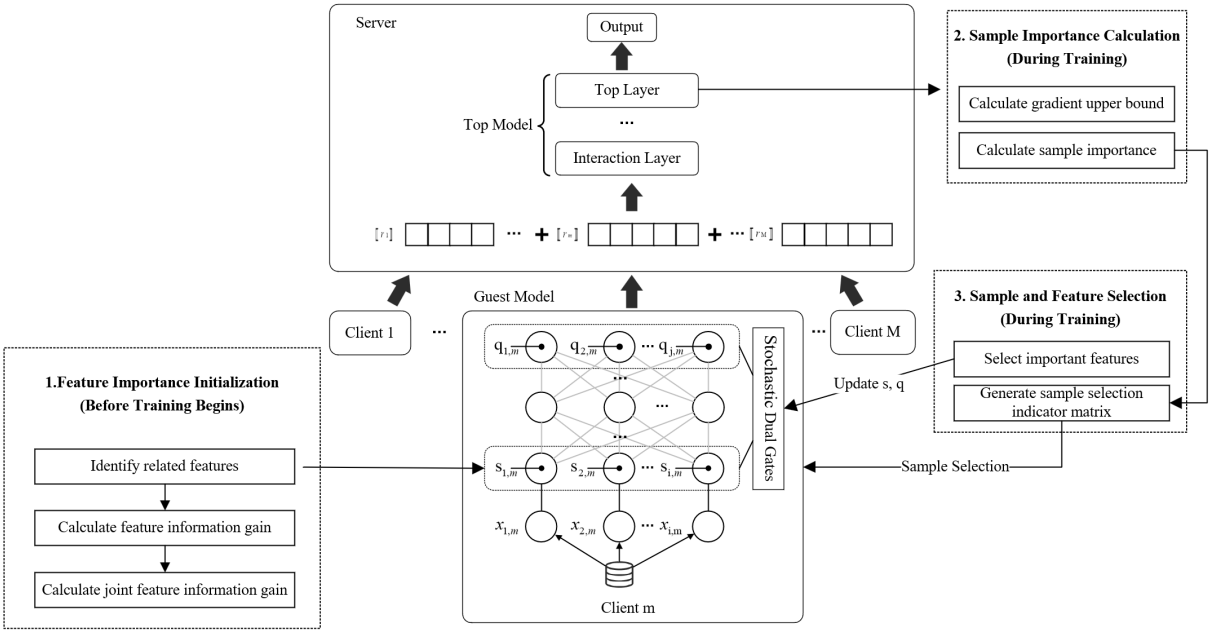

Fig. 2 GUBN-FJIG vertical federated learning framework for sample selection and feature selection

Following the Gaussian stochastic dual-gate used in FedSDG-FS [26], we utilize the l_0 norm to constrain the number of non-zero parameters in the model, minimizing the risk $R(\theta, s, q)$ to construct the global model. Due to the large variance in the Bernoulli variables s_m and q_m during feature selection optimization, a continuous relaxation based on the Gaussian distribution is applied, approximating each Bernoulli variable in s_m and q_m with parameters $\mu_{m,i}$ and $\omega_{m,j}$.

$$R(\theta, s, q) = \mathbb{E}_{x,y} L\left(h\left(\theta_0, r_{n,1}, \ldots, r_{n,M}\right); y_n\right) + \lambda \sum_m \left(|s_m| l_0 + |q_m| l_0\right) \tag{2}$$

4.1 Feature importance initialization based on information gain

For the client m, if feature $f_{m,i}$ is categorical, with possible values $\left\{f_{m,j}^{(1)}, f_{m,j}^{(2)}, \ldots, f_{m,j}^{(k)}\right\}$. The conditional entropy is defined as:

$$H\left(Y \mid f_{m,i}\right) = \sum_{f_{m,i}^{(k)}} p\left(f_{m,i}^{(k)}\right) H\left(Y \mid f_{m,i} = f_{m,i}^{(k)}\right) \tag{3}$$

where $p\left(f_{m,i}^{(k)}\right)$ is the probability of feature $f_{m,i}$ taking the value $f_{m,i}^{(k)}$, and $H\left(Y \mid f_{m,i} = f_{m,i}^{(k)}\right)$ is the entropy of Y given that $f_{m,i}$ takes the value $f_{m,i}^{(k)}$. The calculation of conditional entropy is as follows:

$$H\left(Y \mid f_{m,i} = f_{m,i}^{(k)}\right) = -\sum_{c \in \mathcal{C}} p\left(c \mid f_{m,i}^{(k)}\right) \log_2 p\left(c \mid f_{m,i}^{(k)}\right) \tag{4}$$

For continuous features, methods like binning or box plots can be used to discretize the feature. The information gain of a feature is calculated as $IG\left(Y, f_{k,i}\right) = H(Y) - H\left(Y \mid f_{k,i}\right)$. For two features $f_{m,i}$ and $f_{m,j}$ of client m, their joint conditional entropy is calculated as $H\left(Y \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right) = -\sum_{c \in \mathcal{C}} p(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}) \log_2 p(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')})$. The information gain of joint features is $IG\left(Y, f_{m,i}, f_{m,j}\right) = H(Y) - H\left(Y \mid f_{m,i}, f_{m,j}\right)$. The feature interaction information gain is calculated as $IIG\left(Y; f_{m,i}, f_{m,j}\right) = IG\left(Y, f_{m,i}, f_{m,j}\right) - IG\left(Y, f_{m,i}\right) - IG\left(Y, f_{m,j}\right)$.

The above calculations are completed through client-server collaboration, as shown in Algorithm 1. In this work, we use Paillier as homomorphic encryption method to encrypt the data requiring privacy protection during computation. This method supports addition of encrypted values and multiplication of ciphertexts by constants.

