

Article

Robust Truck Transit Time Prediction through GPS Data and Regression Algorithms in Mixed Traffic Scenarios

Adel Ghazikhani ^{1,2}, Samaneh Davoodipoor ¹, Amir M. Fathollahi-Fard ^{3,4,*} , Mohammad Gheibi ^{5,6} 
and Reza Moezzi ^{6,7} 

- ¹ Department of Computer Engineering, Imam Reza International University, Mashhad 178-436, Iran; aghazi@imamreza.ac.ir (A.G.); sa_davoodipoor@yahoo.com (S.D.)
- ² Big Data Lab, Imam Reza International University, Mashhad 178-436, Iran
- ³ Département d'Analytique, Opérations et Technologies de l'Information, Université de Québec à Montréal, 315, Sainte-Catherine Street East, Montreal, QC H2X 3X2, Canada
- ⁴ Department of Engineering Science, Faculty of Innovation Engineering, Macau University of Science and Technology, Macau 999078, China
- ⁵ Institute for Nanomaterials, Advanced Technologies, and Innovation, Technical University of Liberec, 461 17 Liberec, Czech Republic; mohammad.gheibi@tul.cz
- ⁶ Faculty of Mechatronics, Informatics and Interdisciplinary Studies, Technical University of Liberec, 461 17 Liberec, Czech Republic; reza.moezzi@tultech.eu
- ⁷ Association of Talent Under Liberty in Technology (TULTECH), 10615 Tallinn, Estonia
- * Correspondence: fathollahifard.amirmohammad@courrier.uqam.ca

Abstract: To enhance safety and efficiency in mixed traffic scenarios, it is crucial to predict freight truck traffic flow accurately. Issues arise due to the interactions between freight trucks and passenger vehicles, leading to problems like traffic congestion and accidents. Utilizing data from the Global Positioning System (GPS) is a practical method to enhance comprehension and forecast the movement of truck traffic. This study primarily focuses on predicting truck transit time, which involves accurately estimating the duration it will take for a truck to travel between two locations. Precise forecasting has significant implications for truck scheduling and urban planning, particularly in the context of cross-docking terminals. Regression algorithms are beneficial in this scenario due to the empirical evidence confirming their efficacy. This study aims to achieve accurate travel time predictions for trucks by utilizing GPS data and regression algorithms. This research utilizes a variety of algorithms, including AdaBoost, GradientBoost, XGBoost, ElasticNet, Lasso, KNeighbors, Linear, LinearSVR, and RandomForest. The research provides a comprehensive assessment and discussion of important performance metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2). Based on our research findings, combining empirical methods, algorithmic knowledge, and performance evaluation helps to enhance truck travel time prediction. This has significant implications for logistical efficiency and transportation dynamics.

Keywords: freight truck scheduling; cross-docking terminals; GPS data analysis; regression algorithms; traffic flow modelling; arrival time estimation

MSC: 37M10



Citation: Ghazikhani, A.; Davoodipoor, S.; Fathollahi-Fard, A.M.; Gheibi, M.; Moezzi, R. Robust Truck Transit Time Prediction through GPS Data and Regression Algorithms in Mixed Traffic Scenarios.

Mathematics **2024**, *12*, 2004. <https://doi.org/10.3390/math12132004>

Academic Editors: Xue-Ming Yuan, Hao Yu, Yong He and Ioannis G. Tsoulos

Received: 31 March 2024

Revised: 31 May 2024

Accepted: 27 June 2024

Published: 28 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The precise forecasting of heavy truck movements' timing and location is crucial for the efficient operation of cross-docking terminals, which are integral to the management of consumable goods, just-in-time inventory strategies, and rapid distribution across various locations [1–4]. Over the last three decades, numerous studies on transportation have focused on generating short-term forecasts for traffic variables, including traffic flow patterns, flow velocity, travel durations, and matchmaking distances [5–7]. Estimating

travel times for commercial vehicles in urban environments presents significant challenges due to limited observed data, numerous origin–destination pairs, and variations in travel times primarily caused by traffic-related delays [8–10].

Traditional data collection methods provide general insights but fall short of the in-depth analysis required for modern transportation dynamics [11,12]. The integration of GPS data in traffic monitoring and strategic planning has increased steadily over the past two decades, driven by significant technological advancements [13,14]. This integration is particularly relevant in regions like Iran, which serves as a critical junction for the transportation of goods between European nations and Central Asia, including Turkmenistan, Tajikistan, Kazakhstan, Kyrgyzstan, and Uzbekistan. Iran’s strategic location offers diverse opportunities for trade and transportation interactions.

The accurate utilization of GPS data is essential for understanding and predicting truck traffic behavior in complex scenarios. The numerous border crossings and maritime gateways in Iran facilitate the intricate process of transporting goods across the country, providing trucks with multiple entry and exit options. This predictive model extends beyond logistics, impacting border traffic forecasting by leveraging the consistent trends observed in the trucking industry, where trucks return to the country with or without cargo after delivering shipments abroad [15].

This paper presents a comprehensive framework, combining GPS data with advanced machine learning regression algorithms to forecast travel times for freight trucks accurately. The study explores how to maximize the efficiency of goods in transit through Iran’s strategic location, offering valuable insights for logistics planning, border traffic estimation in strategic trade contexts, and cross-dock management. By examining the intricacies of trucking operations, including border crossings, road transport, and maritime gateways, the research enhances our understanding of these processes.

Key contributions of this study include:

- Introducing a novel method that integrates GPS data with advanced machine learning regression algorithms to forecast freight truck travel times accurately.
- Providing valuable insights for logistics planning and border traffic estimation through an exploration of goods’ transit efficiency in Iran’s strategic trade corridors.
- Enhancing the depth of analysis by examining various factors influencing trucking operations, such as border crossings and maritime gateways.
- Offering a unique and detailed visualization of truck trajectories, grouped by date and time, to improve the understanding of movement patterns.
- Establishing a foundation for informed decision-making in transportation, logistics, and cross-docking terminals by advancing knowledge in travel time prediction and truck traffic flow.

The structure of this paper is as follows: Section 2 presents a literature review to identify research gaps forming the basis of our contributions. Section 3 explains the research method, which involves advanced machine learning regression algorithms. Section 4 provides detailed findings, potential scenarios, and data analysis related to predicting journey times and truck traffic trends. Section 5 discusses practical recommendations and managerial perspectives on cross-docking terminals, transportation, and logistics planning. Finally, Section 6 concludes the research by summarizing the findings, constraints, and potential avenues for future research.

2. Literature Review

Case studies from China, the US, and Europe underscore the significance of accurate truck scheduling and urban planning, particularly for cross-docking terminals [16,17]. Various prediction techniques have been employed in truck scheduling research, with recent studies focusing on data mining and machine learning methods [18–20]. Below, we examine key contributions in this field.

One of the earliest studies, conducted by Zhao and Goodchild [21], investigated port discharge—a critical aspect of intermodal maritime systems affecting supply chain

efficiency. By leveraging truck entry data, they improved port discharge and operational effectiveness. Their research included a reliability assessment of travel time changes across drainage networks, an analysis of truck routing choices, and a method to predict 95% confidence intervals for travel times between origin–destination pairs using GPS data. Morgul et al. [22] assessed the role of GPS data in transportation planning and proposed an integrated strategy to use robust GPS data for predicting travel times for commercial vehicles. Their study showed that the travel times derived from taxi GPS data closely matched those of trucks, suggesting the potential for scaling taxi GPS data to enhance truck travel time insights. Despite limited truck GPS data, taxi GPS data provided citywide travel time estimations, showcasing an innovative synergy between existing data sources and effective estimation strategies. Moniruzzaman et al. [1] utilized one-month volume data from remote microwave traffic sensors and one-year GPS data to develop two sets of artificial neural network (ANN) models. These models predicted short-term truck volumes at a specific crossing, bridge clearance times, and traversal durations. Separate ANN models were trained for volume prediction using a multi-layer feedforward neural network with backpropagation. The predicted crossing times from the ANN models showed a strong correlation with observed values, confirmed by evaluation indices, demonstrating the robust predictive capability of the models.

Jiang [23] focused on predicting bus transit times using bus GPS data and artificial neural networks. Accurate predictions were essential for urban transportation planning and optimizing bus schedules. Jiang introduced three predictive models for travel time estimation, each based on a three-layer neural network architecture. The first model predicted total travel time using calculated features from bus GPS data. The second model utilized information from preceding buses to predict segment travel times, and the third combined segment predictions to estimate the total route travel time. Wang et al. [24] presented an innovative method for predicting truck traffic flow using sampled GPS data within road networks. They employed a two-stage framework: expansion and prediction. The expansion phase used a piecewise constant coefficient method to align the sampled and actual truck flows, considering road gradients and traffic flow magnitudes. The prediction phase applied Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) methodologies, significantly improving prediction accuracy. In recent research, Demissie and Kattan [25] explored large-scale GPS data streams to estimate truck origin–destination flows. They developed an exploratory framework to identify significant events such as truck halts and trips, supported by the Pearson correlation coefficient and an entropy measure. This approach facilitated the comparative analysis of truck movement patterns and identified potential shifts in truck travel dynamics over a year. The researchers then used a multinomial logistic model to develop destination choice models across five time intervals.

