Journal of Science and Transport Technology
Journal homepage: https://jstt.vn/index.php/en

Data-driven approach in predicting truck arrival time in logistics: a field study of India

Manh Hung Nguyen[1], Cam Van Dam[2], Thi Hoai Linh Le[2], Thi Khanh Ngoc Nguyen[3,*]

[1]University of Transport Technology, 54 Trieu Khuc, Thanh Xuan, Hanoi, 100000, Vietnam

[2]Institute of Training and International Cooperation (ITIC), University of Transport Technology, 54 Trieu Khuc, Thanh Xuan, Hanoi, 100000, Vietnam

[3]Department of Management, University of Transport Technology, 54 Trieu Khuc, Thanh Xuan, Hanoi, 100000, Vietnam

Article info
Type of article:
Original research paper

DOI:
https://doi.org/10.58845/jstt.utt.2025.en.5.2.116-135

***Corresponding author:**
Email address:
ngocntk@utt.edu.vn

Received: 05/05/2025
Received in Revised Form: 19/06/2025
Accepted: 24/06/2025

Abstract: This study investigates the application of the Particle Swarm Optimization-tuned Gradient Boosting (GB-PSO) model for predicting truck arrival times. The proposed model incorporates optimized hyperparameters to enhance predictive performance, as measured by the coefficient of determination (R) and Root Mean Square Error (RMSE). Using a real-world logistics dataset, GB-PSO outperforms conventional Gradient Boosting (GB) and expected travel time estimations, achieving a higher R value and lower RMSE across training and testing datasets. The analysis of SHAP values highlights the dominant influence of transportation distance on model predictions. These findings validate the effectiveness of GB-PSO in practical logistics optimization, reducing error and improving reliability in time-sensitive transportation systems.

Keywords: Truck delivery, Logistics, Time of arrival, Machine Learning, Data-driven approach.

1. Introduction

Accurate prediction of truck arrival times is a critical factor in enhancing operational efficiency and customer satisfaction in logistics. Timely and reliable deliveries help reduce costs and strengthen supply chain performance. However, real-world logistics involve a wide range of dynamic and uncertain factors such as traffic conditions, weather variability, and operational delays, making time of arrival (TOA) prediction particularly challenging [1].

Traditional estimation methods such as shortest-path algorithm, deterministic route planning, and basic statistical models offer limited adaptability to real-time changes [2]. These methods, although foundational, often oversimplify the dynamic nature of real-world logistics. Traffic simulation models, such as cellular automata and queuing theory, have been employed to provide more realistic estimates by incorporating traffic congestion patterns and route dependencies. Statistical regression models, including linear regression and time-series forecasting, have also been utilized for modeling historical travel data to predict future TOA. However, these classical methods struggle to account for complex, nonlinear relationships and the interaction of multiple variables affecting travel time. Traditional methods,

such as shortest-path algorithms or linear regression, depend on deterministic models utilizing historical data or predefined parameters. While these models are foundational, they often assume static inputs and struggle to capture the nonlinear, interdependent nature of logistics data. As a result, they tend to underperform in highly dynamic operational environments.

With the rapid evolution of digital technologies and the increasing availability of large-scale datasets, data-driven methods have emerged as a promising alternative. Machine learning (ML) models, in particular, offer the ability to analyze complex, non-linear relationships within data and deliver highly accurate predictions. [3], [4], [5], [6], [7]. Recent advancements in predictive analytics and machine learning (ML) have significantly enhanced the accuracy of arrival time predictions by leveraging real-time data. Studies, such as Spoel et al. [8] and Barlogis et al. [9] demonstrate the use of IoT and data-driven methods to predict truck arrival times at distribution centers and intermodal transport hubs. Other research, including Žunić et al. [10] and Li et al. [11] applies adaptive ML frameworks to real-world logistics problems, integrating spatial-temporal data and dynamic routing. Notably, most ML applications focus on maritime logistics [12], [13], leaving truck-based logistics underexplored. This paper builds on these advances to develop a robust ML-based framework for predicting truck arrival times in the Indian logistics sector, combining insights from both classical approaches and modern adaptive techniques.

Recent advancements in machine learning (ML) have significantly improved estimated time of arrival (ETA) predictions in logistics. Some studies have employed deep learning approaches such as LSTM combined with GPS and traffic data to capture dynamic spatio-temporal patterns [14] while others have developed hybrid ensemble frameworks integrating XGBoost and LightGBM for urban delivery forecasting [15]. More recent efforts have explored privacy-preserving architectures using federated learning for dynamic ETA prediction [16]. Despite these developments, there remains a notable gap in interpretable and optimization-enhanced models for truck-based logistics, especially in the context of emerging economies.

Most of these studies, however, focus on urban last-mile delivery or containerized intermodal transport in developed countries [8]. Research specific to truck-based logistics in emerging markets like India remains sparse, despite vastly different infrastructural and operational characteristics. India's logistics sector presents unique challenges: highly fragmented road infrastructure, with inconsistent quality across regions, traffic unpredictability due to unregulated local flows, mixed vehicle types, and lack of traffic signal harmonization., low telematics penetration in older fleets, leading to gaps in real-time visibility, variable regulatory enforcement across states, causing checkpoint delays, extreme heterogeneity in shipment profiles, vehicle types, and route patterns.

This study presents a data-driven approach to predicting truck arrival times in the Indian logistics sector, leveraging machine learning techniques to address the challenges of variability and uncertainty. A comprehensive dataset encompassing multiple factors, including transportation distance, cargo type, and operational details, is utilized to train and validate predictive models. Gradient Boosting (GB) and its optimization with Particle Swarm Optimization (PSO) are explored to assess the impact of hyperparameter tuning on model performance.

The methodology adopted in this study includes rigorous feature selection, cross-validation techniques, and performance evaluation based on metrics such as R-value and RMSE. Additionally, this research contributes to the existing body of knowledge by focusing on the logistics landscape in India, characterized by its unique infrastructure, traffic patterns, and operational constraints.

The findings of this study aim to provide actionable insights for logistics stakeholders, enabling more precise arrival time estimates and informed decision-making. By addressing the complexities of truck delivery through a data-driven framework, this research paves the way for improved efficiency and competitiveness in logistics operations.

2. Methodology flowchart of this investigation

The methodology flowchart presented in Fig. 1 outlines the step-by-step process used in a data-driven approach for predicting the time of arrival in truck logistics. This approach consists of six distinct stages, each of which contributes to the overall predictive model.

Stage I. Building dataset

The first stage involves collecting the raw data, which consists of 32 features and 6880 rows. This data is typically collected from various logistics operations, capturing relevant information about shipments, transportation, and associated variables. The raw data is used as the foundation for further processing and analysis.

In this stage, the raw data is cleaned and refined to form a usable dataset with 9 features (8 input variables and 1 output variable) and 4653 rows. This involves handling missing data, encoding categorical variables, and possibly eliminating outliers. The dataset represents the input to the machine learning models used for prediction. Descriptive statistics and visualizations, such as histograms or correlation matrices, may also be produced to better understand the data's structure and relationships.

Stage II. Data analyze

This stage focuses on transforming the data into a more suitable form for model training. The process may involve feature engineering, where new features are created from the existing ones, and feature selection, where irrelevant or redundant features are removed. For example, interaction terms between variables or normalization techniques could be applied to enhance the predictive power of the model. Visualizations such as feature importance plots or heatmaps help assess the contribution of each feature to the model's performance.

Stage III. Model Training and Tuning

In this stage, machine learning (ML) models are trained on the preprocessed dataset. The flowchart mentions the use of Gradient Boosting (GB) with default hyperparameters and Gradient Boosting with Particle Swarm Optimization (GB-PSO) algorithms for tuning the hyperparameters. These models are trained on both the training dataset of 70% whole dataset. Hyperparameters of the models, such as the number of trees or maximum depth should be optimized using optimization algorithm as Particle Swarm Optimization. Cost function of optimization process is coefficient of correlation R.

Stage IV. Model Evaluation and Validation

After training, the models undergo evaluation and validation to assess how well they generalize to unseen data. In this stage, the model's R scores and RMSE values on both the training and testing datasets are compared to ensure the performance of each ML model. Validation techniques, as Monte Carlo simulation, are employed to verify the robustness and stability of the model's predictions. A combination of box plots and scatter plots may be used to visualize residuals and assess model fit.

Stage V. Using best ML model for predicting truck arrival time

The best-performing ML model, identified in the previous stage, will be utilized to predict the time of arrival in truck logistics. The predicted time of arrival will also be compared with the estimated time of arrival provided in the raw dataset.

Stage VI. Feature Importance and Interpretation In the final stage, the most important features that contribute to the prediction of the time of arrival are identified and analyzed. Feature importance plots provide insights into which variables (such as Transportation Distance, Type of Vehicle, or Destination of Location) have the strongest influence on the model's predictions. The

model is also interpreted in terms of how each feature impacts the output variable, with an emphasis on how this knowledge can guide decision-making in real-world logistics operations.

In this data-driven methodology, the six stages provide a comprehensive process for preparing, training, evaluating, and interpreting a model that predicts the time of arrival in truck logistics. Starting from the raw data collection, the process advances through preprocessing, model training, and evaluation, culminating in the identification of key features that drive the model's predictions. This methodology can be used to improve logistical operations and optimize delivery times by leveraging machine learning insights.

In this study, the hyperparameters for the Gradient Boosting model were tuned once using Particle Swarm Optimization (PSO), with the objective function evaluated via 10-Fold Cross-Validation on the initial training subset (70% of the dataset). The optimal hyperparameter configuration obtained from this tuning step was then held constant throughout the subsequent 5000 Monte Carlo Simulation (MCS) runs.