Algorithm 1: Information gain-based feature importance initialization algorithm

Input: Client m, Server S
Output: Feature importance (Initialized)
Server S
1 Compute the class entropy $H(Y)$
2 Create an indicator matrix A and encrypt: $[\![A]\!] \leftarrow \mathrm{Enc}(A)$
3 Send the encrypted indicator matrix $[\![A]\!]$ to all clients
Client m
4 Based on feature $f_{m,i}$ discretize sample U into $U_1, \cdots, U_k$
5 For feature $f_{m,i}$ calculate $p\left(f_{m,i}^{(k)}\right) \leftarrow \frac{|U_k|}{|U|}$
6 Compute $[\![p\left(c \mid f_{m,i}^{(k)}\right)]\!] \leftarrow \frac{\sum_{n \in I(U_k)} [\![A]\!]_{n,c}}{|U_k|}$
7 If there is a joint feature $f_{m,j}$, discretize sample U into $U_1, \cdots, U_{k'}$:
8 Compute $p\left(f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)$, $\left[\!\left[p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)\right]\!\right]$,
9 Add the encryption factor: $[\![\epsilon_m p\left(c \mid f_{m,i}^{(k)}\right)]\!] \leftarrow [\![p\left(c \mid f_{m,i}^{(k)}\right)]\!] \cdot \epsilon_m$,

$$\left[\!\left[\epsilon_m p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)\right]\!\right] \leftarrow \left[\!\left[p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)\right]\!\right] \cdot \epsilon_m$$

10 Send $[\![\epsilon_m p\left(c \mid f_{m,i}^{(k)}\right)]\!]$, $\left[\!\left[\epsilon_m p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)\right]\!\right]$ to the server
Server S
11 Compute $\epsilon_m p\left(c \mid f_{m,i}^{(k)}\right) \leftarrow Dec\left([\![\epsilon_m p\left(c \mid f_{m,i}^{(k)}\right)]\!]\right)$
12 Decrypt $\epsilon_m p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right) \leftarrow Dec\left(\left[\!\left[\epsilon_m p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)\right]\!\right]\right)$
13 Compute $\log_2\left(\epsilon_m p\left(c \mid f_{m,i}^{(k)}\right)\right), \log_2\left(\epsilon_m p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)\right)$ and send to the client
Client m
14 Remove noise $\log_2 p\left(c \mid f_{m,i}^{(k)}\right) \leftarrow \log_2\left(\epsilon_m p\left(c \mid f_{m,i}^{(k)}\right)\right) - \log_2 \epsilon_m$

$$\mathrm{Log}_2\, p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right) \leftarrow \log_2\left(\epsilon_m p\left(c \mid f_{m,i}^{(k)}, f_{m,j}^{(k')}\right)\right) - \log_2 \epsilon_m$$

15 Calculate $H\left(Y \mid f_{m,i}, f_{m,j}\right)$, $H\left(Y \mid f_{m,i}\right)$ and send to the Server
Server S
16 Compute information gain $IG\left(Y, f_{m,i}\right)$, $IG\left(Y, f_{m,i}, f_{m,j}\right)$ and send to the client
Client m
17 Compute joint information gain $IIG\left(Y, f_{m,i}, f_{m,j}\right)$, and initialize $\mu_{m,j}$
18 $\mu_{m,j} \propto IG\left(Y; f_{m,j}\right) + \sum_{i \neq j} IIG\left(Y; f_{m,i}, f_{m,j}\right)$

4.2 Sample importance estimation

The Gradient Upper Bound Norm is used as a sample importance indicator. Its calculation is shown below, mainly involving the input and output of the model's top layers. The computation of this norm requires only one forward pass, reducing computational costs. It provides a reasonably accurate estimate of sample importance, whereas conventional norms require both forward and backward passes through the network, making them more expensive to compute.

$$\lambda(x_n, t) = \sqrt{\left| \sum_t \beta_n^t \nabla_{\alpha_n^t} L\left(h\left(\theta_0, z_{n,1}, \cdots, z_{n,M}\right); y_n\right) \right|^2} \tag{5}$$

Here, $\lambda(x_n, t)$ represents the importance of sample x_n at iteration t, and β_n^t,α_n^t are the input and output of the top layer for sample x_n at the t-th iteration. For samples with higher gradient norms in the global model's output, the sample is assigned greater importance. Conversely, to avoid selecting samples that display unusually high values, a predefined threshold parameter $\lambda(x_n, t) \leq \delta_t$ is set, where δ_t is a user-defined parameter (e.g., the median of the sample norm distribution).

The calculation of sample importance is completed through the forward propagation process, as described in Algorithm 2. By adding random noise, the server ensures privacy protection for client data. First, the client selects a batch of samples and calculates the importance score $s_{m,i}$

based on the feature importance initialization. Encrypt intermediate results $r_{n,m}$ using Paillier before send to the Server. Then the Server adds random noise to the model's parameters ϵ_a and sends the encrypted result to the client. The client decrypts the data, removes the added noise, and sends the adjusted result $g_{n,m} + \epsilon_s$ back to the server. The server then removes the final noise ϵ_s, calculates the gradient for the top layer and completes the sample importance calculation $\lambda(x_n, t)$. If $\lambda(x_n, t) \geq \delta_t$, the sample is included for training, and a sample selection indicator matrix P is generated.