Table 1 summarizes the current data mining and machine learning methodologies based on the referenced studies and research environment. Notably, none of these studies combine GPS data with machine learning techniques, such as AdaBoost, GradientBoost, XGBoost, ElasticNet, Lasso, KNeighbors, Linear, LinearSVR, and RandomForest for predicting the arrival times of heavy trucks in transportation systems and cross-docking terminals.

To deeper examine into the landscape of heavy truck arrival time prediction, we introduce a comprehensive Sankey diagram, illustrated in Figure 1. This diagram provides an insightful visual representation of prominent research trends at the confluence of transportation, machine learning, and predictive analysis. Leveraging data sourced from the Scopus database since July 2023, this diagram intricately synthesizes the interrelationships among countries, keywords, and primary sources within this domain.

The diagram distinctly accentuates three pivotal countries that have emerged as influential drivers in shaping this research terrain: China, the United States, and India. These nations have consistently exhibited remarkable prominence in propelling advancements at the nexus of transportation and machine learning. Moreover, the diagram informs three predominant keywords that have captured substantial attention within the discourse: “*machine learning*”, “*deep learning*”, and “*prediction*”. These keywords encapsulate the quintessence of

research endeavors focused on leveraging computational intelligence to elevate predictive modelling and analytical capacities within the ambit of transportation systems. Significantly, the Sankey diagram also spotlights the primary sources that serve as critical conduits for disseminating cutting-edge research outcomes in this domain. Noteworthy among these sources are the journals of *“Transportation Research Part C: Emerging Technologies”*, *“IEEE Access”*, and the *“IEEE Transactions on Intelligent Transportation Systems”*.

Table 1. Review of existing machine learning and data mining techniques in this field.

Data Mining and Machine Learning Methods	Reference
Regression and Classification Tree	Pani et al. [26]
Support Vector Machine (SVM), Regression, and Mixed Integer Programming	Bhattacharya et al. [27]
Gradient Boosting Regression (GBR), and Decision Tree (DT)	Li and Bai [28]
KNeighbors, DT, SVM, Ensemble learning	Van der Spoel et al. [29]
Bayesian Network using Fuzzy rule	Salleh et al. [30]
Logistics Regression	Alcoba, and Ohlund, [31]
Random Forest (RF), Non-Linear and Linear SVM, and ANN	Barbour et al. [32]
Distribution, Spatiotemporal data mining, Pattern, and Social Representation and Relation analysis	Wu et al. [33]
Backpropagation, Regression and Classification Tree, and RF	James et al. [34]
Segment-Based Ordinary Kriging and Regression Kriging for Spatial Prediction	Yu et al. [35]
RF, GBR and Linear Regression Trees	Balster et al. [36]
SVM, Adaptive Boosting, and Extremely Randomized Tree	Servos et al. [37]
Graph Neural Network (GNN)	Verma et al. [38]
Dynamic Graph Convolutional Recurrent Imputation Network (DGCRIN)	Li et al. [10]
SVM, DT, ANN, and Gaussian process regression	Liu et al. [39]
RF, GBR, Bagging, and WaveNet	Antamis et al. [40]
RF, GBR, Natural GBR, Extreme GBR and Bagging	Valatsos et al. [41]

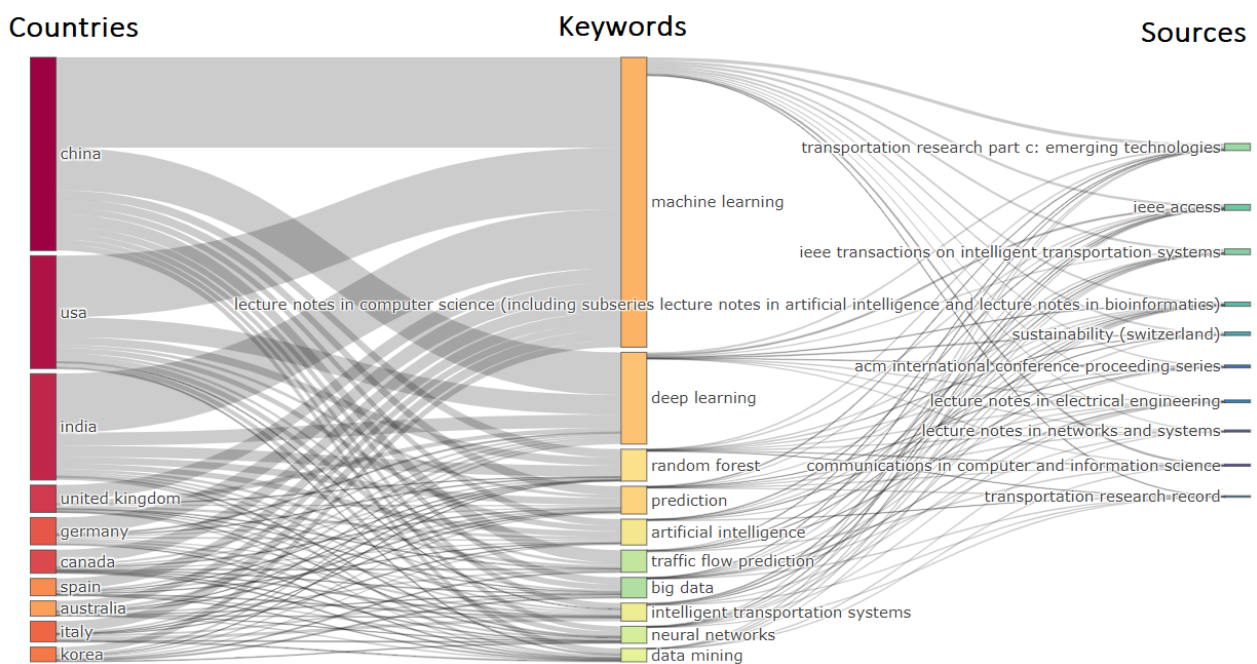


Figure 1. Sankey diagram for highlighting the main countries, keywords, and journal sources in the field.

Realizing the critical importance of predicting vehicle timing and location on both domestic and international roads, particularly in regions like Iran where research in this area is scarce, we initiated a comprehensive investigation into this problem. While extensive studies have focused on truck scheduling, urban planning, and the application of machine learning in transportation, there remains a significant gap in accurately predicting heavy truck transit times using GPS data combined with advanced machine learning algorithms. Existing studies have largely utilized traditional data sources or applied machine learning techniques to related but distinct problems, such as bus transit time prediction or using taxi GPS data for estimating truck travel times. None have comprehensively integrated GPS data with a wide array of machine learning algorithms specifically tailored for predicting heavy truck arrival times in cross-docking terminals and complex transportation networks.

This paper addresses this gap by introducing a novel approach that leverages GPS data alongside advanced regression algorithms, including AdaBoost, GradientBoost, XGBoost, ElasticNet, Lasso, KNeighbors, Linear, LinearSVR, and RandomForest. To facilitate accurate prediction, the geographic coordinates are converted into real addresses, followed by a normalization process using the MinMax method to account for differences in time and place. Data segmentation then categorizes each route based on parameters such as date and time, allowing for a detailed visualization of each truck's full path and location on specific dates through scatter plots. This approach not only enhances the visualization of truck trajectories but also connects their movement paths chronologically.

For evaluation, the study employs the k-fold method, utilizing 80% of the dataset for training and 20% for thorough testing. The integration of the K-Nearest Neighbors (KNN) algorithm with the Leave One Out technique further refines the evaluation framework, streamlining both the training and testing phases. By doing so, this research not only improves the accuracy of truck travel time predictions but also provides valuable insights for logistics planning, border traffic estimation, and the optimization of cross-dock operations. This comprehensive integration of empirical data and machine learning techniques offers a robust framework for enhancing the efficiency and reliability of truck transit time predictions, thereby filling a critical void in the existing body of research.

3. Methodology and Empirical Applications

Accurate freight truck traffic flow prediction is crucial for solving problems in urban planning, truck scheduling, and cross-docking terminals. These predictions help urban planners optimize traffic management, allocate resources, and design road infrastructure for freight truck movements, ultimately reducing congestion and accidents and improving traffic safety. In logistics, predictive truck scheduling enables companies to streamline operations by optimizing delivery schedules, reducing idle time, and enhancing fleet efficiency. This leads to reduced costs and improved sustainability. Cross-docking terminals, which are critical for time-sensitive goods and just-in-time inventory management, benefit from accurate predictions by optimizing inbound and outbound shipment coordination, minimizing wait times, and ensuring smooth goods flow. Our research provides significant benefits for urban infrastructure planning, logistics operations, and cross-docking terminal efficiency, extending beyond algorithmic advancements (Figure 2).