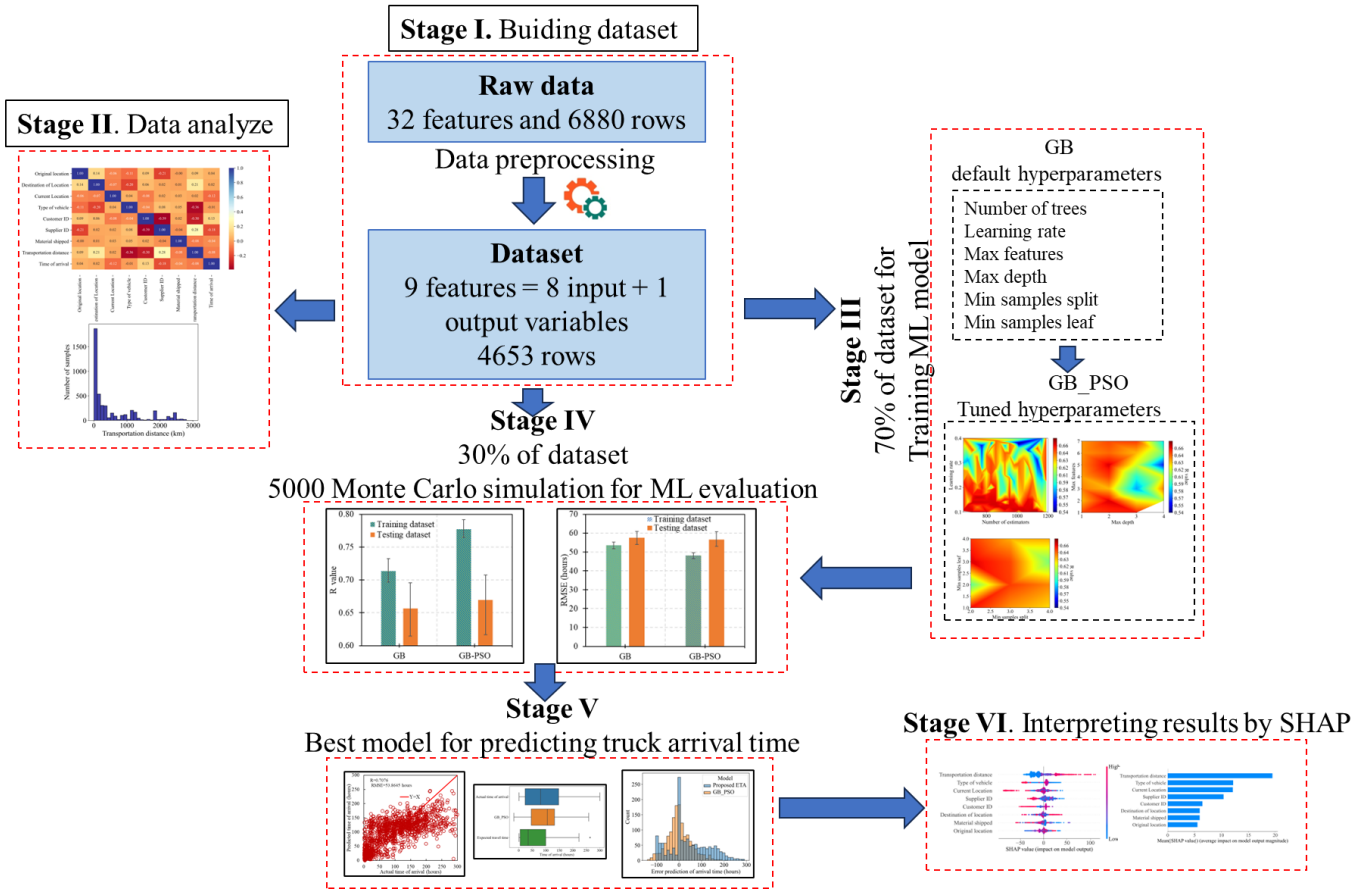


Fig. 1. Methodology flowchart of data driven approach for predicting time of arrival in truck logistics

During each MCS run, only the train/test split was randomly varied, while the model architecture and hyperparameters remained fixed. This design isolates the effect of data variability on model performance and ensures a fair comparison between models. The use of k-Fold Cross-Validation was strictly limited to the hyperparameter tuning phase and not repeated during the MCS evaluation.

3. Description of database

To construct a robust and efficient machine learning model, the original dataset comprising 32 features was refined and reduced to a final set of 9 features, including 8 input variables and 1 output variable (Time of Arrival). The feature selection process was conducted using a combination of statistical analysis, domain relevance, and data quality criteria.

Firstly, features were retained based on their logical and statistical relevance to the target

variable. Variables directly related to transportation—such as original location, destination, current location, type of vehicle, customer and supplier identifiers, material shipped, and transportation distance—were prioritized. Features deemed redundant or irrelevant, such as driver names, mobile numbers, and text-based identifiers, were excluded due to their lack of predictive utility.

Secondly, features with a high proportion of missing data were removed from the dataset to maintain consistency and model robustness. This included several administrative or operational fields that were either sparsely populated or inconsistent across records.

Categorical variables were then encoded using label encoding to transform them into numerical formats suitable for use in tree-based algorithms like Gradient Boosting. Additionally, geospatial fields were simplified by focusing on numerical measures such as transportation distance, rather than using raw latitude and longitude pairs.

The final processed dataset included the following features: Original location, Destination of location, Current location, Type of vehicle, Customer ID, Supplier ID, Material shipped, Transportation distance (input features), and Time of arrival (output variable). This reduced and cleaned feature set preserved the most relevant information while ensuring data quality, interpretability, and computational efficiency for the learning models.

The Origin location spans 162 unique places, with the most frequent being Mugabala, Bangalore Rural, Karnataka (415 occurrences), while the Destination location includes 391 unique places, led by DAIMLER INDIA COMMERCIAL VEHICLES, KANCHIPURAM (323 occurrences). The Current location field, which has 394 missing values, lists 2,128 unique entries, with Perumalpattu - Kottamedu Rd, Oragadam Industrial Corridor appearing 178 times. Vehicle information is represented by the Type of vehicle, featuring 42 distinct categories, with 40 FT 3XL Trailer 35MT dominating (1,691 instances) despite 694 missing entries.

During the data cleaning and preprocessing phase, missing values were carefully addressed to ensure the completeness and consistency of the dataset. A total of 394 missing entries were identified in the "Current location" field. These were retained and marked with a placeholder value (*"Unknown"*) to reflect the potential status of shipments in transit, thereby avoiding the risk of introducing bias through imputation. In the case of the "Type of vehicle" field, 694 missing entries were detected. These were imputed using the most frequently occurring category (*"40 FT 3XL Trailer 35MT"*), as this type of vehicle was predominantly used and considered representative. This strategy was adopted to preserve the dataset's analytical validity while minimizing information loss.

The Customer ID attribute identifies 38 unique customers, with LTLEXMUM40 being the most frequent (2,478 records). Regarding numerical data, Transportation distance varies from 3 km to 2,954.7 km, with an average of 587.83 km, while Time of arrival ranges from 1 to 300.21 hours, averaging 95.19 hours. Lastly, the Supplier ID field lists 262 unique suppliers, with 55471 as the most common (402 occurrences), and the Material shipped feature encompasses 1,158 types, led by AUTO PARTS (1,079 instances). In the 8 features, there are only one quantitative variable as Transportation distance and time and 7 categorical variables as Original location, Destination of location, Current location, type of vehicle, customer ID, supplier ID, material shipped.

Fig. 2 depicts two quantitative variables as input variable Transportation distance (cf. Fig. 2a) and output variable Time of arrival (cf. Fig. 2b) which illustrates the distribution of the time of arrival (in hours) for a set of samples. The x-axis represents the time of arrival, ranging from 0 to 300 hours, while the y-axis represents the number of samples observed. The data is highly skewed towards shorter arrival times, with the majority of

samples (over 600) arriving within the first-time interval (close to 0 hours). The frequency decreases as the time of arrival increases, showing a long tail extending towards higher times. This distribution suggests a rapid initial accumulation of samples, followed by a gradual decline over time.

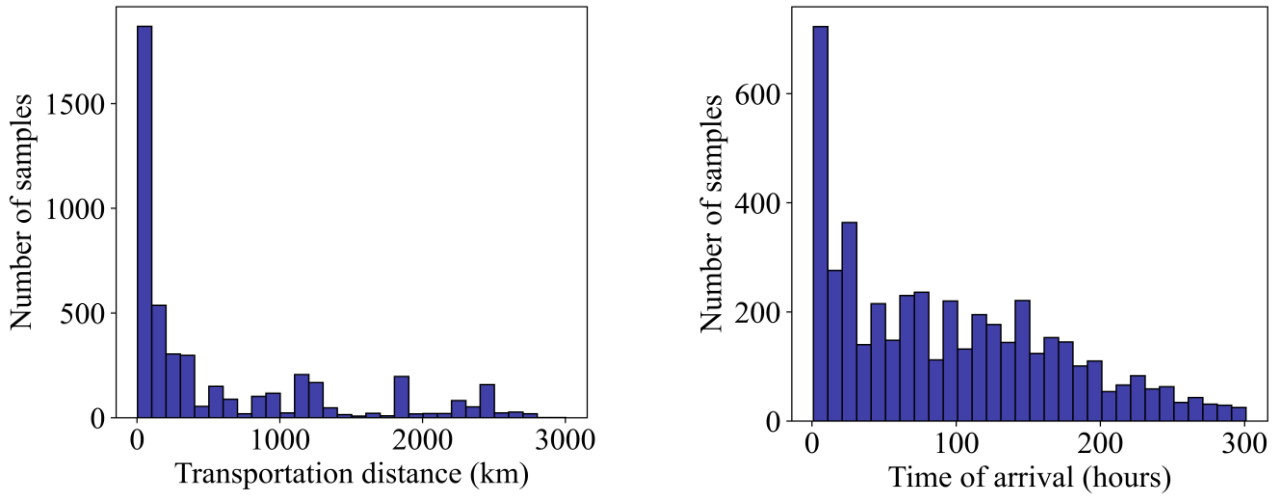


Fig. 2. Histogram of two quantitative (a) input variable Transportation distance and (b) output Time of arrival

Fig. 3 presents a Pearson correlation matrix that quantifies the linear relationships among multiple variables related to logistics and transportation. Each cell in the matrix represents the correlation coefficient between a pair of variables, with values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation).