Algorithm 2： Privacy-Preserving Forward Propagation Process

Input: Client m, Server S
Output: Loss L_n, Sample importance $\lambda(x_n, t)$
Client m
Select a batch of samples $x_{n,m}$ based on the set batch size
Sample $\rho_{m,i}, \gamma_{m,j}$ from $\mathcal{N}(0, \sigma^2), i \in [d_m], j \in [\mathrm{d}']$
Calculate $\mathrm{s}_{m,i} = max\left(0, min\left(1, \mu_{m,i} + \rho_{m,i}\right)\right)$,
$\mathrm{q}_{m,j} = max\left(0, min\left(1, \omega_{m,j} + \gamma_{m,j}\right)\right)$
Record $R_m = \sum_{i \in [d_m]} \Phi\left(\frac{\mu_{m,i}}{\sigma}\right) + \sum_{j \in [d'_m]} \Phi\left(\frac{\omega_{m,j}}{\sigma}\right)$
$z_{n,m} \leftarrow h_m(\theta_m; x_{n,m} \odot s_m), r_{n,m} = z_{n,m} \odot q_m$
Encrypt $[\![r_{n,m}]\!] \leftarrow \mathrm{Enc}(r_{n,m})$
Send $[\![r_{n,m}]\!]$ to the Server S
Server S
Add random noise to the interact layer parameters: $w'_m \leftarrow w_m + \epsilon_a$
$[\![g'_{n,m}]\!] \leftarrow [\![r_{n,m}]\!] \cdot w'_m$, add random noise ϵ_s
Send $[\![g'_{n,m} + \epsilon_s]\!]$ to client m
Client m
$g'_{n,m} + \epsilon_s \leftarrow Dec([\![g'_{n,m} + \epsilon_s]\!])$
Remove the random noise ϵ_a, $g_{n,m} + \epsilon_s \leftarrow g'_{n,m} + \epsilon_s - \epsilon_a r_{n,m}$
Send $g_{n,m} + \epsilon_s$ back to the Server S
Server S
Remove the noise $g_{n,m} = g_{n,m} + \epsilon_s - \epsilon_s$
Compute $L_n \leftarrow L\left(h\left(\alpha_0, g_{n,1}, \cdots, g_{n,M}\right); \mathrm{y_n}\right)$
Obtain top layer input β_n^t, calculate $\nabla_{\alpha_n^t} L\left(h\left(\alpha_0, g_{n,1}, \cdots, g_{n,M}\right); \mathrm{y_n}\right)$
Calculate $\lambda(x_n, t)$

4.3 Backpropagation update

Based on the sample selection indicator matrix, the selected data participates in training and undergoes forward propagation, followed by model updates through backpropagation. As shown in Algorithm 3, to prevent data leakage, the server adds noise ϵ_s to the gradient $\left[\!\left[\frac{\partial L_n}{\partial w_m}\right]\!\right]$ during transmission. The client decrypts the result and adjusts the gradient by a scaling factor η_s before sending it back to the server. The cumulative noise ϵ_m is recorded during this process. Server updates interaction layer parameters $w_m^{'} = w_m + \epsilon_m$ with noisy gradients. The update of client-side model requires no noise, as the server uses encrypted cumulative noise for gradient calculations $\frac{\partial L_n}{\partial g_{n,m}} \cdot w_m^{'} - [\![\epsilon_a]\!] \cdot \frac{\partial L_n}{\partial g_{n,m}}$. The server sends the updated gradient back to the client, where the client uses backpropagation to update parameters such as $\mu_m, \omega_m, \theta_m$, thereby completing feature selection with s_m, q_m and updating the client model.

Algorithm 3: Privacy-Preserving Backpropagation Process

Input: Sample loss L_n, Server learning rate η_s, Client learning rate η_m
Output: Global model
Server S

1 Compute the gradient $\left[\!\left[\frac{\partial L_n}{\partial w_m}\right]\!\right] \leftarrow \frac{\partial L_n}{\partial g_{n,m}} \cdot [\![r_{n,m}]\!], \left(\frac{\partial L_n}{\partial r_{n,m}}\right)' \leftarrow \frac{\partial L_n}{\partial g_{n,m}} \cdot w_m', \frac{\partial L_n}{\partial \alpha_0}$

2 Add random noise ϵ_s, and send $\left[\!\left[\frac{\partial L_n}{\partial w_m} + \epsilon_s\right]\!\right]$ to client m

Client m

3 $\frac{\partial L_n}{\partial w_m} + \epsilon_s \leftarrow Dec\left(\left[\!\left[\frac{\partial L_n}{\partial w_m} + \epsilon_s\right]\!\right]\right)$

4 Add random noise ϵ_m, $\left(\frac{\partial L_n}{\partial w_m} + \epsilon_s\right)' \leftarrow \frac{\partial L_n}{\partial w_m} + \epsilon_s - \frac{\epsilon_m}{\eta_s}$

5 Encrypt the noise $[\![\epsilon_a]\!] \leftarrow Enc(\epsilon_a)$ and accumulate noise $\epsilon_a \leftarrow \epsilon_a + \epsilon_m$

6 Send $\left(\frac{\partial L_n}{\partial w_m} + \epsilon_s\right)'$, $[\![\epsilon_a]\!]$ to the server

Server S

7 Remove the noise $\left(\frac{\partial L_n}{\partial w_m}\right)' \leftarrow \left(\frac{\partial L_n}{\partial w_m} + \epsilon_s\right)' - \epsilon_s$

8 Update the interaction layer parameters

$$w_m' \leftarrow w_m' - \eta_s \left(\frac{\partial L_n}{\partial w_m}\right)', \alpha_0 \leftarrow \alpha_0 - \eta_s \nabla_{\alpha_0} L_n$$

9 Compute gradients, update other layer parameters

10 Remove the noise $\left[\!\left[\frac{\partial L_n}{\partial r_{n,m}}\right]\!\right] \leftarrow \left(\frac{\partial L_n}{\partial r_{n,m}}\right)' - [\![\epsilon_a]\!] \cdot \frac{\partial L_n}{\partial g_{n,m}}$, and send to client m

Client m

11 $\frac{\partial L_n}{\partial r_{n,m}} = Dec\left(\left[\!\left[\frac{\partial L_n}{\partial r_{n,m}}\right]\!\right]\right)$, compute gradients $\frac{\partial L_n}{\partial \mu_m}, \frac{\partial L_n}{\partial \omega_m}, \frac{\partial L_n}{\partial \theta_m}$

12 Update the client model

$$\mu_m \leftarrow \mu_m - \eta_m \left(\frac{\partial L_n}{\partial \mu_m} + \lambda \frac{\partial R_m}{\partial \mu_m}\right), \omega_m \leftarrow \omega_m - \eta_m \left(\frac{\partial L_n}{\partial \omega_m} + \lambda \frac{\partial R_m}{\partial \omega_m}\right)$$

$$\theta_m \leftarrow \theta_m - \eta_m \frac{\partial L_n}{\partial \theta_m}$$

5. Framework evaluation using practical data

In this section, we apply the proposed framework in a practical scenario to identify potential containers at a port in China. We choose the metrics of accuracy and to evaluate the performance of the proposed framework. We also compare our results against baseline models.