Within this paper, we have considered the applied methods, which include both the use of regression algorithms (Section 3.1) and the smart selection of the right time series techniques (Section 3.2). After that, we go into more detail about the complicated steps that went into making our complete algorithm (Section 3.3). Figure 3 shows the conceptual roadmap of our research methodology.

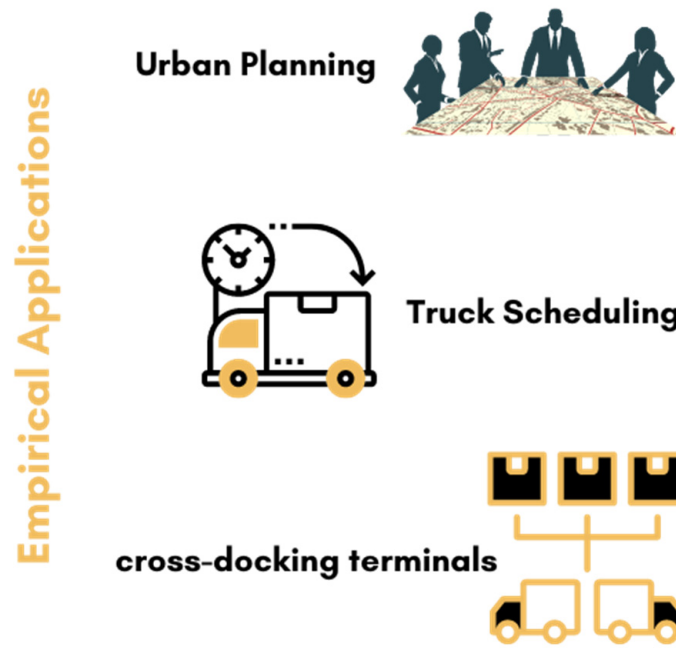


Figure 2. Empirical application of this research.

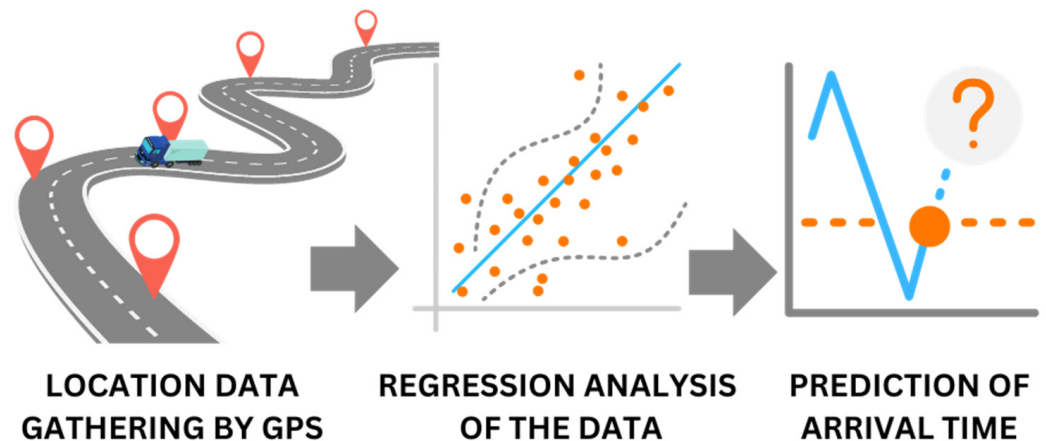


Figure 3. Conceptual diagram of this research.

3.1. Data Preprocessing

We began by collecting raw GPS data from freight trucks navigating various routes in mixed traffic scenarios. The initial step involved cleaning the data to address outliers and missing values. We then normalized the features using the MinMax method to ensure equal contribution during model training. Additionally, we engineered features such as road type and truck characteristics to enhance predictive performance.

For feature selection, we employed correlation analysis, feature importance assessment, and dimensionality reduction techniques to identify the most informative subset of features. This selection process was guided by domain-specific insights, ensuring the inclusion of factors known to influence truck transit time.

By systematically preprocessing the data, we ensured that our models were trained on high-quality, relevant information, thereby enhancing the accuracy and reliability of our predictions. The following sections provide detailed descriptions of the regression algorithms employed and the selection of time series techniques, followed by the intricate steps of our algorithm development process.

3.2. Regression

One of the most popular methods to model data is linear regression, which has a robust and simple mathematical foundation [28]. This method is beneficial for identifying linear correlations between two variables, enabling the estimation of one variable's value based on another. In this context, a linear relationship indicates that a change in one variable directly impacts the other [42]. The independent variable, which influences change, is a key term in this model.

A scatter plot is used to depict the relationship between two variables by plotting one against the other [43]. If the graph forms a straight line, it indicates a linear relationship between the variables [28]. Four conditions must be satisfied to validate the proposed model that defines the link between the data and the dependent variable:

1. **Linearity:** the relationship between the variables should manifest as a linear pattern on the scatter plot.
2. **Independence:** each data point should remain independent, without any connection or reliance on others.
3. **Homoscedasticity:** the variability in the dependent variable should remain consistent across the range of the independent variable, with data points scattering similarly around the regression line for all values of the independent variable.
4. **Normality:** The residuals, representing the differences between observed and predicted values, should follow a normal distribution. This condition is crucial for hypothesis testing and confidence interval calculations.

Ensuring these conditions are met enhances confidence in the accuracy of the proposed model, establishing a strong foundation for further investigation.

3.3. Time Series

The data windowing approach is also utilized in data prediction. Sometimes, a time series can be viewed as a regression issue, where "time" serves as the independent variable [44]. The primary objective of time series analysis is to forecast the future value of the dependent variable. Stationarity, a crucial quality in time series analysis, allows for effective data analysis [45].

Time windows are essential for determining the long-term movement of a truck or vehicle. Forecasting time depends on the location parameter (Loc). In our research, we examine several time frames, such as 1, 3, 5, 7, 9, and 11 units. By carefully analyzing outcomes from different time frame configurations, we gain a deeper understanding of the effectiveness of our predictive models.

3.4. Full Algorithmic Framework

Figure 4 displays the flowchart of the entire procedure included in our comprehensive method. The process begins with loading and reading raw data. Geographical coordinates must be converted into a uniform numerical representation due to the various data formats used. Data grouping is performed to enable the creation of a scatter plot. Simultaneously, data undergoes time windowing to enhance the quality of our study.

The core phase, moving ahead, involves the machine learning process of training and testing. This phase concludes with the development of a machine learning model that is carefully designed using regression algorithms. We improve our ability to forecast by utilizing the k-fold approach to enhance the accuracy of our predictions.

Here is an overview of each ensemble algorithm utilized in our research:

- **AdaBoost (Adaptive Boosting):** AdaBoost is an ensemble learning method that combines multiple weak classifiers to create a strong classifier. It iteratively adjusts the weights of incorrectly classified instances to focus on the difficult-to-classify samples. Each weak classifier is trained sequentially, and its predictions are combined using a weighted majority vote.

- GradientBoosting: Gradient boosting is a machine learning technique that builds a strong predictive model by sequentially fitting new models to the residuals of the previous models. It minimizes a loss function by iteratively adding new decision trees, where each tree is trained to correct the errors of the previous ones.
- XGBoost (Extreme Gradient Boosting): XGBoost is an optimized implementation of gradient boosting that offers improvements in speed and performance. It incorporates features such as parallelized tree construction and hardware optimization to achieve state-of-the-art results in many machine learning tasks.
- ElasticNet: ElasticNet is a regularization technique that combines the penalties of both the L1 (Lasso) and L2 (Ridge) regularization methods. It is used to address multicollinearity and perform feature selection by encouraging sparse coefficients while still allowing for correlated predictors.
- Lasso (Least Absolute Shrinkage and Selection Operator): Lasso is a linear regression method that performs both variable selection and regularization by adding a penalty term to the absolute values of the regression coefficients. It encourages sparsity in the model by shrinking some coefficients to zero, effectively performing feature selection.
- KNeighbors (K-Nearest Neighbors): KNeighbors is a non-parametric algorithm used for classification and regression tasks. It predicts the output of a data point by averaging the target values of its k nearest neighbors in the feature space.
- Linear Regression: Linear regression is a simple linear model that predicts the target variable as a linear combination of the input features. It is widely used for regression tasks when the relationship between the features and the target variable is assumed to be linear.
- LinearSVR (Linear Support Vector Regression): LinearSVR is a variant of support vector regression that uses a linear kernel function to map the input features into a higher-dimensional space. It aims to find a hyperplane that best fits the training data while maximizing the margin.
- RandomForest: RandomForest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the average prediction of the individual trees. It improves upon the decision tree algorithm by reducing overfitting and increasing robustness.