The diagonal elements of the matrix, which are all equal to 1, reflect the perfect correlation of each variable with itself. Off-diagonal elements provide insights into the strength and direction of relationships between variables. Positive correlations are represented by shades of blue, while negative correlations are shown in shades of red, with the intensity indicating the magnitude of the correlation.

The Pearson correlation matrix provides comprehensive insights into the relationships among the eight key features: Original Location, Destination of Location, Current Location, Type of Vehicle, Customer ID, Supplier ID, Material Shipped, and Transportation Distance. Below is a detailed summary of these relationships.

Original Location shows a positive correlation with Destination of Location (0.14) and Transportation Distance (0.09), indicating that the origin of shipments is modestly linked to the destinations and distances traveled. However, a moderate negative correlation with Supplier ID (-0.21) suggests that certain suppliers are more associated with specific origin locations, potentially reflecting structured supply chain dynamics.

Destination of Location is moderately correlated with Transportation Distance (0.21), which aligns with the expectation that destinations further away are associated with longer travel distances. However, a weak negative correlation with Type of Vehicle (-0.20) suggests that certain destinations may influence vehicle selection, possibly due to infrastructure constraints or shipment requirements.

Current Location exhibits negligible correlations with most features. A very weak positive correlation with Type of Vehicle (0.04) and a slight negative correlation with Customer ID (-0.08) indicate that its variability is not strongly dependent on these factors, reflecting a lack of influence from other features.

Type of Vehicle is negatively correlated with Transportation Distance (-0.36), suggesting that different vehicle types are used for shipments with

varying travel lengths, potentially due to cost, efficiency, or payload capacity. A weak positive correlation with Supplier ID (0.08) indicates that certain suppliers may prefer or rely on specific vehicle types.

Customer ID demonstrates a strong negative correlation with Supplier ID (-0.39), highlighting distinct supplier-customer associations in the supply chain. A weak positive correlation with Transportation Distance (0.13) implies that specific customers may require shipments over longer distances, possibly due to geographic dispersion or demand patterns.

Supplier ID shows a moderate positive correlation with Transportation Distance (0.28), reflecting that certain suppliers are associated with longer transportation routes. A negative correlation with Original Location (-0.21) and Customer ID (-0.39) further supports the existence of structured supplier-customer-location relationships in the dataset.

Material Shipped exhibits weak correlations with other features, such as a slight positive relationship with Supplier ID (0.02) and Type of Vehicle (0.05). The absence of meaningful correlations, such as with Original Location (0.00) and Transportation Distance (-0.08), suggests that shipment material plays a minimal role in shaping the observed logistics patterns.

Finally, Transportation Distance is positively correlated with Destination of Location (0.21) and Supplier ID (0.28), reflecting logical dependencies on destination and supplier behavior. A moderate negative correlation with Type of Vehicle (-0.36) highlights the influence of vehicle selection on shipment distances, potentially related to efficiency or route optimization.

Summary: The strongest relationships were observed for Transportation Distance, which is significantly tied to Destination of Location, Supplier ID, and Type of Vehicle, reflecting the central role of these factors in determining shipment distances. Conversely, features such as Material Shipped and Current Location exhibited weak or negligible correlations with most other variables, indicating limited influence on the dataset's overall variability.

The analysis of Time of Arrival revealed generally weak correlations with all eight features. The strongest relationship was a weak negative correlation with Supplier ID (-0.18), suggesting that certain suppliers might be associated with more efficient operations or shorter delivery times. A slight positive correlation with Customer ID (0.13) further implies potential prioritization or geographical effects related to specific customers. Minimal relationships were observed between Time of Arrival and other features, including Original Location (0.04), Destination of Location (0.02), and Transportation Distance (-0.08), indicating that shipment times are likely influenced by factors external to the dataset, such as traffic conditions, weather, or operational scheduling.

Overall, the features are not enough strong relation to reduce number of features in building the machine learning model for predicting Truck arrival time.

Although the Pearson correlation matrix (Fig. 2) indicates generally weak linear relationships between individual input features and the target variable (Time of Arrival), the GB-PSO model still achieves high predictive accuracy. This apparent contradiction is due to the inherent limitation of Pearson correlation, which measures only linear pairwise dependencies and does not capture non-linear or interaction effects among variables.

Gradient Boosting, especially when enhanced by PSO-tuned hyperparameters, excels at modeling non-linear, high-order relationships between features. It recursively partitions the feature space to capture complex patterns that cannot be revealed through linear correlation analysis. For example, features like transportation distance, vehicle type, and destination may exhibit weak direct linear correlation with arrival time but, when combined, form strong non-linear interactions that significantly impact predictions.

Moreover, the SHAP analysis (Fig. 9)

confirms that the GB-PSO model identifies and utilizes such interactions effectively, with transportation distance and vehicle characteristics emerging as dominant predictors despite their low Pearson coefficients. This highlights the advantage of tree-based models in handling non-additive effects and structured heterogeneity, especially in real-world logistics datasets.

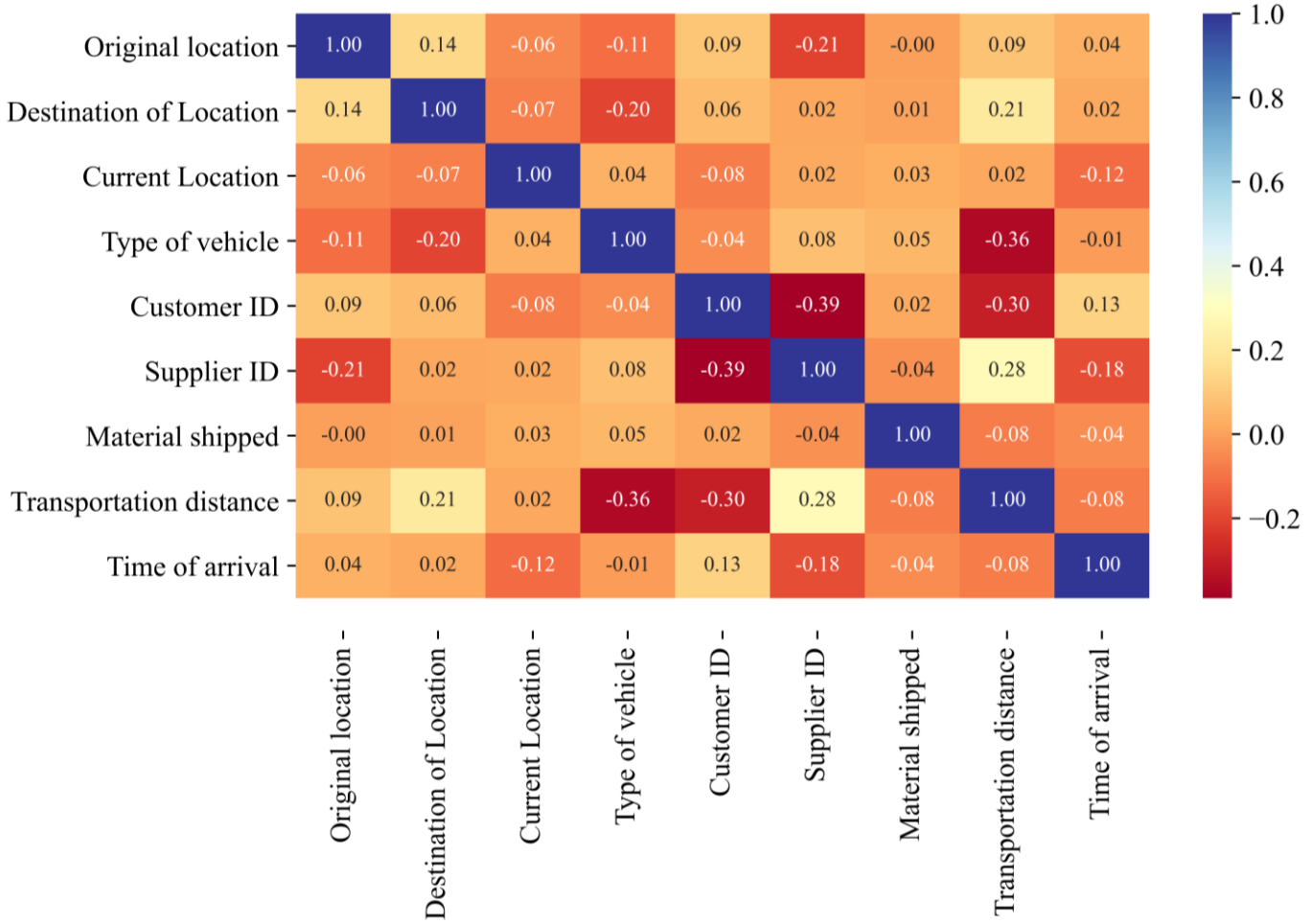


Fig. 3. Pearson correlation matrix of 9 features in the dataset

In summary, the model's superior performance reflects its capacity to transcend linear assumptions, validating the choice of non-linear machine learning over classical statistical approaches for the truck ToA prediction task.