5.1 Data description and preprocessing

The integration of the framework faces several challenges. First, there is the issue of data integration. Since the data formats used in port and railway management systems vary, significant effort will be required for data standardization and preprocessing. Secondly, many existing systems rely on outdated infrastructure, which may not be compatible with the proposed framework and may require upgrades. Finally, collaboration among multiple stakeholders is key to ensuring smooth integration.

The data for this study were gathered from several sources. Container-related data, including basic information, shipping schedules, stack storage, and operational records, were obtained from the port's container management system. The railway transport data was collected from the China Railway Research Institute, covering two transport stations at the port, with data on daily demands, waybills, and trajectory information. Road transportation data for container trucks was sourced from Baidu Maps, utilizing truck route planning services based on primary truck models and destinations. Data collection spanned from June 2022 to July 2023. Table 1 provides a summary of the datasets, including descriptions, record counts, and ownership.

Table 1 Data sources

Dataset	Fields	Numbers	Data Owner
Container Basic Information Dataset	Container ID, Type, Size, Weight, Goods Description, Trade Type, etc.	7322158	Port System
Shipping Schedules Dataset	Estimated & Confirmed Arrival Times, Work Start & Completion Times, Departure Time, etc.	1048575	Port System
Stack Storage Dataset	Stack Entry & Departure Times, etc.	9414876	Port System
Container Operation Dataset	Destination, Dispatch Time, Mode of Transportation, etc.	485792	Port System
Container Truck Road Transportation Dataset	Transportation Distance, Fuel cost, Toll Fee, Freight Charges, Duration of Transportation, etc.	150	Baidu Maps
Container Railway Transportation Dataset	Departure & Arrival Stations, Distance Covered, Freight Charges, Discount Policy, Transportation Duration, etc.	76840	China Railway Research Institute

The proposed framework is designed to be highly adaptable to different geographic regions and logistics networks with varying data structures. It employs a flexible preprocessing pipeline that can accommodate diverse data formats and structures, allowing it to integrate and process data from different sources, such as ports and railways. The framework is capable of handling differences in data granularity, such as variations in feature sets or missing values, by applying localized feature augmentation and alignment methods. This enables the framework to function effectively across regions with distinct logistical setups or data sources.

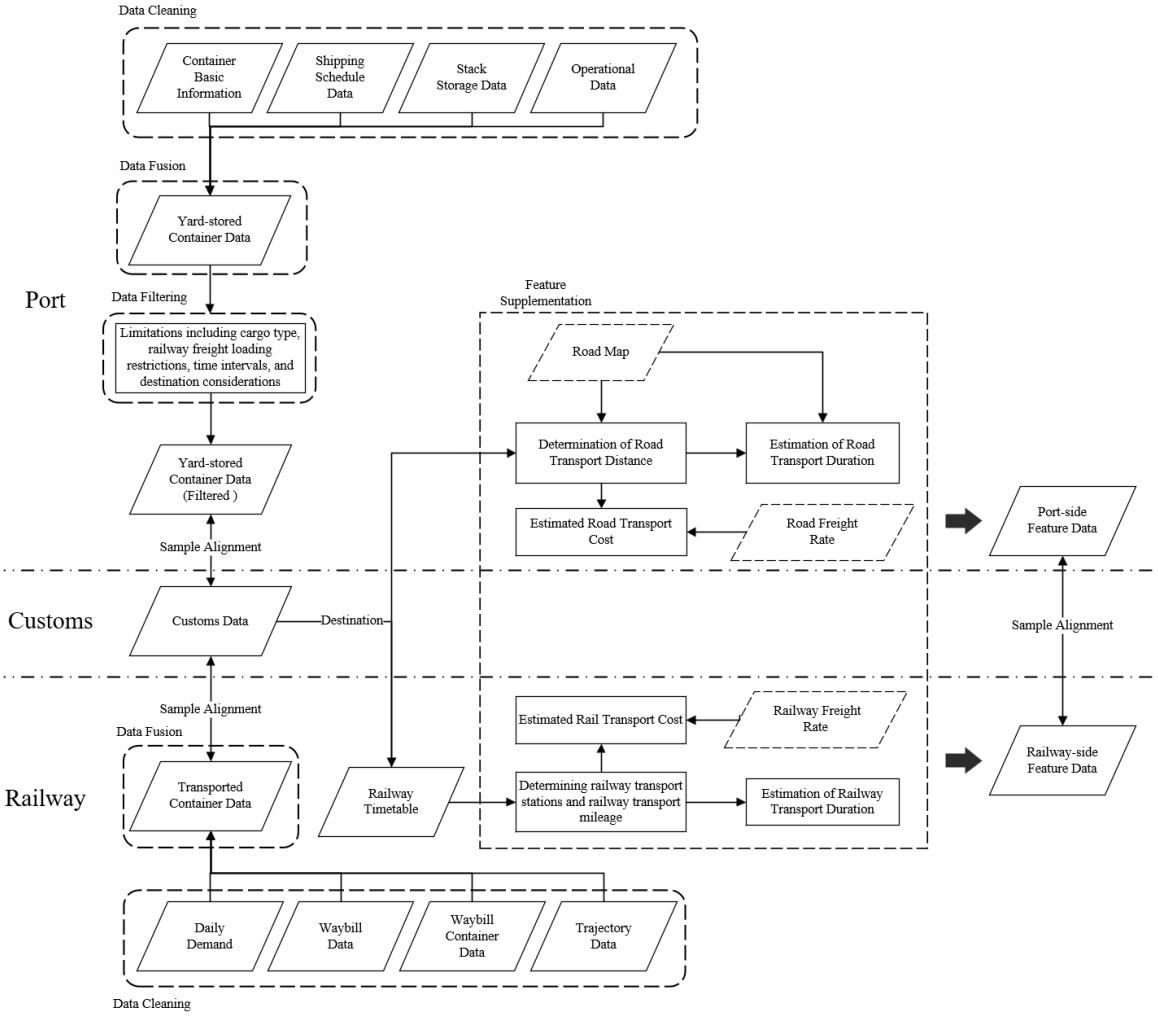

Fig. 3 Data preprocessing process in multi-party systems

All data from these three sources were integrated, as illustrated in Fig. 3. The data preprocessing in this study involves four steps: data cleaning, data integration, data filtering, and feature augmentation. The same preprocessing was applied to the port and railway datasets, though they were handled separately. After sample alignment, containers transported by road lacked railway data (distance, cost, and duration), and those transported by rail lacked road transport data. To address this, feature augmentation was applied. Missing railway features were supplemented using destination information and historical transport data, while missing road transport features were added based on destination and basic container information. Since customs hold domestic destination data, both the port and railway used pre-calculated tables for transport distance, cost, and duration for all origin-destination pairs. These were indexed using destination hashes for efficient lookup. Finally, the augmented port and railway features were aligned for consistency.