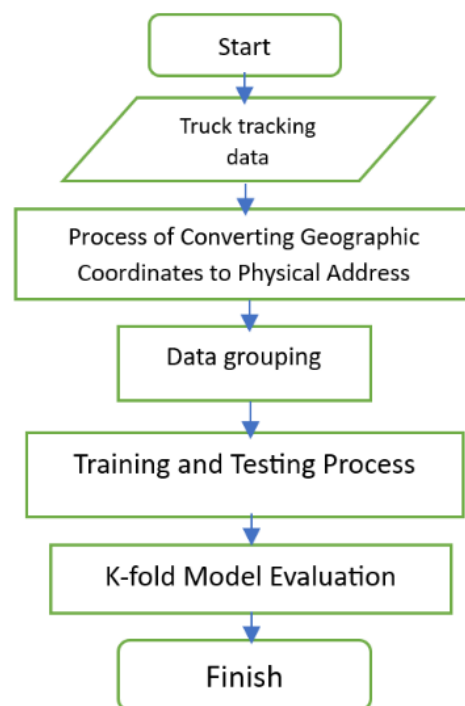


Figure 4. Flowchart for the full algorithm.

3.5. Model Validation

Ensuring the dependability and accuracy of our predictive models is crucial for the strength of our research. The validation phase evaluates the accuracy of our created models in forecasting truck transit times by analyzing the collected GPS location data.

To validate our models, we adopted a rigorous approach, employing the following key methodologies:

- **Dataset Splitting:** The dataset was divided into training and testing sets. The training set was utilized for model training, while the testing set remained unseen during the training phase, allowing us to evaluate the model's generalization to new, unseen data.
- **Evaluation Metrics:** We employed standard evaluation metrics to assess the performance of our models. Key metrics include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2). These metrics provide a comprehensive understanding of the model's accuracy and predictive capabilities.
- **Cross-Validation:** To further enhance the robustness of our models, we implemented cross-validation techniques. This involved dividing the dataset into multiple folds, training the model on subsets of the data, and evaluating its performance across different subsets. This approach helps mitigate overfitting and ensures the model's consistency across various data partitions.
- **Comparison with Baseline:** We compared the performance of our models against a baseline model, such as a simple linear regression or a basic algorithm. This comparison provides insights into the added value of our proposed approach. Through these validation procedures, it aims to ascertain the effectiveness of our models in accurately predicting truck transit times based on GPS location data. The results of this validation process will be presented in the next section.

4. Results

This research is based on a detailed dataset consisting of 14 Excel files specifically organized to store truck tracking data from 17 June 2021, to 3 August 2022. Consistently, each file contains an astounding aggregation of 1,048,576 records. This data includes locations in the routes of the trucks, from different cities in Iran to cities in Turkey, Turkmenistan, Tajikistan and Afghanistan (Figure 5).

The dataset contains crucial attributes such as truck number, x-coordinate position, y-coordinate position, Gregorian date, and time. The sole concern is converting the coordinates into geographic locations, for which Section 4.1 provides a detailed explanation.

Regression techniques are well-suited for estimating truck trip times due to their compatibility and effectiveness when used in combination. Utilize the extensive truck monitoring data spanning a full year to enable the platform to generate very precise forecasts regarding the arrival times and locations of trucks. This section delves into the insights obtained by utilizing GPS data to predict the travel time of a truck. A comprehensive analysis is being compiled to provide various perspectives on the outcomes of predictive modeling. Section 4.1 explains the process of converting raw geographic coordinates into actual addresses, a crucial step in the research. The narrative transitions to Section 4.2, which discusses the methodical categorization of the data into several groups. Section 4.3 presents a visual tale using scatter plots, while Section 4.4 breaks down the structure of the data. Section 4.5 outlines the challenging processes of data training and testing as the trip progresses. Section 4.6 tests the proposed method in various scenarios, resulting in a comprehensive evaluation across multiple settings.



Figure 5. Geographical locations covered by truck routes connecting cities in Iran with cities in Turkey, Turkmenistan, Tajikistan, and Afghanistan (In this map, there are some Persian text as follows: Tehran: تهران, Riyadh: الرياض, Jeddah: جدة, and Cairo: القاهرة).

4.1. Process of Transforming Geographic Coordinates into Physical Addresses

Geographical coordinates are essential in mapping vehicles to convert an address into specific latitude and longitude map points. Geocoding is the process of converting a physical address into its corresponding geographical coordinates. Reverse geocoding converts geographic coordinates into physical addresses. Geocoding assigns unique coordinates to each address on the map. The dynamic change is facilitated by a Locator service, which is simplified by the Locator class.

When geocoding, several input parameters must be considered. The Address object is crucial for matching addresses during the geocoding process. During reverse geocoding, the Point object holds more significance than the Address object. The Address object within the geocoding framework communicates with the geocoding service. The geocoding service returns an Address Candidate object that includes the matched address and its matching map point. Subsequently, this map point becomes the focal point on the map.

The geographical coordinates of address-linked sites have been meticulously assigned, as evident from a detailed examination of Table 2. Assigning a distinct identifying field to each vehicle is crucial in this architecture, as it simplifies monitoring the routes they have travelled.

Table 2. Physical aspects.

Index	Id	x1	y1	x2	y2	Loc1	Loc2	Date	des Second
0	0	44.38008	39.401642	44.380080	39.401642	24,227.816042	24,227.816042	2021-06-1704:47:58	900.0
1	0	44.38008	39.401642	44.380080	39.401642	24,227.816042	24,227.816042	2021-06-1705:02:58	900.0
2	0	44.38008	39.401642	44.380080	39.401642	24,227.816042	24,227.816042	2021-06-1705:17:58	900.0
3	0	44.38008	39.401642	44.380080	39.401642	24,227.816042	24,227.816042	2021-06-1705:32:58	900.0
4	0	44.38008	39.401642	44.380080	39.401642	24,227.816042	24,227.816042	2021-06-1705:47:58	900.0
...
6,995,098	4896	25.986813	43.974322	25.986813	43.974322	20,921.600716	20,921.600716	2022-07-2810:38:47	901.0
6,995,099	4896	25.986813	43.974322	25.986813	43.974322	20,921.600716	20,921.600716	2022-07-2810:40:27	100.0
6,995,100	4896	25.986813	43.974322	25.986813	43.974322	20,921.600716	20,921.600716	2022-07-2810:42:09	102.0
6,995,101	4896	25.986813	43.974322	25.986813	43.974322	20,921.600716	20,921.600716	2022-07-2810:57:11	902.0
6,995,102	4896	25.986813	43.974322	25.986813	43.974322	20,921.600716	20,921.600716	2022-07-2811:03:30	379.0
6,995,103 rows × 9 columns									

4.2. Data Grouping

An essential aspect of our study involves categorizing each route and linking it to precise time markers, which include both date and time. This thorough classification is beneficial for locating and identifying all the locations visited by each truck. The segmentation provides a precise depiction of the whereabouts of each vehicle on specific dates. Please review Table 3 for complete information on the locations and positioning of each vehicle at various periods.

Table 3. Data grouping.

Number of Data	Date	Id	Loc	Time	Time_Cumsum
0	02021-06-1701:00:00	0	24,228.181138	45,000.0	45,000.0
1	12021-06-1702:00:00	0	24,228.181138	3600.0	48,600.0
2	22021-06-1703:00:00	0	24,228.181138	3600.0	52,200.0
3	32021-06-1704:00:00	0	24,228.434423	21,522.0	73,722.0
4	42021-06-1705:00:00	0	24,227.816042	3600.0	77,322.0
...
1,158,510	2022-07-2906:00:00	4896	20,692.553074	3118.0	4,806,768.0
1,158,511	2022-07-2907:00:00	4896	20,698.443486	3545.0	4,810,313.0
1,158,512	2022-07-2908:00:00	4896	20,802.591637	46,731.0	4,857,044.0
1,158,513	2022-07-2909:00:00	4896	20,916.917395	122,581.0	4,979,625.0
1,158,514	2022-07-2912:00:00	4896	20,884.905480	87,301.0	5,066,926.0
1,158,515 rows × 5 columns					

4.3. Scatter Plots

Visualizing data graphically is a strategic approach to clarify and reveal the intricate patterns, trends, and relationships within the data. We utilized the elements id, Loc, and time to create a three-dimensional graph to analyze the intricate network of vehicle paths. This research on dynamic visualization provides us with the means to go beyond static images and experience the immersive capabilities of a 3D platform. Figure 6 visually represents the multidimensional trip and acts as evidence of this visual narrative.

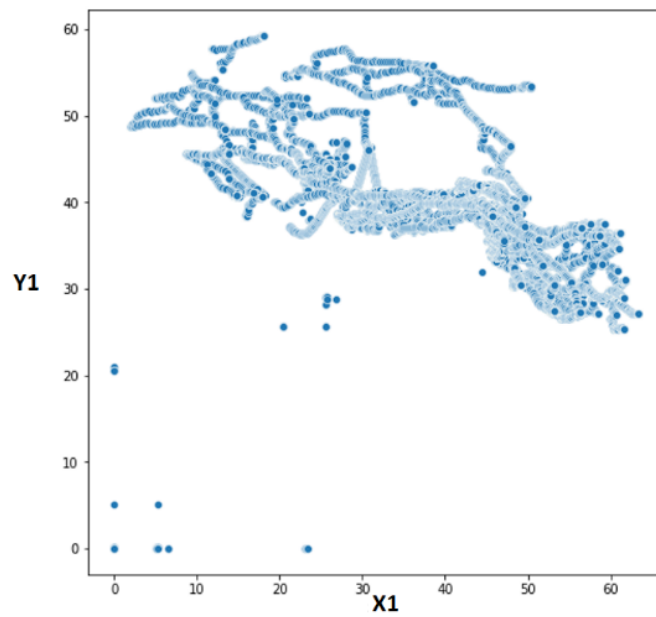


Figure 6. Data scatter diagram for track of trucks.