4. Data-driven approach

The choice of Gradient Boosting (GB) and Particle Swarm Optimization (PSO) in this study is motivated by both theoretical and practical considerations. Gradient Boosting is well-regarded for its high predictive accuracy in regression tasks involving tabular data with heterogeneous feature types and potential missing values. It constructs additive models in a forward stage-wise manner and is particularly effective in capturing complex nonlinear relationships, which are common in logistics systems. To enhance GB's performance, PSO is applied for hyperparameter tuning due to its effectiveness in global optimization. Compared to grid search or random search, PSO provides a more computationally efficient mechanism to explore large and continuous hyperparameter spaces. It does not require gradient information, making it suitable for non-differentiable and complex objective landscapes.

Alternative models such as Random Forest and XGBoost were considered. However, Random Forests often lack the fine-grained optimization capacity of GB, while XGBoost, although powerful, introduces additional complexity and memory

overhead, which was not critical for the scale of our dataset. Recurrent neural networks like LSTM were excluded because the dataset lacks temporal sequence dependencies and is better represented in tabular form rather than time series.

Therefore, the combination of GB and PSO offers a balance between accuracy, interpretability (via SHAP), and computational efficiency, making it an appropriate choice for the prediction task in this study.

4.1. Gradient Boosting algorithm (GB)

Gradient Boosting (GB) is a powerful ensemble learning algorithm designed to improve predictive performance by combining multiple weak learners, typically decision trees, into a strong [17]. The algorithm works iteratively by building successive models that correct the errors of previous models, minimizing a loss function in the process.

The model starts with a simple initial prediction, often the mean value for regression problems. At each iteration, a decision tree is trained to minimize the residuals (errors) of the previous model, using gradient descent to optimize the loss function. The predictions from all trees are aggregated (summed) to form the final model output. Each tree is weighted by a learning rate, which controls the contribution of individual trees to the final prediction.

Gradient Boosting often achieves superior accuracy by effectively reducing bias and variance in the data. It can handle various loss functions and be adapted for both regression and classification tasks. The algorithm provides insights into the importance of features, aiding in interpretability. By tuning hyperparameters like the learning rate, maximum depth, and number of trees, GB can achieve a good balance between model complexity and generalization.

Gradient Boosting in solving regression problems, particularly when dealing with complex, nonlinear relationships in the data. Its ability to handle missing values, outliers, and mixed data types makes it a robust choice for predicting the time of arrival in truck logistics, as demonstrated in this study.

4.2. Particle Swarm Optimization (PSO) for Tuning Hyperparameter

Particle Swarm Optimization (PSO) is a nature-inspired optimization technique that mimics the social behavior of swarms, such as birds or fish, to identify optimal solutions in a given search space [18]. In this study, PSO is employed to fine-tune the hyperparameters of the machine learning model, thereby improving its performance in predicting the time of arrival.

A swarm of particles (potential solutions) is randomly initialized within the search space, where each particle represents a set of hyperparameters. Each particle adjusts its position in the search space based on: its personal best-known position (local best) and the best-known position of the swarm (global best).

The velocity updates are guided by the following equation (1):

$$v_i(t+1)=wv_i(t)+c_1r_1\left(p_{best}-x_i(t)\right)+c_2r_2\left(g_{best}-x_i(t)\right)$$

The position updates are guided by the following equation (2):

$$x_i(t+1)=x_i(t)+v_i(t+1)$$

Where: $v_i(t)$: Velocity of particle i at iteration t, $x_t(t)$: Current position of particle i, w: Inertia weight balancing exploration and exploitation

c_1, c_2: Acceleration coefficients for local and global bests, r_1, r_2: Random numbers in [0 ,1], p_{best}: Personal best position of the particle. g_{best}: Global best position among the swarm.

The values for parameters c_1, c_2 and w of PSO model are assigned as 0.4, 0.7, and 0.7, respectively.

Each particle's fitness is evaluated using an objective function, such as maximizing the correlation coefficient (R) or minimizing the root mean squared error (RMSE) on the training dataset. The swarm iteratively updates positions and velocities until convergence criteria are met, such as reaching a maximum number of iterations or achieving a predefined error threshold.

PSO is used to optimize key hyperparameters of the Gradient Boosting (GB) model, such as the number of estimators, learning rate, and maximum depth of trees. By leveraging PSO, the hyperparameter search becomes efficient and avoids the exhaustive computation of grid search methods. The integration of PSO significantly enhances the model's predictive accuracy and robustness by identifying the optimal combination of hyperparameters for the dataset. This ensures a better fit for predicting the time of arrival in truck logistics, as demonstrated in subsequent evaluation stages.

4.3. Evaluation criteria of Machine Learning model

To evaluate the performance of the machine learning model in predicting the time of arrival, two key metrics are employed: Correlation Coefficient (R) and Root Mean Squared Error (RMSE). These metrics provide insights into the accuracy and reliability of the model's predictions. The correlation coefficient measures the strength and direction of the linear relationship between the observed values $p_{0,j}$ and the predicted values $p_{t,j}$. The formula is expressed as (3):

$$R=\frac{\sum_{j=1}^{N}\left(p_{0,j}-\overline{p_0}\right)\left(p_{t,j}-\overline{p_t}\right)}{\sqrt{\sum_{j=1}^{N}\left(p_{0,j}-\overline{p_0}\right)^2\sum_{j=1}^{N}\left(p_{t,j}-\overline{p_t}\right)^2}}$$

RMSE quantifies the average magnitude of errors between the observed $p_{0,j}$ and predicted $p_{t,j}$ values. The formula for RMSE is given as (4):

$$RMSE=\sqrt{\frac{1}{N}\sum_{j=1}^{N}(p_{0,j}-p_{t,j})^2}$$

Where $p_{0,j}$: Observed values from the dataset, $p_{t,j}$: Predicted values by the ML model, $\overline{p_0}$: Mean of the observed values, $\overline{p_t}$: Mean of the predicted values, N: Total number of data points.

R ranges from -1 to +1, A higher absolute value of R indicates a stronger linear relationship, signifying better model performance in capturing trends. RMSE provides an absolute measure of prediction accuracy in the same unit as the target variable. Lower RMSE values indicate better model performance by reducing prediction errors.

4.4. K-Fold Cross Validation and Monte Carlo simulation

K-Fold Cross-Validation is a robust technique used to evaluate the performance of machine learning models and reduce the risk of overfitting [19]. It involves dividing the dataset into *K* equally sized folds or subsets, ensuring each fold serves as both a training and validation set at different iterations.

Four steps of the technique are described in this following:

- The dataset is split into *K* folds.
- The model is trained on *K-1* folds and validated on the remaining fold.
- This process is repeated *K* times, with each fold serving as the validation set once.
- The final evaluation metric (e.g., RMSE, R value) is averaged across all folds to provide a more reliable estimate of the model's performance.

K-Fold Cross-Validation is particularly useful during hyperparameter tuning. It ensures that the selected hyperparameters generalize well to unseen data by evaluating their performance across multiple train-test splits. This reduces the risk of overfitting the model to a single train-test split.

A common choice in practice is 10-Fold Cross-Validation, where the dataset is divided into 10 subsets. This configuration balances computational efficiency and performance evaluation. By leveraging 10 folds, the model benefits from comprehensive validation while maintaining manageable computational costs.

This technique is critical in fine-tuning machine learning algorithms such as Gradient Boosting or models optimized with Particle Swarm Optimization (PSO), as it ensures the hyperparameters selected are robust and lead to consistent performance improvements across the dataset.

Monte Carlo simulation is a validation technique in machine learning that involves

randomly splitting the dataset into training and testing subsets multiple times to evaluate model performance [20]. Unlike traditional methods like k-fold cross-validation, Monte Carlo simulation does not use fixed partitions but instead creates multiple random splits, ensuring variability in data subsets across iterations.

For each split, the model is trained on the training subset and evaluated on the testing subset, and performance metrics (e.g., accuracy, RMSE) are computed. The results are averaged over all iterations to provide a robust estimate of the model's generalization ability.

This technique is particularly useful for large datasets or when the dataset is highly imbalanced, as it enables diverse sampling and reduces the risk of bias introduced by a single partitioning. However, it requires more computational resources due to repeated training and testing processes.

4.5. SHapley Additive exPlanations for interpreting predicted result

SHAP (SHapley Additive exPlanations) is a powerful interpretability method used to explain individual predictions of machine learning models by assigning each feature a SHAP value, which quantifies its contribution to the predicted value relative to a baseline, such as the mean prediction [21]. The method decomposes the prediction into a sum of SHAP values for all features, providing a clear and additive attribution of the predicted outcome. SHAP offers both global interpretabilities, by identifying feature importance across the entire dataset, and local interpretability, by explaining specific predictions. Based on Shapley values from cooperative game theory, SHAP ensures a fair and consistent distribution of contributions, making it a robust tool for understanding and trusting model decisions.

5. Results and Discussion

5.1. Tuning Hyperparameters

The hyperparameter tuning process for the Gradient Boosting (GB) model aimed to optimize the predictive performance, as measured by the coefficient of determination (R). The hyperparameter space and tuning configurations are detailed in Table 1, which lists the ranges and increments for six key hyperparameters: the number of trees (n_estimators: 500–1200, step: 30), learning rate (0.1–0.4, step: 0.01), maximum depth of each tree (max_depth: 1–5), maximum number of features considered (max_features: 1–8), minimum samples required to split a node (min_samples_split: 2–5), and minimum samples required at a leaf node (min_samples_leaf: 1–5). The objective function for tuning was R, and the validation process employed 10-Fold Cross-Validation to ensure robustness.