After completing the above procedures, data from multiple sources were integrated into a single dataset in logic. The dataset can be vertically divided into two parts based on the ownership of data features: port-owned features and railway-owned features, as outlined in Table 2.

Table 2 Data features and examples

Feature	Values	Type	Data Owner
Cargo weight	25.5 t, 26.3 t, …	Numeric	Port
Arrival interval	8.95 h, 6.61 h, …	Numeric	Port
Wait interval	3.13 h, 1.38h, …	Numeric	Port
Work interval	14.14 h, 6.70 h, …	Numeric	Port
Leave interval	2.80 h, 1.73 h, …	Numeric	Port
Transport interval	4.37 min, 8.20 min, …	Numeric	Port
Stack interval	249.44 h, 98.29h, …	Numeric	Port
Container type	HC, RH, FR, RF, RH, TK, …	Factor	Port
Container size	20 ft, 40 ft, …	Factor	Port
Road transportation distance	580.26 km, 149.08 km, …	Numeric	Port
Road transportation time	7.16 h, 1.93 h, …	Numeric	Port
Road fuel cost	303.58 CNY, 77.99 CNY, …	Numeric	Port
Road tolls	1047 CNY, 215 CNY, …	Numeric	Port
Road total cost	3261.04 CNY, 763.26 CNY, …	Numeric	Port
Empty container	E, F, …	Factor	Port
Trade type	D, F, …	Factor	Port
Rail transportation distance	825 km, 174 km, …	Numeric	Rail
Rail transportation time	10.31 h, 2.18 h, …	Numeric	Rail
Rail total cost	3067.6 CNY, 994.2 CNY, …	Numeric	Rail
95306 rail freight cost	3744.5 CNY, 853 CNY, …	Numeric	Rail
Discount	1439 CNY, 430.5 CNY, …	Numeric	Rail

5.2 Experimental setup

Considering that the GUBN-FJIG framework aims to identify similar transportation containers as potential freight demand, the data used for this case study was in-bureau container transport data, which has shown steady growth.

The GUBN-FJIG framework's model was trained until the prediction accuracy reached the maximum allowable iteration of 2,00. The Paillier method was used for privacy homomorphic encryption (PHE), and the Adam optimizer was applied with the learning rate and weight decay, were tuned from a grid of $\{0.01,0.005,0.002,0.001\}$, and a batch size of 128, with $\lambda = 0.1$. All other hyperparameters within the network remain at their default settings.

First, the overall model's performance was compared in terms of accuracy and on the test set, to evaluate the effectiveness of feature importance initialization based on information gain within the framework, and its comparison with other feature selection methods such as all-features, Stochastic Gates (STG) and Gini impurity using similar data protection mechanisms. The same network architecture and hyperparameters were used for all methods, with the ReLU activation function and R^2.

After completing the evaluation of feature selection methods, the effectiveness of the sample selection strategy based on the gradient upper bound norm in the framework was validated by comparing the model's training efficiency and accuracy on the test set with and without a sample selection strategy.

5.3 Results and discussion

In the case study, 5-fold cross-validation was used to evaluate the impact of different feature selection methods on model performance. The dataset was divided into five subsets, with each subset used as a validation set while the remaining four were used for training. The average performance across all five folds was taken as the final evaluation metric to reduce bias introduced by data splitting and ensure the stability and reliability of the results.

Fig. 4a shows the change in training loss for different feature selection methods during training. The FJIG method achieved the fastest initial decline in training loss and eventually reached the lowest final training loss, indicating its high efficiency and good overall model performance. Its feature selection process effectively filtered out irrelevant features, allowing the model to focus on more valuable ones, improving training efficiency. Fig. 4b shows the average validation accuracy of the five-fold cross-validation using different feature selection methods (including no feature selection). STG, Gini, and FJIG methods are compared in terms of average accuracy. The results show that the FJIG method achieved significantly better validation accuracy than the other methods, especially after epoch 100, where its accuracy remained stable with less fluctuation. In contrast, the testing accuracy of the all-features method was significantly lower than other methods.

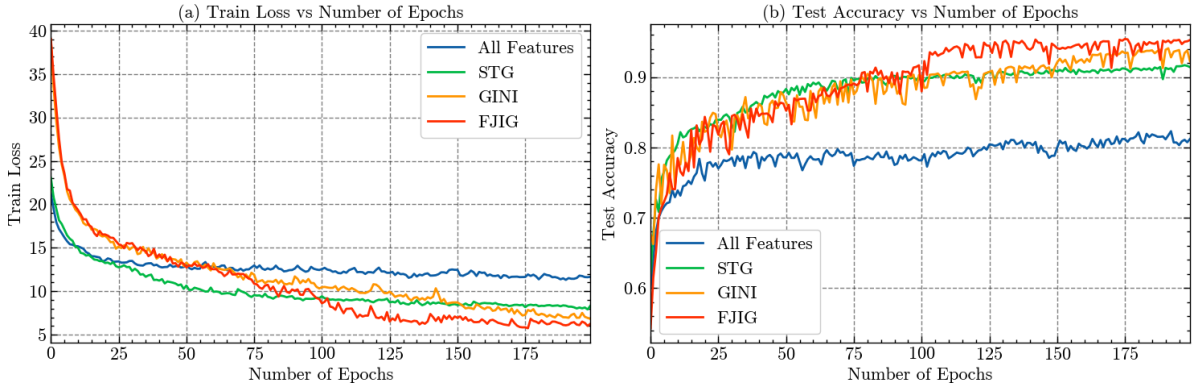

Fig. 4 Training loss and test accuracy of different feature selection methods over training epochs

Fig. 5 shows the average R^2 values across different feature selection methods during the 5-fold cross-validation. The results indicate that the FJIG method consistently maintained the highest R^2 value throughout the training process. In particular, towards the later stages of training, FJIG's R^2 value stabilized at a high level, significantly outperforming other feature selection methods. This suggests that the FJIG method, by removing noisy features, better fits the data and improves the model's predictive ability. Both STG and GINI also performed well in terms of R^2 but slightly lagged behind FJIG. The R^2 value for the all-features selection method was significantly lower than the other methods, indicating that it struggled to effectively utilize the features, especially in the presence of many noisy features.