Strategic color usage enhances the graphical scene with deeper significance. The color palette is an effective tool for categorizing and labelling different data classes. Cars are categorized based on a unique identifier (id) and each car is associated with a particular color, occupying a specified location. Vehicles are arranged using a chromatic orchestration, with each one vividly colored to represent a specific genre. This division provides energy and visual differentiation to the graphical representation. Please refer to Figure 7 for a colorful and compelling graphic portrayal of this image.

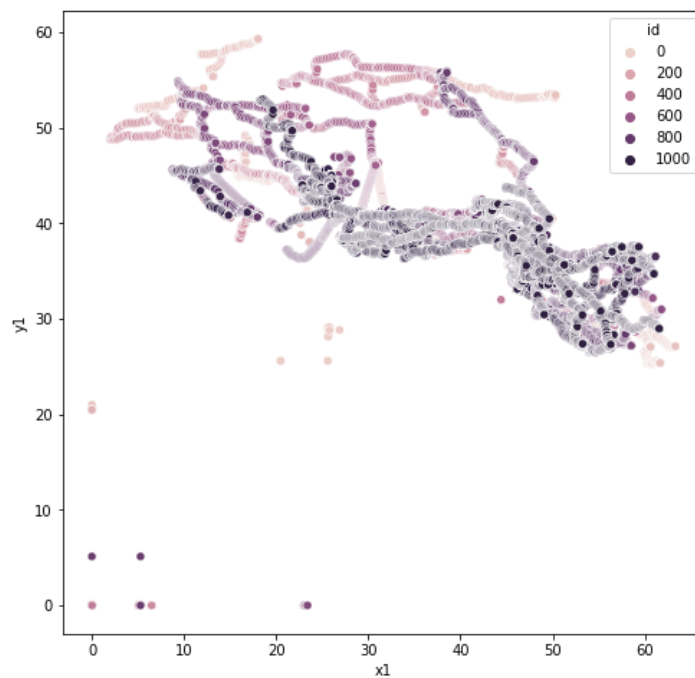


Figure 7. Color diagram of data dispersion for trucks routing.

Figure 8 illustrates the evolution of a vehicle’s trajectory over time. This graphic provides a detailed representation of the vehicle’s chronological voyage by clearly illustrating its path and movements. Figure 8 illustrates the complete trajectory of a vehicle, from the

beginning of its voyage to the conclusion of its route. Overall, these stats provide more than just numerical details, and offer informative comments.

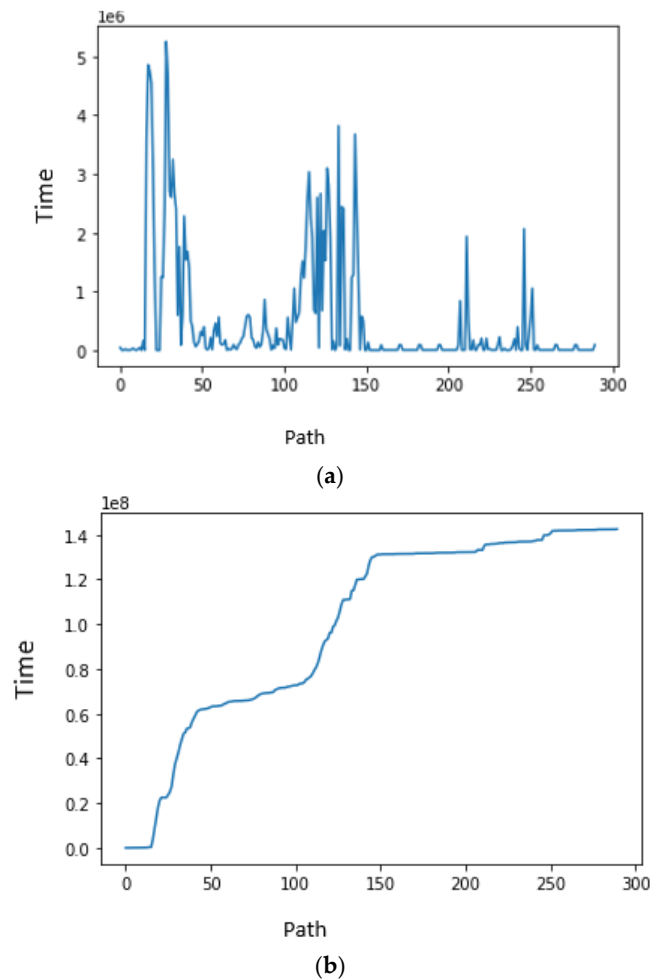


Figure 8. Analyses on the travel path: (a) Travel path of a vehicle with a given ID based on time. (b) Changing a truck's route according to time.

4.4. Data Structure

This research is based on a large dataset consisting of 14 Excel files. This data repository stores intricate truck tracking data from 17 June 2021 to 3 August 2022, spanning a complete year. There are five crucial characteristics in the dataset: (1) truck number; (2) x- and y-coordinates of the place; (3) date; and (4) time.

Once we have carefully developed our machine learning regression model to address the issue, the subsequent crucial stage is evaluation. It is crucial to thoroughly evaluate the model's performance by examining a well-chosen set of parameters from this perspective. The paper will discuss key parameters, such as MSE, RMSE, MAE, and R^2 . Table 4 meticulously documents the results of this assessment, which is the crucial aspect of validation.

In the study, we employed an ensemble of algorithms to predict truck transit time. These algorithms include AdaBoost, GradientBoosting, XGBoost (Extreme Gradient Boosting), ElasticNet, Lasso (Least Absolute Shrinkage and Selection Operator), KNeighbors (K-Nearest Neighbors), Linear Regression, LinearSVR (Linear Support Vector Regression), and RandomForest.

AdaBoost iteratively adjusts the weights of incorrectly classified instances to focus on difficult-to-classify samples. GradientBoosting sequentially fits new models to the residuals of the previous models, minimizing a loss function by adding decision trees that correct the errors of the preceding ones. XGBoost, an optimized implementation of gradient boosting,

incorporates features like parallelized tree construction and hardware optimization to achieve state-of-the-art results. ElasticNet combines L1 and L2 regularization to address multicollinearity and perform feature selection. Lasso performs variable selection and regularization by adding a penalty term to the absolute values of regression coefficients, encouraging sparsity in the model. KNeighbors predicts the output of a data point by averaging the target values of its k nearest neighbors. Linear Regression predicts the target variable as a linear combination of input features and is used when the relationship is assumed to be linear. LinearSVR uses a linear kernel function to find a hyperplane that best fits the training data while maximizing the margin. RandomForest constructs multiple decision trees during training and outputs the average prediction of individual trees, reducing overfitting and increasing robustness.

Table 4. Comparison of algorithms based on metrics.

	Algorithm	MSE	RMSE	MAE	R ²
0	KNeighborsRegressor	0.000050	0.007063	0.004540	0.430742
1	LinearRegression	0.000052	0.007212	0.004717	0.406420
2	LinearSVR	0.000054	0.007359	0.004423	0.381872
3	RandomForestRegressor	0.000052	0.007223	0.004656	0.404593
4	AdaBoostRegressor	0.000101	0.010039	0.008029	−0.150267
5	GradientBoostingRegressor	0.000051	0.007151	0.004599	0.416330
6	XGBRegressor	0.000051	0.007129	0.004598	0.419922
7	Lasso	0.000088	0.009361	0.006873	−0.000002
8	ElasticNet	0.000088	0.009361	0.006873	−0.000002

4.5. Training and Testing Process

Our machine learning model has undergone extensive training, utilizing 80% of the dataset just for this purpose. The meticulous training occurred in the context of Figure 9, illustrating the convergence of data and learning. The primary concept of this approach is to evaluate the performance of a model on data subsets that it has not been exposed to previously. After completing the training phase, the main validation step was initiated, utilizing the final 20% of the data. Proficiently isolating this section for examination is an excellent way to evaluate the effectiveness of a model by contrasting it with unfamiliar data. The model can assess its capacity to apply acquired knowledge to unfamiliar scenarios by precisely categorizing items. This demonstrates its potential for application in new environments.

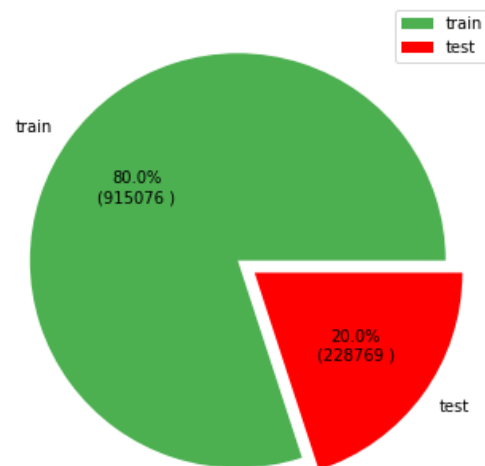


Figure 9. Training and testing data.