The contour plots in Fig. 4 provide a detailed visualization of how different hyperparameters influence the predictive performance of the Gradient Boosting (GB) model, as quantified by the R value (coefficient of determination). Each plot captures the interaction between two specific hyperparameters while keeping the others fixed, thereby illustrating their combined effect on the model's performance. The color gradient, ranging from blue to red, represents the R value, with red indicating higher values (better performance) and blue indicating lower values (poorer performance).

Fig. 4a shows the interaction between the number of estimators (n_estimators) and the learning rate. A higher number of estimators (approaching 1200) generally leads to improved performance, provided the learning rate is low (close to 0.1). Increasing the learning rate while maintaining a high number of estimators results in reduced performance, indicating the importance of a small learning rate when using more estimators.

Fig. 4b examines the interaction between max_features (the number of features considered at each split) and max_depth (the maximum depth of each tree). The optimal region for R lies in a combination of lower max_features (around 2–3) and shallower tree depths (max_depth around 2–3). Larger max_features (e.g., greater than 5) and deeper trees (max_depth > 3) tend to result in lower R values, possibly due to overfitting or model complexity.

Fig. 4c explores the relationship between min_samples_split (minimum number of samples required to split an internal node) and min_samples_leaf (minimum number of samples required at a leaf node). The highest R values are observed when min_samples_split is around 2–3, combined with min_samples_leaf values around 2–3. Higher values for both parameters (e.g., min_samples_split > 4 and min_samples_leaf > 3) reduce performance, likely because the model becomes too constrained and underfits the data.

Overall, these contour plots serve as a powerful tool for understanding how hyperparameters interact and for identifying optimal combinations. The optimal hyperparameter combination yielded the best R value of 0.6626, achieved with the following parameter set: *n_estimators* = 1160, *learning_rate* = 0.1, *max_depth* = 2, *max_features* = 2, *min_samples_split* = 2, and *min_samples_leaf* = 3.

Table 1. Hyperparameters space of Gradient Boosting models in tuned process

Number of trees	500-1200
Learning rate	0.1-0.4
Max features	1-8
Max depth	1–5
Min samples split	2-5
Min samples leaf	1-5
Validation technique: 10-Fold CV	
Objective function for tuning hyperparameters: maximizing R	

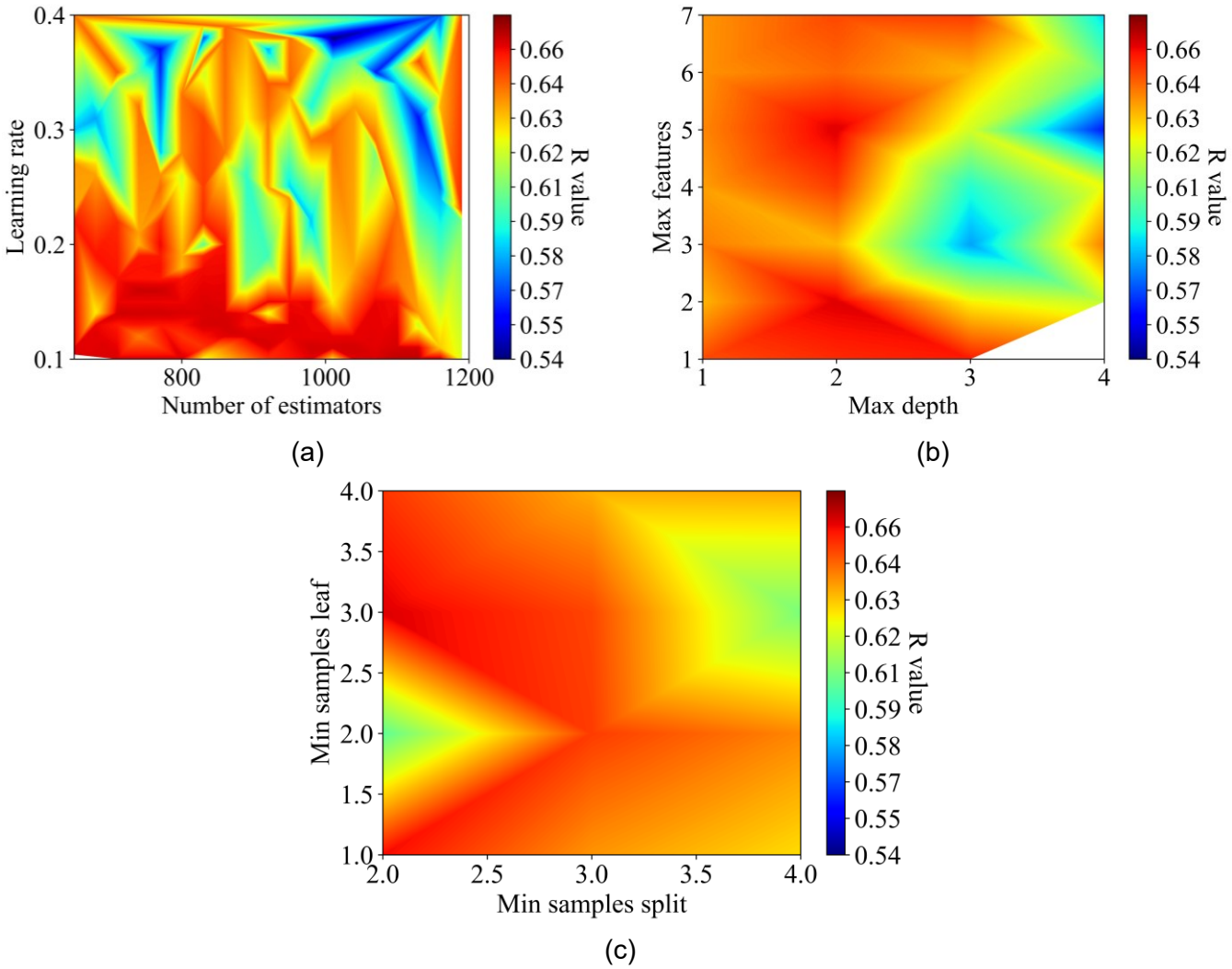


Fig. 4. Contour plot for objective function value R in tuning hypermeters of GB model

These optimal hyperparameters will be used for comparing performance between GB-PSO model with GB model using default hyperparameters in the next section.

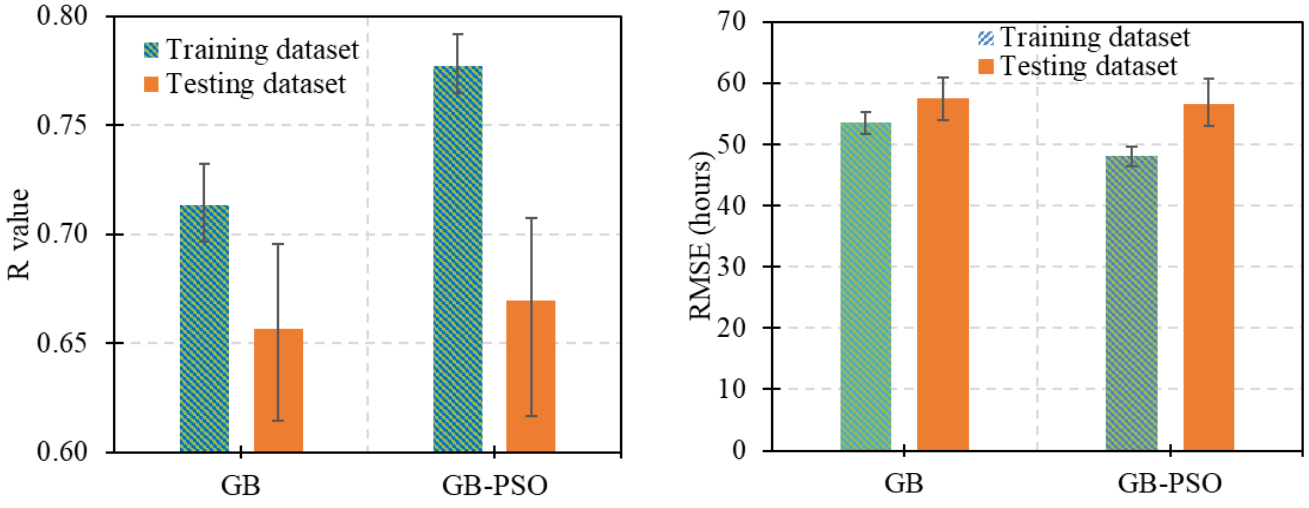
5.2. Performance comparison of Machine Learning model

The performance of the Gradient Boosting (GB) and Particle Swarm Optimization-tuned Gradient Boosting (GB-PSO) models is compared based on 5000 Monte Carlo simulation runs, as summarized in Table 2 and illustrated in Fig. 5. The performance metrics include the mean and standard deviation (Std) of the R value (coefficient of determination) and RMSE (Root Mean Square Error) for both training and testing datasets.

The results demonstrate that the GB-PSO model consistently outperforms the standard GB model on the training dataset, achieving a higher mean R value (0.7769 vs. 0.7131) and a lower mean RMSE (48.0497 hours vs. 53.5045 hours). Additionally, the standard deviation for R is significantly larger in the GB-PSO model (0.0039 vs. 0.0047), reflecting the impact of optimized hyperparameters in GB-PSO.