Fig. 6 shows the changes in the number of selected features during training for different feature selection methods. The FJIG method quickly reduced the number of features early in the training process and stabilized at the minimum number of features towards the later stages. In contrast, the STG and GINI methods selected slightly more features than the FJIG method. The ability of the FJIG method to maintain high model performance with fewer features demonstrates its effectiveness in the feature selection process.

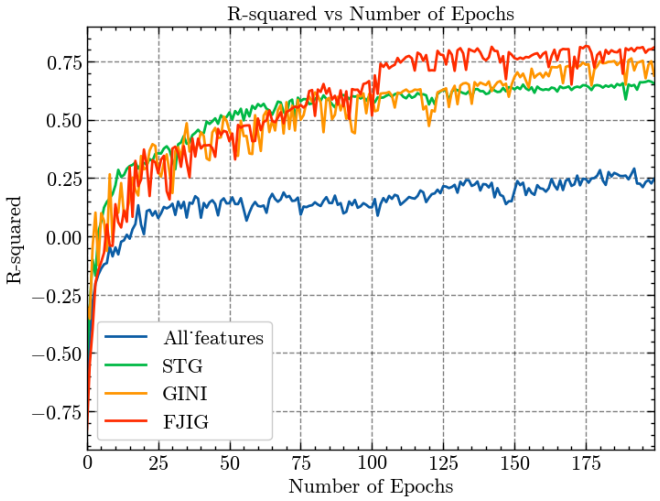

Fig. 5 R-squared of different feature selection methods over training epochs

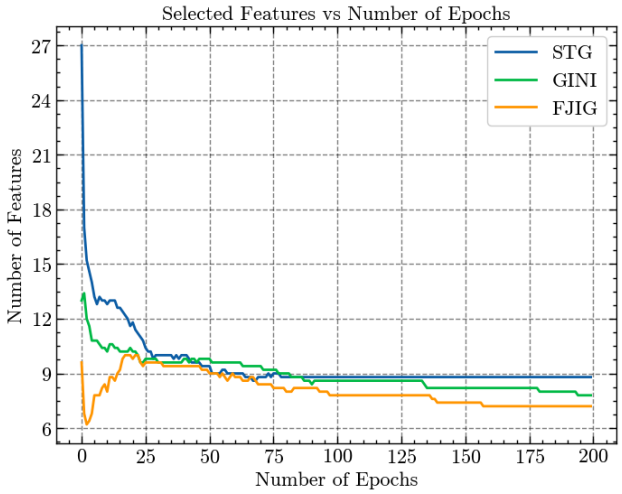

Fig. 6 Number of selected features by different feature selection methods over training epochs

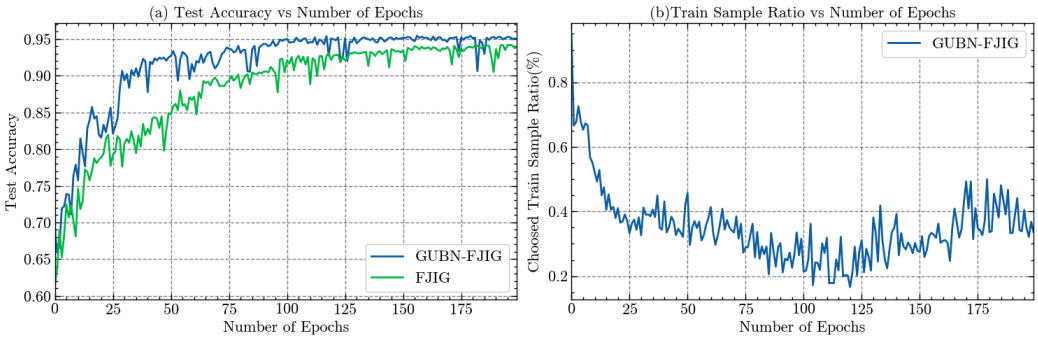

Fig. 7 Test Accuracy and Training Sample Selection Ratio of GUBN-FJIG Method over Training Epochs

Fig. 7a shows the changes in test accuracy during training for the GUBN-FJIG method with and without sample selection. In the early stages of training (around the first 50 epochs), the test accuracy of the GUBN-FJIG method increased rapidly and gradually stabilized around 0.95. In contrast, the FJIG method's test accuracy was slightly lower throughout the training process, stabilizing between 0.90 and 0.95. The superior performance of the GUBN-FJIG method in terms of test accuracy indicates that its sample selection effectively improved the model's ability to generalize to unseen test data.

Fig. 7b shows the proportion of selected training samples as the training progresses for the GUBN-FJIG method. In the early stages of training, the proportion of selected samples gradually decreased, likely because the model had not yet converged, making sample importance judgments

less stable, leading to more samples being excluded. As training progressed, the proportion of selected samples stabilized and slightly increased towards the later stages, with the final selection rate stabilizing at around 40 %. This demonstrates that the GUBN-FJIG method can dynamically adjust the number of training samples involved in the process. Reducing the number of training samples in the early stages may help accelerate model convergence. In the later stages, slightly increasing the number of selected samples ensures that the model is exposed to enough information near convergence, further optimizing performance. In conclusion, the GUBN-FJIG method enhances training efficiency and generalization performance by effectively selecting training samples. The dynamic changes in the sample selection ratio reflect the method's advantage in evaluating and adapting to sample importance at different stages of training.