4.6. Evaluation of Algorithms

The evaluation, a crucial component of algorithmic application, is conducted based on the carefully prepared and trained data from the preceding section. This section demonstrates how our suggested algorithms engage with real-world data through a range of scenarios and methodically selected criteria.

- **First Scenario**

We investigate truck movement prediction in the first scenario and analyze the impact of different window sizes on the predictive accuracy. Window sizes of 1, 3, 5, 7, 9, and 11 play a significant role in enhancing prediction accuracy, with each size contributing distinct nuances. Our inquiry progresses through various stages with precise attention to detail, thoroughly examining the complexity of predicting truck movements within a specific range.

Table 5 displays a curated set of results, showing the effects of using our nine regression techniques with different window sizes. We have meticulously gathered many factors that affect prediction accuracy to analyze the interaction between algorithms and the changes in window sizes.

Table 5. Prediction results for different windows.

Window Size	Algorithms	MSE	RMSE	MAE	R ²
1	KNeighborsRegressor	0.0028	0.0528	0.0352	0.409
1	LinearRegression	0.0029	0.0535	0.0366	0.3912
1	LinearSVR	0.0029	0.054	0.0356	0.3814
1	RandomForestRegressor	0.003.2	0.0565	0.0375	0.3224
1	AdaBoostRegressor	0.0052	0.0724	0.0601	−0.1129
1	GradientBoostingRegressor	0.0028	0.0527	0.0355	0.4108
1	XGBRegressor	0.0028	0.0525	0.0352	0.4156
1	Lasso	0.0047	0.0686	0.0515	0
1	ElasticNet	0.0047	0.0686	0.0515	0
3	KNeighborsRegressor	0.0026	0.05.15	0.0340	0.4399
3	LinearRegression	0.002.7	0.05.17	0.0350	0.4338
3	LinearSVR	0.0028	0.0533	0.0326	0.398.9
3	RandomForestRegressor	0.002.7	0.05.2	0.0347	0.4275
3	AdaBoostRegressor	0.003.5	0.0594	0.0454	0.2543
3	GradientBoostingRegressor	0.0026	0.05.11	0.0340	0.4486
3	XGBRegressor	0.0026	0.0506	0.0335	0.458
3	Lasso	0.0047	0.0687	0.0514	-
3	ElasticNet	0.0047	0.0687	0.0514	0
5	KNeighborsRegressor	0.0026	0.0512	0.0338	0.424
5	LinearRegression	0.0026	0.0514	0.0347	0.4197
5	LinearSVR	0.0028	0.053.1	0.0327	0.3813
5	RandomForestRegressor	0.0026	0.05.13	0.0342	0.4236
5	AdaBoostRegressor	0.0034	0.0584	0.0448	0.2508
5	GradientBoostingRegressor	0.0026	0.0506	0.0337	0.4387

Table 5. Cont.

Window Size	Algorithms	MSE	RMSE	MAE	R ²
5	XGBRegressor	0.0025	0.05.01	0.0332	0.4494
5	Lasso	0.0046	0.0675	0.0509	-
5	ElasticNet	0.0046	0.0675	0.0509	-
7	KNeighborsRegressor	0.0027	0.05.18	0.0339	0.4227
7	LinearRegression	0.0027	0.05.15	0.0346	0.4298
7	LinearSVR	0.0028	0.0533	0.0326	0.3885
7	RandomForestRegressor	0.0026	0.05.11	0.0341	0.4387
7	AdaBoostRegressor	0.004	0.063.2	0.0514	0.1419
7	GradientBoostingRegressor	0.0026	0.0507	0.0335	0.4483
7	XGBRegressor	0.0025	0.05	0.0330	0.4628
7	Lasso	0.0047	0.0682	0.0510	0
7	ElasticNet	0.0047	0.0682	0.0510	0
9	KNeighborsRegressor	0.0027	0.0519	0.0339	0.4199
9	LinearRegression	0.0026	0.0512	0.0343	0.4343
9	LinearSVR	0.0028	0.0525	0.0324	0.4059
9	RandomForestRegressor	0.0026	0.0507	0.0338	0.4458
9	AdaBoostRegressor	0.0045	0.067	0.0553	0.0336
9	GradientBoostingRegressor	0.0025	0.0505	0.0333	0.4508
9	XGBRegressor	0.0025	0.0498	0.0328	0.4659
9	Lasso	0.0046	0.0681	0.0508	-0.0001
9	ElasticNet	0.0046	0.068.1	0.0508	-0.0001
11	KNeighborsRegressor	0.002.7	0.05.16	0.0339	0.4133
11	LinearRegression	0.0026	0.0508	0.0342	0.4311
11	LinearSVR	0.0037	0.0609	0.0497	0.1835
11	RandomForestRegressor	0.0025	0.0504	0.0338	0.4407
11	AdaBoostRegressor	0.0061	0.078.1	0.0685	-0.346
11	GradientBoostingRegressor	0.0025	0.05	0.0332	0.4491
11	XGBRegressor	0.0024	0.0494	0.0328	0.4624
11	Lasso	0.0045	0.0674	0.0505	0
11	ElasticNet	0.0045	0.0674	0.0505	0

The complex patterns of predicted performance vary across different window sizes, with each size imparting distinct characteristics to the algorithms.

- With a window size of 1, the XGBRegressor method stands out as the top performer, with an R² parameter value of 0.4156. The AdaBoostRegressor performs poorly, showing a weak stride with a value of 0.1129. The Lasso and ElasticNet algorithms show mediocre performance.
- Increasing the window size to 3, the XGBRegressor demonstrates its superiority by achieving an R² parameter value of 0.458. On the other hand, the Lasso and ElasticNet

algorithms encounter a situation where they stop making progress, resulting in a value of 0.

- The XGBRegressor demonstrates predictive ability, with an R^2 score of 0.4494 and a window size of 5. The Lasso and ElasticNet algorithms exhibit a subdued trajectory, converging to a value of 0.
- The XGBRegressor shows a crescendo in performance, achieving an R^2 parameter value of 0.4628 with a window size of 7. On the contrary, the Lasso and ElasticNet algorithms experience reduced output when the value is 0.
- The XGBRegressor stands out with an R^2 parameter value of 0.4659 in a window size of 9. Both the Lasso and ElasticNet algorithms exhibit a gradual decrease, reaching a value of 0.0001.
- With a window size of 11, the XGBRegressor's performance improves, resulting in an R^2 value of 0.4624. The AdaBoostRegressor demonstrates modesty by yielding a value of 0.346.

Table 5 provides detailed information on the interaction between algorithms and window sizes. The research concludes that a window size of 7 indicates optimal performance. Around 70% of the algorithms perform optimally throughout this period, leading to a dependable outcome.

• Second Scenario

As we delve further into our exploration, the second scenario involves dividing the data into segments of varying sizes—50,000, 75,000, and 100,000 records—each revealing unique insights through regression methods. Algorithms play various roles in these explanations by creating patterns that are well-suited to the dataset's size. Table 6 illustrates the interaction between data pieces and regression algorithms in response to this case.

From the outcomes of each segment size, we may draw the following conclusions from Table 6:

- With a segment size of 50,000, the XGBRegressor method demonstrates an R^2 parameter value of 0.4486, taking the lead in the ensemble of algorithms. The AdaBoostRegressor method communicates its story softly, with a coefficient of 0.0434. The GradientBoosting and KNeighbors algorithms stand out as top performers, leaving a unique mark in the story.
- The XGBRegressor algorithm emerges as the top performer with an R^2 value of 0.4395, while the AdaBoostRegressor algorithm performs less impressively at 1.124 in a segment size of 75,000. Every algorithm contributes to the discussion, creating its unique mark on the overall picture.
- With a segment size of 100,000, the XGBRegressor algorithm achieves an R^2 parameter value of 0.4417, outperforming the Lasso and ElasticNet algorithms, which both score 0. Three narratives arise, reflecting the interaction between algorithms and data records.

When applied to a wide range of data record counts, the XGBRegressor algorithm consistently delivers more satisfactory results.

• Third Scenario

As our research progresses to the third scenario, the k-fold method provides valuable insights into the practical utility of the model, revealing the effectiveness of predictions. The parameter "k" determines the division of data samples into k subsets. These subsets are utilized for both training and validation purposes.

K-fold cross-validation involves dividing the dataset into k subsets. One subset is selected for testing, while the remaining $k-1$ subsets are used for training. There are 10 threads of data that form the tapestry, encompassing nine regression algorithms. Table 7 displays the unexpected findings and the effectiveness of the k-fold method in achieving accurate results.

Table 6. Results of different data parts for the algorithms based on different parameters.