On the testing dataset, the GB-PSO model shows a slight improvement in the R value (0.6695 vs. 0.6564) but a marginal reduction in RMSE (56.6335 hours vs. 57.5289 hours). The standard deviations of both R and RMSE for testing indicate minor variability between the two models, with GB-PSO slightly increasing the standard deviation for R but reducing it for RMSE.

Overall, the GB-PSO model demonstrates enhanced performance, particularly for the training dataset, confirming the effectiveness of the PSO algorithm in tuning hyperparameters and improving the predictive power of the Gradient Boosting model.



(a) R (b) RMSE

Fig. 5. Performance index including R and RMSE after 5000 Monte Carlo simulations for (a) GB and (b) GB_PSO

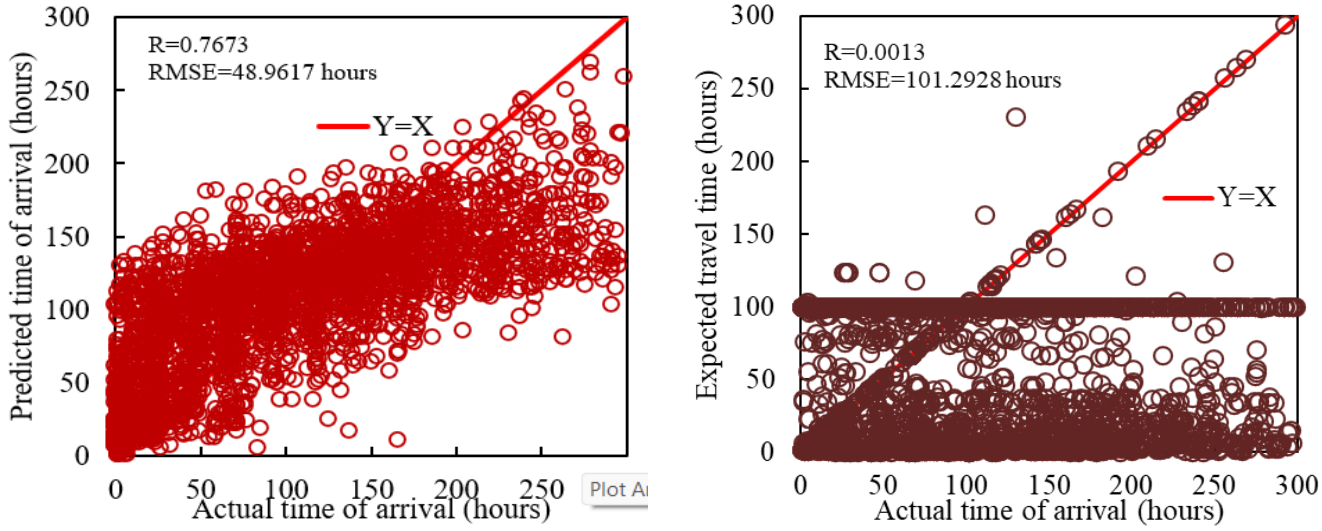
Table 2. Performance metric of GB and GB-PSO models including mean value and Std value of 5000 Monte Carlo simulation runs

ML model	Training dataset				Testing dataset			
	Mean		Std		Mean		Std	
	R	**RMSE**	**R**	**RMSE**	**R**	**RMSE**	**R**	**RMSE**
GB	0.7131	53.5045	0.0047	0.4463	0.6564	57.5289	0.0113	1.0209
GB_PSO	0.7769	48.0497	0.0039	1.4981	0.6695	56.6335	0.0126	1.0506

5.3. Prediction of Arrival Time using Machine Learning model

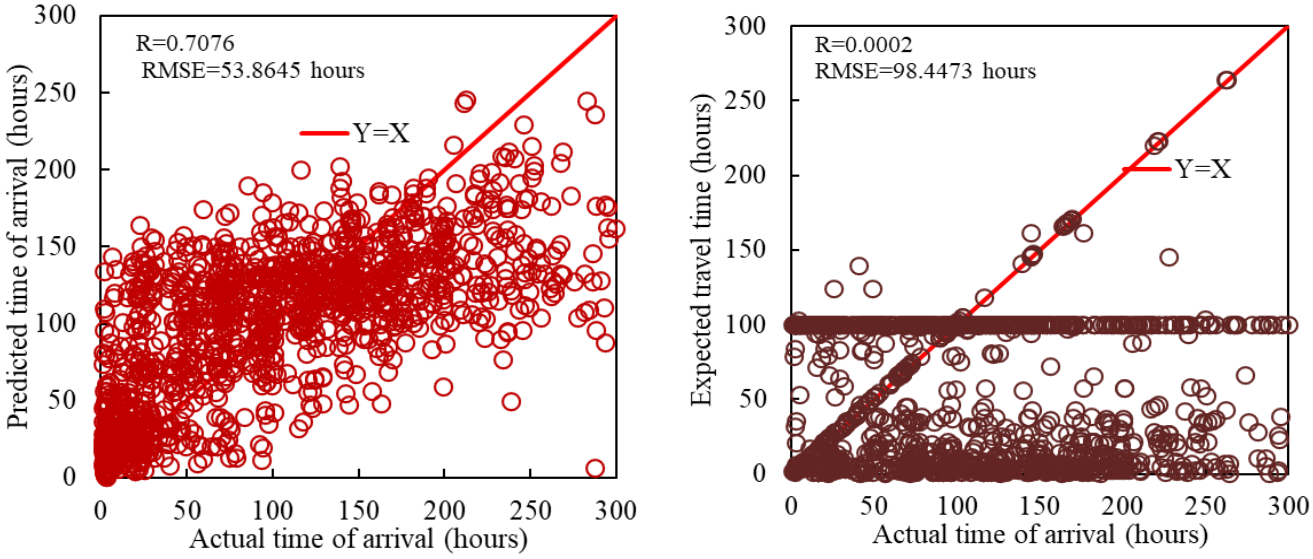
In this section, the best ML model GB_PSO is used to predicting truck arrival time. The predicted time of arrival is not only compared with actual time of arrival but also compared with expected travel time found in View The Space Data (VTS Data 280820) of Kaggle plaform. The

analysis of Figs. 6 and 7 emphasizes the superior accuracy of the GB-PSO model compared to the expected travel time in predicting truck arrival times. For the training dataset, GB-PSO achieves a strong correlation with actual times (R=0.7673) and a lower RMSE of 48.96 hours, whereas expected travel time shows negligible correlation (R=0.0013) and a higher RMSE of 101.29 hours. Similar trends are observed in the testing dataset, with GB-PSO showing R=0.7076 and RMSE=53.86 hours, significantly outperforming expected travel time (R=0.0002, RMSE=98.44 hours). This demonstrates the robust predictive performance of GB-PSO.



(a) GB-PSO (b) Expected travel time

Fig. 6. Case of training dataset for actual time of arrival compared with (a) time of arrival predicted by GB-PSO and (b) Expected travel time



(a) GB-PSO (b) Expected travel time

Fig. 7. Case of testing dataset for actual time of arrival compared with (a) time of arrival predicted by GB-PSO and (b) Expected travel time (hours)

The “expected travel time” in the raw dataset represents a baseline estimate typically computed using simple heuristic calculations based primarily on distance and average speed assumptions. It does not incorporate real-time dynamic factors such as traffic congestion, weather conditions, vehicle-specific attributes, or operational delays.

Consequently, this estimate serves as a naive or rudimentary benchmark rather than a predictive model output. Our comparison shows

that the GB-PSO model substantially outperforms this baseline by capturing complex, non-linear relationships in the data, as reflected by significantly higher correlation coefficients and lower RMSE values on both training and testing sets.

Fig. 8 illustrates the data distribution of arrival times for both training and testing datasets, comparing the actual time of arrival, predictions generated by the GB_PSO model, and the expected travel times, measured in hours. Subfigure (a) represents the distribution for the training dataset, while subfigure (b) pertains to the testing dataset. The box plots display variations in time, capturing the median, interquartile range, and potential outliers for each category. Notably, the actual time of arrival exhibits a wider range compared to both the GB_PSO predictions and expected travel times. The GB_PSO predictions demonstrate a narrower spread, suggesting consistent performance, while the expected travel time appears to have the smallest variability. These comparisons highlight the alignment and deviations among actual outcomes, model predictions, and expected values across both datasets, providing insights into the model's predictive accuracy and reliability.

Fig. 9 provides a detailed analysis of the error distribution for arrival time predictions, comparing the performance of the GB_PSO model and the proposed ETA model against the true time of arrival. Subfigure (a) represents the training dataset, while subfigure (b) focuses on the testing dataset. The histograms depict the frequency of error predictions, measured in hours. In both datasets, the GB_PSO model exhibits a narrower and more symmetric error distribution, indicating improved accuracy and consistency over proposed ETA. Conversely, the proposed ETA model shows a wider spread with higher variability, particularly with errors deviating significantly from zero. These findings further validate the effectiveness of the GB_PSO model in reducing prediction errors and enhancing reliability across both datasets.

By integrating these insights, the discussion emphasizes the comparative strengths and predictive accuracy of the proposed GB_PSO model in practical applications.

Fig. 10 provides an in-depth interpretation of the GB_PSO model's predictions for truck arrival times by analyzing the importance and influence of input features using SHAP (SHapley Additive exPlanations) values. The two subfigures present complementary perspectives on the role of each feature in the model's decision-making process.