6. Conclusion

In this study, we proposed the GUBN-FJIG framework, which combines Gradient Upper Bound Norms (GUBN) and Feature Joint Information Gain (FJIG) for effective sample and feature selection in container transportation demand forecasting. Our approach addresses the challenges of identifying potential freight containers in the sea-rail intermodal transportation system while ensuring data privacy and computational efficiency.

Through extensive experiments, we demonstrated that the GUBN-FJIG method significantly improves model performance in terms of both accuracy and efficiency. By dynamically selecting important samples during training and filtering out irrelevant features, the method accelerates model convergence, reduces overfitting, and enhances the model's generalization ability. Our results showed that GUBN-FJIG consistently outperformed other feature selection methods, such as STG and GINI, especially in scenarios with noisy features and large datasets.

Moreover, the GUBN-FJIG method's dynamic adjustment of the number of training samples during different stages of training contributed to its superior performance. By selecting fewer samples in the early stages to speed up convergence and increasing the sample size near convergence, the model maintained a balance between training efficiency and final performance.

In conclusion, the GUBN-FJIG framework offers a robust solution for container transportation demand forecasting in sea-rail intermodal systems. It not only optimizes model performance but also ensures data privacy protection through privacy-preserving techniques such as homomorphic encryption and random noise. From a business perspective, the framework enhances the ability of railway operators to more accurately identify potential container freight demand, leading to more informed decision-making for resource allocation and capacity planning. This results in improved operational efficiency, reduced transportation costs, and better coordination between sea and rail modes, ultimately improving service reliability and customer satisfaction. Future work could explore other scenarios of intermodal transportation systems and further improving the feature and sample selection strategies for even more efficient training and prediction.

Acknowledgements

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 52172311) and the Fundamental Research Funds for the Central Universities (Grant No. 2024JBZX042).

References

[1] Hou, D.N., Liu, S.C. (2024). Optimization of cold chain multimodal transportation routes considering carbon emissions under hybrid uncertainties, *Advances in Production Engineering & Management,* Vol. 19, No. 3, 315-332, doi: 10.14743/apem2024.3.509.

[2] Muñuzuri, J., Onieva, L., Cortés, P., Guadix, J. (2020). Using IoT data and applications to improve port-based intermodal supply chains, *Computers & Industrial Engineering*, Vol. 139, Article No. 105668, doi: 10.1016/j.cie.2019.01.042.

[3] Song, D. (2021). A literature review, container shipping supply chain: Planning problems and research opportunities, *Logistics*, Vol. 5, No. 2, Article No. 41, doi: 10.3390/logistics5020041.