Size of Each Data Part	Algorithms	MSE	RMSE	MAE	R ²
50,000	KNeighborsRegressor	0.0042	0.065	0.043326	0.4353
50,000	LinearRegression	0.0043	0.0653	0.044548	0.429
50,000	LinearSVR	0.0048	0.0692	0.043532	0.3596
50,000	RandomForest Regressor	0.0044	0.0664	0.044555	0.4108
50,000	AdaBoostRegressor	0.0078	0.08.83	0.075135	−0.0434
50,000	GradientBoostingRegressor	0.0042	0.0644	0.043308	0.4445
50,000	XGBRegressor	0.0041	0.0642	0.042847	0.4486
50,000	Lasso	0.0075	00865	0.064728	0
50,000	ElasticNet	0.0075	00865	0.064728	0
75,000	KNeighborsRegressor	0.0026	0.0511	0.033681	0.4357
75,000	LinearRegression	0.0027	0.0515	0.034808	0.4166
75,000	LinearSVR	0.0028	0.0531	0.032382	0.3799
75,000	RandomForest Regressor	0.0027	0.0518	0.034425	0.411
75,000	AdaBoostRegressor	0.0097	0.0983	0.089883.	−1.124
75,000	GradientBoostingRegressor	0.0026	0.0508	0.033757	0.432
75,000	XGBRegressor	0.0025	0.0505	0.033364	0.4395
75,000	Lasso	0.0045	0.0674	0.05034	0
75,000	ElasticNet	0.0045	0.0674	0.05034	0
100,000	KNeighborsRegressor	0.0027	00518	0.034224	0.4242
100,000	LinearRegression	0.0027	0.0521	0.035349	0.4177
100,000	LinearSVR	0.003	0.055	0.034161	0.3519
100,000	RandomForest Regressor	0.0027	0.0523	0.034851	0.4134
100,000	AdaBoostRegressor	0.004	0.0632	0.052425	0.1442
100,000	GradientBoostingRegressor	0.0026	0.0514	0.034216	0.4333
100,000	XGBRegressor	0.0026	0.0511	0.033777	0.4417
100,000	Lasso	0.0047	0.0683	0.051225	0
100,000	ElasticNet	0.0047	0.0683	0.051225	0

Table 7. Outcomes of our algorithms employing our parameters in the k-fold technique.

k	Algorithm_Name	MSE	RMSE	MAE	R ²
1	KNeighborsRegressor	0	0.007	0.004554	0.4319
1	LinearRegression	0	0.007	0.00473	0.4228
1	LinearSVR	0.0001	0.0093	0.00688	0.0029
1	RandomForestRegressor	0	0.0071	0.004652	0.4205
1	AdaBoostRegressor	0.0001	0.0083	0.006639	0.209
1	GradientBoostingRegressor	0	0.007	0.0046	0.4314
1	XGBRegressor	0	0.0069	0.004516	0.4433
1	Lasso	0.0001	0.0093	0.006879	0
1	ElasticNet	0.0001	0.0093	0.006879	0

Table 7. Cont.

k	Algorithm_Name	MSE	RMSE	MAE	R²
2	KNeighborsRegressor	0	0.007	0.004545	0.4354
2	LinearRegression	0	0.007	0.004724	0.4226
2	LinearSVR	0.0001	0.0071	0.005144	0.4067
2	RandomForestRegressor	0	0.007	0.004652	0.4206
2	AdaBoostRegressor	0.0001	0.0082	0.006646	0.2141
2	GradientBoostingRegressor	0	0.0069	0.004586	0.4381
2	XGBRegressor	0	0.0069	0.004514	0.4476
2	Lasso	0.0001	0.0093	0.006886	0
2	ElasticNet	0.0001	0.0093	0.006886	0
3	KNeighborsRegressor	0	0.007	0.004553	0.4317
3	LinearRegression	0	0.0071	0.004732	0.4167
3	LinearSVR	0.0001	0.0073	0.004471	0.3704
3	RandomForestRegressor	0.0001	0.0071	0.004656	0.4128
3	AdaBoost Regressor	0.0003	0.016	0.01224	−1.9995
3	GradientBoostingRegressor	0	0.007	0.004599	0.4301
3	XGBRegressor	0	0.0069	0.00452	0.4402
3	Lasso	0.0001	0.0093	0.006873	0
3	ElasticNet	0.0001	0.0093	0.006873	0
4	KNeighborsRegressor	0	0.0071	0.004562	0.4228
4	LinearRegression	0.0001	0.0071	0.004743	0.4127
A	LinearSVR	0.0001	0.0095	0.008205	−0.0491
4	RandomForestRegressor	0.0001	0.0071	0.004662	0.4108
4	AdaBoostRegressor	0.0004	0.0199	0.016922	−3.6026
4	GradientBoostingRegressor	0.0001	0.0071	0.004616	0.4214
4	XGBRegressor	0	0.007	0.004531	0.436
4	Lasso	0.0001	0.0093	0.006885	0
4	ElasticNet	0.0001	0.0093	0.006885	0
5	KNeighborsRegressor	0	0.007	0.004566	0.4267
5	LinearRegression	0	0.007	0.004745	0.4225
5	LinearSVR	0.0001	0.0075	0.004588	0.3488
5	RandomForestRegressor	0.0001	0.0071	0.004663	0.407
5	AdaBoostRegressor	0.0001	0.0088	0.006909	0.1045
5	GradientBoostingRegressor	0.0001	0.0073	0.004623	0.3747
5	XGBRegressor	0.0001	0.0073	0.004541	0.3761
5	Lasso	0.0001	0.0093	0.006882	0
5	ElasticNet	0.0001	0.0093	0.006882	0
6	KNeighborsRegressor	0	0.0069	0.004538	0.4372
6	LinearRegression	0.0001	0.0071	0.004717	0.3977
6	LinearSVR	0.0001	0.0082	0.006636	0.2024
6	RandomForestRegressor	0	0.007	0.004638	0.422
6	AdaBoostRegressor	0.0001	0.0084	0.006924	0.1487

Table 7. Cont.

k	Algorithm_Name	MSE	RMSE	MAE	R²
6	GradientBoostingRegressor	0	0.0069	0.004595	0.4336
6	XGBRegressor	0	0.0068	0.00451	0.4477
6	Lasso	0.0001	0.0092	0.006883	0
6	ElasticNet	0.0001	0.0092	0.006883	0
7	KNeighborsRegressor	0.0001	0.0076	0.004562	0.3993
7	LinearRegression	0.0001	0.0075	0.004743	0.4201
7	LinearSVR	0.0001	0.0109	0.008452	−0.2307
7	RandomForestRegressor	0.0001	0.0077	0.004662	0.3857
7	AdaBoostRegressor	0.0018	0.0428	0.036901	−18.0133
7	GradientBoostingRegressor	0.0001	0.0076	0.004603	0.3981
7	XGBRegressor	0.0001	0.0076	0.004525	0.4055
7	Lasso	0.0001	0.0098	0.006902	0
7	ElasticNet	0.0001	0.0098	0.006902	0
8	KNeighborsRegressor	0	0.007	0.004564	0.4295
8	LinearRegression	0.0001	0.0071	0.00474	0.4114
8	LinearSVR	0.0002	0.0134	0.012251	−1.1097
8	RandomForestRegressor	0.0001	0.0071	0.004686	0.4115
8	AdaBoostRegressor	0.0001	0.0084	0.006911	0.1732
8	GradientBoostingRegressor	0	0.007	0.004618	0.4253
8	XGBRegressor	0	0.0069	0.004541	0.4374
8	Lasso	0.0001	0.0092	0.006881	0
8	ElasticNet	0.0001	0.0092	0.006881	0
9	KNeighborsRegressor	0	0.0069	0.004527	0.44
9	LinearRegression	0	0.007	0.004718	0.4198
9	LinearSVR	0	0.007	0.004775	0.4185
9	RandomForestRegressor	0	0.007	0.004633	0.4208
9	AdaBoostRegressor	0.0001	0.0082	0.006648	0.2064
9	GradientBoostingRegressor	0	0.0069	0.004583	0.4296
9	XGBRegressor	0	0.0069	0.004505	0.433
9	Lasso	0.0001	0.0092	0.006887	0
9	ElasticNet	0.0001	0.0092	0.006887	0
10	KNeighborsRegressor	0	0.0071	0.00457	0.4275
10	LinearRegression	0.0001	0.0071	0.004749	0.4209
10	LinearSVR	0.0001	0.0074	0.004489	0.3769
10	RandomForestRegressor	0.0001	0.0072	0.004678	0.4049
10	AdaBoostRegressor	0.0001	0.0107	0.009156	−0.3251
10	GradientBoostingRegressor	0	0.007	0.004618	0.4292
10	XGBRegressor	0	0.007	0.004542	0.4389
10	Lasso	0.0001	0.0093	0.00689	0
10	ElasticNet	0.0001	0.0093	0.00689	0

Algorithms showcased their proficiency through multiple trials by adjusting to the rhythm of “k” values in the evolving script of k-fold validation. The main findings from Table 7 can be summarized as follows:

The XGBRegressor demonstrates its power for $k = 1$, achieving an R^2 parameter value of 0.4433. The Lasso and ElasticNet have coefficients of 0, while the LinearSVR model performs poorly. The KNeighbors, Linear, RandomForest, and GradientBoost algorithms synergize effectively. During the $k = 2$ iteration, the XGBRegressor model achieves an R-squared value of 0.4476. The Lasso, ElasticNet, and AdaBoost algorithms are experiencing setbacks. KNeighbors, Linear, LinearSVR, RandomForest, and GradientBoost models harmonize perfectly.