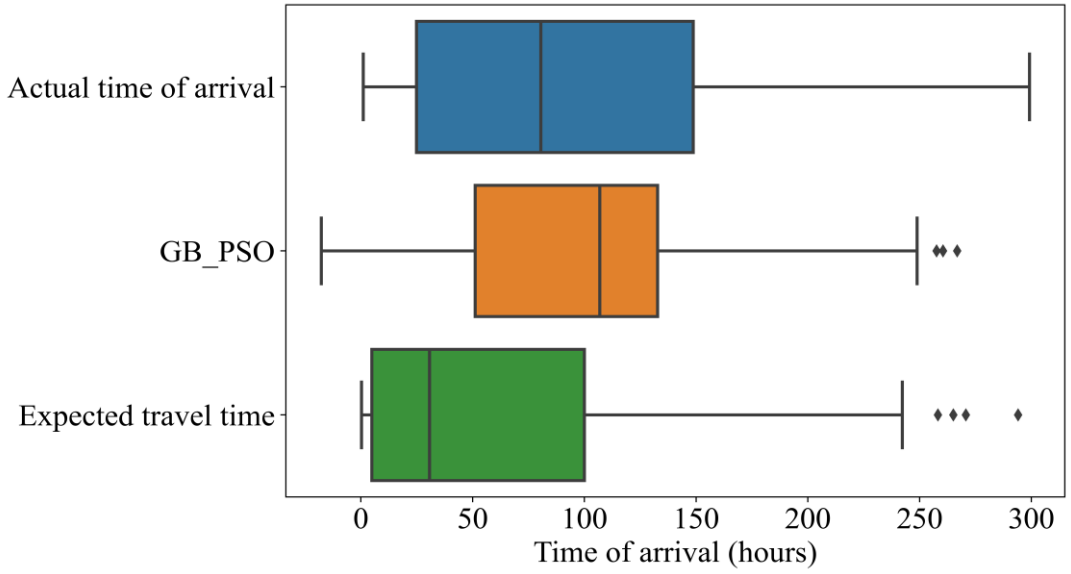
Subfigure (a) shows the distribution of SHAP values for individual features, providing insights into how each feature impacts the predicted arrival time. Each dot represents a data point, colored by the feature's value (blue for low and pink for high). For instance, "Transportation distance" has a wide spread of SHAP values, indicating a strong and diverse effect on the model's predictions. High values of transportation distance generally lead to positive SHAP values, increasing the predicted arrival time, while shorter distances reduce it. Similarly, the "Type of vehicle" feature exhibits varying impacts, with certain vehicle types significantly influencing predictions.

Subfigure (b) presents a bar chart of the mean SHAP values for each feature, which quantifies the average magnitude of their contribution to the model's output. "Transportation distance" is identified as the most critical factor, with the highest mean SHAP value, underscoring its dominant role in predicting arrival time. The "Type of vehicle" and "Current location" follow closely, reflecting their substantial influence. Other features, such as "Supplier ID," "Customer ID," and "Destination of location," show moderate impacts, whereas "Material shipped" and "Original location" are less significant in comparison.

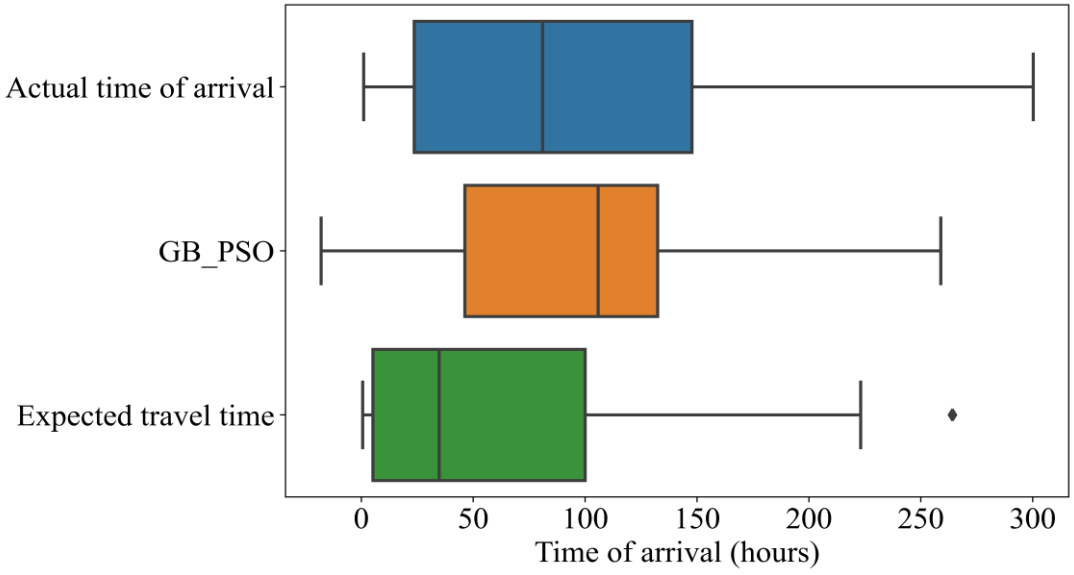
The SHAP analysis reveals that Transportation Distance is the most influential factor driving truck arrival time predictions, which is intuitive since longer distances naturally require more travel time. However, beyond this dominant feature, subtler yet critical factors such as Type of

Vehicle, Current Location, Supplier ID, and Customer ID capture variations related to vehicle performance, traffic congestion levels, supplier reliability, and customer-specific delivery requirements. Leveraging these secondary features enables logistics managers to optimize vehicle assignment to appropriate routes, proactively monitor and address potential traffic bottlenecks, and customize delivery scheduling based on customer profiles. This holistic approach enhances operational efficiency, mitigates risks, and strengthens competitive advantage. Therefore, while transportation distance sets the baseline for predictions, integrating the full spectrum of influential features allows stakeholders to make more informed and effective supply chain decisions.

This analysis highlights the transparency of the GB_PSO model by identifying the key factors that affect arrival time predictions. It underscores the importance of transportation-related variables while also acknowledging the contributions of contextual features like supplier and customer details. This information can guide decision-makers in optimizing operations and understanding the sensitivity of predictions to various inputs.

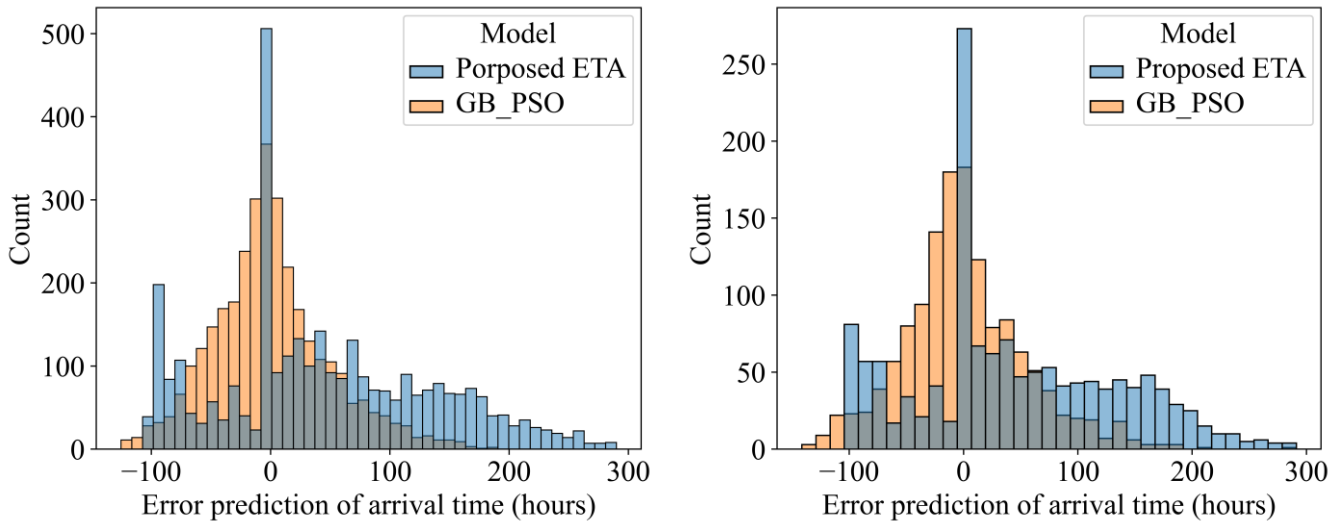


(a) Training dataset



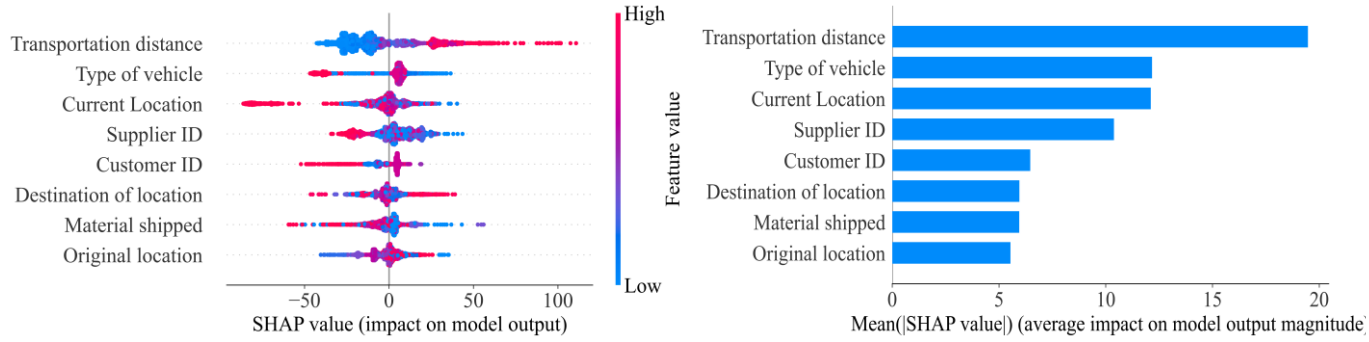
(b) Testing dataset

Fig. 8. Data distribution of arrival time data including actual, GB_PSO and Expected travel time (hours)



(a) Training dataset (b) Testing dataset

Fig. 9. Error of arrival time predicted by GB-PSO and Expected travel time of service compared with true time of arrival



(a) (b)

Fig. 10. Global SHAP value for interpreting input effect on the predicted arrival time

The proposed GB-PSO model delivers not only strong predictive performance but also meaningful operational insights for logistics stakeholders, particularly in emerging markets like India.