[4] Lee, C., Ahmed, G. (2021). Improving IoT privacy, data protection and security concerns, *International Journal of Technology, Innovation and Management (IJTIM)*, Vol. 1, No. 1, 18-33, doi: 10.54489/ijtim.v1i1.12.
[5] McMahan, B., Moore, E., Ramage, D., Hampson, S., Aguera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data, In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics,* Fort Lauderdale, USA, 1273-1282.
[6] Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., Ouyang, Y., Zhang, Y.-Q., Yang, Q. (2024). Vertical federated learning: Concepts, advances, and challenges, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, No. 7, 3615-3634, doi: 10.1109/TKDE.2024.3352628.
[7] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., Zhang, W. (2023). A survey on federated learning: Challenges and applications, *International Journal of Machine Learning and Cybernetics*, Vol. 14, No. 2, 513-535, doi: 10.1007/s13042-022-01647-y.
[8] Khan, A., Thij, M.T., Wilbik, A. (2023). Vertical federated learning: A structured literature review, *ArXiv,* doi: 10.3934/mbe.2023122.
[9] Khan, M.Z., Khan, F.N. (2020). Estimating the demand for rail freight transport in Pakistan: A time series analysis, *Journal of Rail Transport Planning & Management*, Vol. 14, Article No. 100176, doi: 10.1016/j.jrtpm.2019.100176.
[10] Zhao, L., Cao, N., Yang, H. (2023). Forecasting regional short-term freight volume using QPSO-LSTM algorithm from the perspective of the importance of spatial information, *Mathematical Biosciences and Engineering*, Vol. 20, No. 2, 2609-2627, doi: 10.3934/mbe.2023122.
[11] Salais-Fierro, T.E., Martínez, J.A.S. (2022). Demand forecasting for freight transport applying machine learning into the logistic distribution, *Mobile Networks and Applications*, Vol. 27, No. 5, 2172-2181, doi: 10.1007/s11036-021-01854-x.
[12] Hassan, L.A.H., Mahmassani, H.S., Chen, Y. (2020). Reinforcement learning framework for freight demand forecasting to support operational planning decisions, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 137, Article No. 101926, doi: 10.1016/j.tre.2020.101926.
[13] Liu, C., Zhang, J., Luo, X., Yang, Y., Hu, C. (2023). Railway freight demand forecasting based on multiple factors: Grey relational analysis and deep autoencoder neural networks, *Sustainability*, Vol. 15, No. 12, Article No. 9652, doi: 10.3390/su15129652.
[14] Ling, S., Yu, Z., Cao, S., Zhang, H., Hu, S. (2023). STHAN: Transportation demand forecasting with compound spatio-temporal relationships, *ACM Transactions on Knowledge Discovery from Data*, Vol. 17, No. 4, 1-23, doi: 10.1145/3565578.
[15] Lu, C., Fu, S., Fang, J., Huang, J., Ye, Y. (2021). Analysis of factors affecting freight demand based on input-output model, *Mathematical Problems in Engineering*, Vol. 2021, No. 1, Article No. 5581742, doi: 10.1155/2021/5581742.
[16] Yang, Q., Liu, Y., Chen, T., Tong, Y. (2019). Federated machine learning: Concept and applications, *ACM Transactions on Intelligent Systems and Technology,* Vol. 10, No. 2, Article No. 12, doi: 10.1145/3298981.
[17] Liu, Y., Kang, Y., Zou, T., Pu, Y., He, Y., Ye, X., Ouyang, Y., Zhang, Y.-Q., Yang, Q. (2022). Vertical federated learning: Concepts, advances, and challenges, *IEEE Transactions on Knowledge and Data Engineering,*Vol. 36, No. 7, 3615-3634, doi: 10.1109/TKDE.2024.3352628.
[18] Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D.; Yang, Q. (2021). Secureboost: A lossless federated learning framework, *IEEE Intelligent Systems*, Vol. 36, No. 6, 87-98, doi: 10.1109/MIS.2021.3082561.
[19] Feng, Z., Xiong, H., Song, C., Yang, S., Zhao, B., Wang, L., Chen, Z., Yang, S., Liu, L., Huan, J. (2019). SecureGBM: Secure multi-party gradient boosting, In: *Proceedings of 2019 IEEE International Conference on Big Data (Big Data),* Los Angeles, USA, 1312-1321, doi: 10.1109/BigData47090.2019.9006000.
[20] Romanini, D., Hall, A.J., Papadopoulos, P., Titcombe, T., Ismail, A., Cebere, T., Sandmann, R., Roehm, R., Hoeh, M.A. (2021). Pyvertical: A vertical federated learning framework for multi-headed splitNN, *ArXiv,* doi: 10.48550/arXiv.2104.00489.
[21] Fu, C., Zhang, X., Ji, S., Chen, J., Wu, J., Guo, S., Zhou, J., Liu, A.X., Wang, T. (2022). Label inference attacks against vertical federated learning, In: *Proceedings of 31st USENIX Security Symposium,* Boston, USA, 1397–1414.
[22] He, H., Wang, Z., Jain, H., Jiang, C., Yang, S. (2023). A privacy-preserving decentralized credit scoring method based on multi-party information, *Decision Support Systems*, Vol. 166, Article No. 113910, doi: 10.1016/j.dss.2022.113910.
[23] Wang, Z., Xiao, J., Wang, L., Yao, J. (2024). A novel federated learning approach with knowledge transfer for credit scoring, *Decision Support Systems*, Vol. 177, Article No. 114084, doi: 10.1016/j.dss.2023.114084.
[24] Xu, X., Peng, H., Bhuiyan, M.Z.A., Hao, Z., Liu, L., Sun, L., He, L. (2022). Privacy-preserving federated depression detection from multisource mobile health data, *IEEE Transactions on Industrial Informatics*, Vol. 18, No. 7, 4788-4797, doi: 10.1109/TII.2021.3113708.
[25] Pan, F., Meng, D., Zhang, Y., Li, H., Li, X. (2020). Secure federated feature selection for cross-feature federated learning, In: *Proceedings of NeurIPS Workshop SpicyFL,* 1-12.
[26] Li, A., Huang, J., Jia, J., Peng, H., Zhang, L., Tuan, L.A., Yu, H., Li, X.-Y. (2023). Efficient and privacy-preserving feature importance-based vertical federated learning, *IEEE Transactions on Mobile Computing,* Vol. 23, No. 6, 7238-7255, doi: 10.1109/TMC.2023.3333879.
[27] Castiglia, T., Zhou, Y., Wang, S., Kadhe, S., Baracaldo, N., Patterson, S. (2023). LESS-VFL: Communication-efficient feature selection for vertical federated learning, *ArXiv,* doi: 10.48550/arXiv.2305.02219.
[28] Jiang, J., Burkhalter, L., Fu, F., Ding, B., Du, B., Hithnawi, A., Li, B., Zhang, C. (2022). VF-PS: How to select important participants in vertical federated learning, efficiently and securely?, In: *Proceedings of 36th Conference on Neural Information Processing Systems (NeurIPS 2022),* New Orleans, Louisiana, USA, 2088-2101.

[29] Liu, T., Lyu, F., Ma, J., Deng, Y., Chen, J., Tan, Q., Zhang, Y. (2023). LEARN: Selecting samples without training verification for communication-efficient vertical federated learning, In: *Proceedings of GLOBECOM 2023-2023 IEEE Global Communications Conference,* Kuala Lumpur, Malaysia, 1217-1222, doi: 10.1109/GLOBECOM54140.2023.10437557.
[30] Khaire, U.M., Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review, *Journal of King Saud University - Computer and Information Sciences*, Vol. 34, No. 4, 1060-1073, doi: 10.1016/j.jksuci.2019.06.012.
[31] Li, X., Dowsley, R., De Cock, M. (2021). Privacy-preserving feature selection with secure multiparty computation, *ArXiv,* doi: 10.48550/arXiv.2102.03517.
[32] Roy, D., Murty, K.S.R., Mohan, C.K. (2015). Feature selection using deep neural networks, In: *Proceedings of 2015 International Joint Conference on Neural Networks (IJCNN),* Killarney, Ireland, 1-6, doi: 10.1109/IJCNN.2015.7280626.
[33] Huang, Y., Jin, W., Yu, Z., Li, B. (2020). Supervised feature selection through deep neural networks with pairwise connected structure, *Knowledge-Based Systems*, Vol. 204, Article No. 106202, doi: 10.1016/j.knosys.2020.106202.
[34] Mlinarič, J., Pregelj, B., Boškoski, P., Dolanc, G., Petrovčič, J. (2024). Optimization of reliability and speed of the end-of-line quality inspection of electric motors using machine learning, *Advances in Production Engineering & Management*, Vol. 19, No. 2, 182-196, doi: 10.14743/apem2024.2.500.
[35] Xu, X., Liang, T., Zhu, J., Zheng, D., Sun, T. (2019). Review of classical dimensionality reduction and sample selection methods for large-scale data processing, *Neurocomputing*, Vol. 328, 5-15, doi: 10.1016/j.neucom.2018.02.100.
[36] Wang, R., Kwong, S. (2010). Sample selection based on maximum entropy for support vector machines, In: *Proceedings of 2010 International Conference on Machine Learning and Cybernetics*, Qingdao, China, 1390-1395, doi: 10.1109/ICMLC.2010.5580848.
[37] Tipton, E. (2013). Stratified sampling using cluster analysis: A sample selection strategy for improved generalizations from experiments, *Evaluation Review*, Vol. 37, No. 2, 109-139, doi: 10.1177/0193841X13516324.