First, improved arrival time prediction enables more efficient scheduling and resource utilization. Accurate ETAs allow dispatchers and warehouse operators to better coordinate loading/unloading activities, reduce vehicle idling, and minimize labor idle time. This directly enhances throughput and cost-efficiency, especially in high-volume distribution centers. Second, the model's capacity to capture non-linear interactions among features such as vehicle type, supplier, and transportation distance enables dynamic routing and fleet allocation. Operators can use predictive outputs to reroute or reassign shipments in near-real-time, mitigating the impact of delays caused by traffic congestion or unforeseen disruptions. Third, the system supports service-level improvements for third-party logistics (3PL) providers. Accurate ETAs allow for tighter delivery windows and increased reliability in meeting contractual obligations. This not only reduces customer complaints but also strengthens long-term business relationships and brand credibility. Fourth, the insights generated by the model can support strategic decision-making under infrastructural and regulatory constraints. In India, where road conditions, traffic patterns, and regulatory practices vary significantly across regions, the model helps identify bottlenecks and optimize supply chain strategies accordingly. For example, specific vehicle types may be prioritized for certain routes based on predicted delay

patterns. Fifth, the GB-PSO framework offers potential value for policymakers and regulators. Authorities can integrate such models into freight monitoring systems to predict and manage congestion at urban entry points, enforce emissions or delivery-time regulations in low-emission zones, and plan logistics infrastructure investment based on data-driven risk factors.

To operationalize these benefits, stakeholders must invest in continuous data collection, integration with real-time traffic systems, and interface tools for visualizing predictions. Additionally, the interpretability of the GB-PSO model (via SHAP) makes it suitable for deployment in settings where model transparency is essential for decision trust.

In summary, the GB-PSO model bridges the gap between data complexity and actionable logistics intelligence, making it a practical decision-support tool for enhancing both tactical operations and long-term logistics planning.

6. Conclusions and perspectives

This study presents a robust framework for optimizing truck arrival time predictions using the GB-PSO model. Through hyperparameter tuning, the model achieves substantial improvements in predictive accuracy, outperforming both standard GB models and traditional expected travel time calculations. The GB-PSO achieved a higher predictive accuracy, with R = 0.7769 and RMSE = 48.05 hours on training data, outperforming the GB model (R = 0.7131, RMSE = 53.50 hours). Similarly, in testing data, GB-PSO demonstrated superior results (R = 0.6695, RMSE = 56.63 hours) compared to the GB model (R = 0.6564, RMSE = 57.53 hours).

Furthermore, GB-PSO significantly outperformed expected travel time estimations, as seen in the testing dataset, where expected travel time produced negligible correlation (R = 0.0002) and a high RMSE of 98.44 hours.

The SHAP analysis underscored the dominant influence of transportation distance and contextual features on model predictions. These findings affirm the effectiveness of GB-PSO in reducing errors and improving reliability in logistics. Future studies could explore real-time applications and broader datasets for enhanced operational insights.

This study highlights the GB-PSO model's potential as a foundation for developing digital soft tools in supply chain logistics, specifically for predicting and scheduling truck transportation.

Future work could involve experimenting with new optimization algorithms for hyperparameter tuning, such as Bayesian optimization or genetic algorithms, and improving data quality by incorporating real-time traffic, weather conditions, and shipment-specific details to further enhance predictive performance.

Conflict of Interest: The authors declare that there is no conflict of interest.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Availability of data and material: Data will be made available on request.

References

[1]. B. Medina-Salgado, E. Sánchez-DelaCruz, P. Pozos-Parra, J.E. Sierra. (2022). Urban traffic flow prediction techniques: A review. *Sustainable Computing: Informatics and Systems*, 35, 100739. https://doi.org/10.1016/j.suscom.2022.100739

[2]. Suardinata, R. Rusmi, M.A. Lubis. (2022). Determining Travel Time and Fastest Route Using Dijkstra Algorithm and Google Map. *SISTEMASI:Jurnal Sistem Informasi*, 11(2), 496-505. https://doi.org/10.32520/stmsi.v11i2.1836

[3]. V.Q. Tran, H.Q. Do. (2021). Prediction of California Bearing Ratio (CBR) of Stabilized Expansive Soils with Agricultural and Industrial Waste Using Light Gradient Boosting Machine. *Journal of Science and Transport Technology*, 1(1), 1-8. https://doi.org/10.58845/jstt.utt.2021.en.1.1.1-8

[4]. T.Q. Ngo, L.Q. Nguyen, V.Q. Tran. (2022). Predicting tensile strength of cemented paste

backfill with aid of second order polynomial regression. *Journal of Science and Transport Technology*, 2(4), 43-51. https://doi.org/10.58845/jstt.utt.2022.en.2.4.43-51

[5]. T.A. Pham, H.-L.T. Vu. (2022). Optimizing the architecture of the artificial neural network by genetic algorithm to improve the predictability of pile bearing capacity based on CPT results. *Journal of Science and Transport Technology*, 2(1), 1-8. https://doi.org/10.58845/jstt.utt.2022.en.2.1.1-8

[6]. V.L. Giap, T.A. Pham. (2024). Developing a Machine Learning Model for Predicting the Settlement of Bored Piles. *Journal of Science and Transport Technology*, 4(4), 95-109. https://doi.org/10.58845/jstt.utt.2024.en.4.4.95-109

[7]. M.V. Le, I. Prakash, D.D. Nguyen. (2023). Predicting Load-Deflection of Composite Concrete Bridges Using Machine Learning Models. *J Sci Transp Technol*, 3(4), 43-51. https://doi.org/10.58845/jstt.utt.2023.en.3.4.43-51

[8]. S. van der Spoel, C. Amrit, J. van Hillegersberg. (2017). Predictive analytics for truck arrival time estimation: a field study at a European distribution centre. *International Journal of Production Research*, 55(17), 5062-5078. https://doi.org/10.1080/00207543.2015.1064183

[9] R. Barlogis, A. Montarnal, C. Ouedraogo, D. Gourc. (2025). Predicting container intermodal transport arrival times: An approach based on IoT data. *Internet of Things*, 29, 101460. https://doi.org/10.1016/j.iot.2024.101460

[10] E. Žunić, D. Đonko, E. Buza. (2020). An Adaptive Data-Driven Approach to Solve Real-World Vehicle Routing Problems in Logistics. *Complexity*, 2020(1), 7386701. https://doi.org/10.1155/2020/7386701

[11] M. Li, J. Chen, G. Jiang, F. Li, R. Zhang, S. Gong, Z. Lv. (2025). TAS-TsC: A data-driven framework for Estimating Time of Arrival using Temporal-Attribute-Spatial Tri-space Coordination of truck trajectories. *Applied Soft Computing*, 178, 113214. https://doi.org/10.1016/j.asoc.2025.113214

[12] R. Wang, J. Li, R. Bai. (2023). Prediction and Analysis of Container Terminal Logistics Arrival Time Based on Simulation Interactive Modeling: A Case Study of Ningbo Port. *Mathematics*, 11(15), 3271. https://doi.org/10.3390/math11153271

[13] M.H. Nguyen, H.V. Nguyen, V.Q. Tran. (2024). Machine learning-based model for predicting arrival time of container ships. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 46(5-6), 11293-11310. https://doi.org/10.3233/JIFS-234552

[14]. E. Antypas, G. Spanos, A. Lalas, K. Votis, D. Tzovaras. (2024). A time-series approach for estimated time of arrival prediction in autonomous vehicles. *Transportation Research Procedia*, 78, 166-173. https://doi.org/10.1016/j.trpro.2024.02.022

[15]. Z. Mei, F. Xiang, L. Zhen-hui. (2018). Short-Term Traffic Flow Prediction Based on Combination Model of Xgboost-Lightgbm. 2018 International Conference on Sensor Networks and Signal Processing (SNSP). https://doi.org/10.1109/SNSP.2018.00069

[16]. A. Balster, O. Hansen, H. Friedrich, A. Ludwig. (2020). An ETA Prediction Model for Intermodal Transport Networks Based on Machine Learning. *Business & Information Systems Engineering*, 62, 403-416. https://doi.org/10.1007/s12599-020-00653-0

[17]. J.H. Friedman. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist*, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

[18]. R. Eberhart, J. Kennedy. (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43. https://doi.org/10.1109/MHS.1995.494215

[19]. J. Brownlee. (2018). A Gentle Introduction to k-fold Cross-Validation. MachineLearningMastery.com, <https://machinelearningmastery.com/k-fold-

cross-validation/> (accessed: 03/18/2023).
[20]. G. Shan. (2022). Monte Carlo cross-validation for a study with binary outcome and limited sample size. *BMC Medical Informatics and Decision Making*, 22, 270. https://doi.org/10.1186/s12911-022-02016-z
[21]. S.M. Lundberg, S.-I. Lee. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems 30 (NIPS 2017).*