During the beginning of $k = 3$, the XGBRegressor shows an R^2 value of 0.4402. AdaBoost fails at 1.9995, while Lasso and ElasticNet exhibit weakness. When $k = 4$, XGBRegressor achieves an R^2 of 0.436, while AdaBoost drops to 3.6026, and LinearSVR only reaches 0.0491. KNeighbors performs well with an R^2 of 0.4267, while Lasso and ElasticNet do not perform as strongly. In $k = 6$'s analysis, XGBRegressor outperformed Lasso and ElasticNet with an R^2 of 0.4477. AdaBoost's explanation lacks impact.

When $k = 7$, this correlates with Linear's R^2 at 0.4201, but AdaBoost shows a discrepancy at 18.0133. LinearSVR's optimization fails. In chapter 8, the XGBRegressor achieved an R^2 score of 0.4374, while the LinearSVR scored 1.1097. An orchestra of equality exists alongside the others. KNeighbors' R^2 in stanza $k = 9$ is 0.44, with Lasso and ElasticNet following behind. The XGBRegressor model with $k = 10$ produced a final R^2 score of 0.4389. AdaBoost registers a value of 0.3251, while the other models show a consistent response.

- **Fourth Scenario**

In the fourth case, the model's error rate is equivalent to the average error rate observed in all iterations. In this system, “k” is equivalent to “n,” which represents the number of samples in the dataset. This plot resembles an extended k-fold cross-validation and involves the utilization of the Leave One Out (LOO) method. Table 8 displays the results of this case and the knowledge acquired through this approach.

Table 8. Comparison of parameters using LOO technique.

Metrics	Results
MSE	0.009389521
RMSE	0.096899541
MAE	0.06528897
R^2	0.404312521

5. Discussion and Managerial Insights

The above-mentioned findings offer a comprehensive overview of outcomes derived from different scenarios, parameters, and evaluations, aiming to provide deep insights into the practicality and efficiency of the proposed algorithms through an Iranian case study.

The analysis of different scenarios has uncovered interesting patterns. The XGBRegressor algorithm consistently outperformed others in predictive accuracy across various window sizes, demonstrating its ability to effectively manage variations in the data's level of detail. The XGBRegressor also performed well in data-partitioning scenarios, handling different data quantities adeptly. In particular, the dataset containing 75,000 records required further investigation into data distribution and algorithmic patterns. The k-fold cross-validation method revealed nuanced variations in algorithm efficacy, with XGBRegressor remaining the top performer, although other algorithms showed strengths in different iterations. This underscores the importance of using diverse validation strategies in model assessment. Both the Leave One Out (LOO) technique and the k-fold method highlighted the importance of validation techniques in achieving conclusive results. The parameters MSE, RMSE, MAE, and R^2 served as benchmarks for model evaluation in all

situations. These metrics provided a comprehensive view of each algorithm’s predictive capability. The XGBRegressor consistently demonstrated strong performance, aligning with its reputation for reliable and accurate prediction. The managerial implications suggest that algorithms like RandomForest, KNeighbors, and XGBRegressor hold significant potential for practical use.

5.1. Managerial Insights

The outcomes can provide crucial managerial insights that can significantly enhance transportation and logistics management, as depicted in Figure 10.

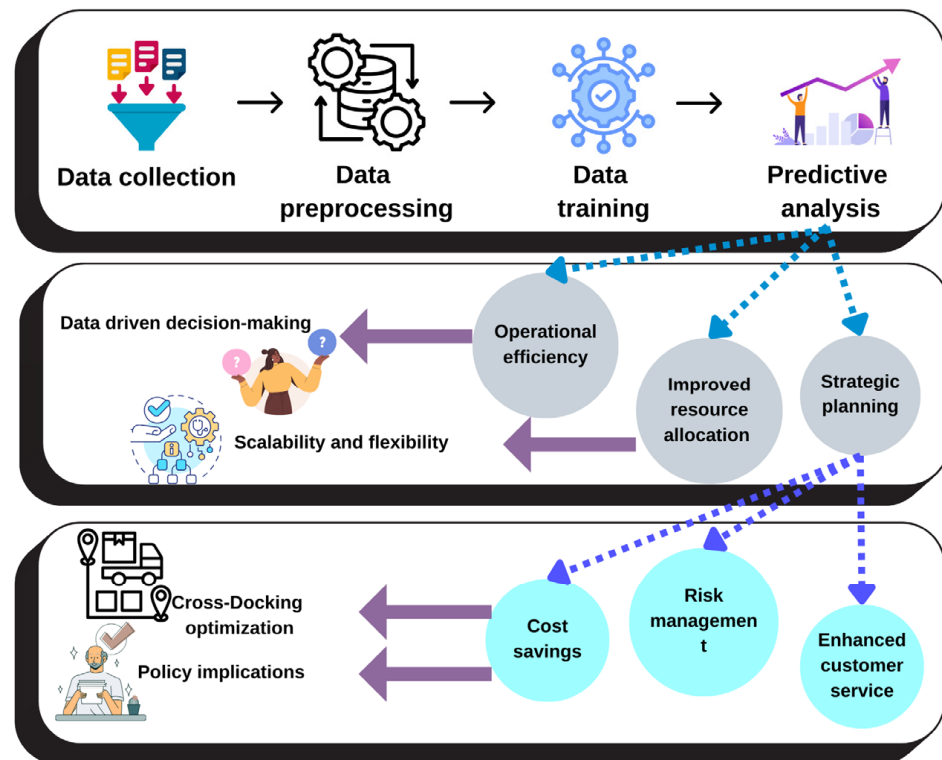


Figure 10. The managerial insight plan of the present study.

By leveraging GPS data and advanced regression algorithms such as XGBoost, RandomForest, and GradientBoosting, the study achieves high accuracy in predicting truck transit times. This precision is vital for logistics managers to effectively plan and optimize delivery schedules, ultimately improving operational efficiency. For example, Zhao et al. (2019) used GPS data from Beijing’s Sixth Ring Road to predict truck travel speeds under various conditions with an optimized GRU algorithm, demonstrating the practical application of similar methodologies [46].

Accurate transit time predictions enable logistics companies to streamline their operations by minimizing idle time, optimizing routes, and improving fleet utilization, leading to cost reductions and increased operational efficiency. Better predictions also allow for more precise resource allocation, such as scheduling loading and unloading activities and managing driver shifts. This reduces bottlenecks at cross-docking terminals and other logistics hubs, ensuring a smoother flow of goods. Wang et al. (2020) similarly applied machine learning techniques to improve driving style identification in open-pit mining, demonstrating the broader applicability of these methods [47].

The integration of GPS data and machine learning provides a robust foundation for data-driven decision-making. Managers can rely on empirical data and sophisticated algorithms rather than heuristics or past experiences. This data-driven approach enhances the scalability and flexibility of predictive models, allowing for adaptation to different

regions, traffic conditions, and logistical scenarios. Predictive analytics also aid in long-term strategic planning. Understanding traffic patterns and potential delays can inform infrastructure development, investment in new technologies, and partnerships with other logistics providers. Predictive models help identify potential delays and disruptions in advance, enabling managers to develop contingency plans to mitigate risks, ensuring more reliable delivery schedules and improved customer satisfaction. Rivera-Campoverde et al. (2024) demonstrated similar applications in the management of vehicle emissions, showing the versatility of these approaches [48].

The study highlights the importance of synchronizing inbound and outbound logistics at cross-docking terminals. Efficiently managing these terminals reduces wait times and ensures a smooth flow of goods, especially crucial for time-sensitive and perishable items. For policymakers, the study provides valuable insights into traffic management and infrastructure development, informing policies aimed at reducing congestion and improving road safety.

5.2. Alignment with Sustainable Development Goals (SDGs)

The present study aligns with several Sustainable Development Goals (SDGs) through its implications in logistics, urban planning, and transportation efficiency. The following sections provide a detailed assessment of the study's impact on specific SDGs [49–51]:

SDG 3: Improved health and safety on roads due to reduced congestion and accidents.

SDG 7: Lower fuel consumption through optimized routes.

SDG 8: Increased economic productivity and better working conditions for drivers.

SDG 9: Innovations in transportation infrastructure and logistics.

SDG 11: Enhanced urban sustainability and livability.

SDG 12: More efficient use of resources in production and distribution.

SDG 13: Reduced greenhouse gas emissions.

The conceptual model in Figure 11 illustrates the flow from data collection and processing to the positive impacts on various SDGs, showcasing the broader societal benefits of the research on truck transit time prediction.

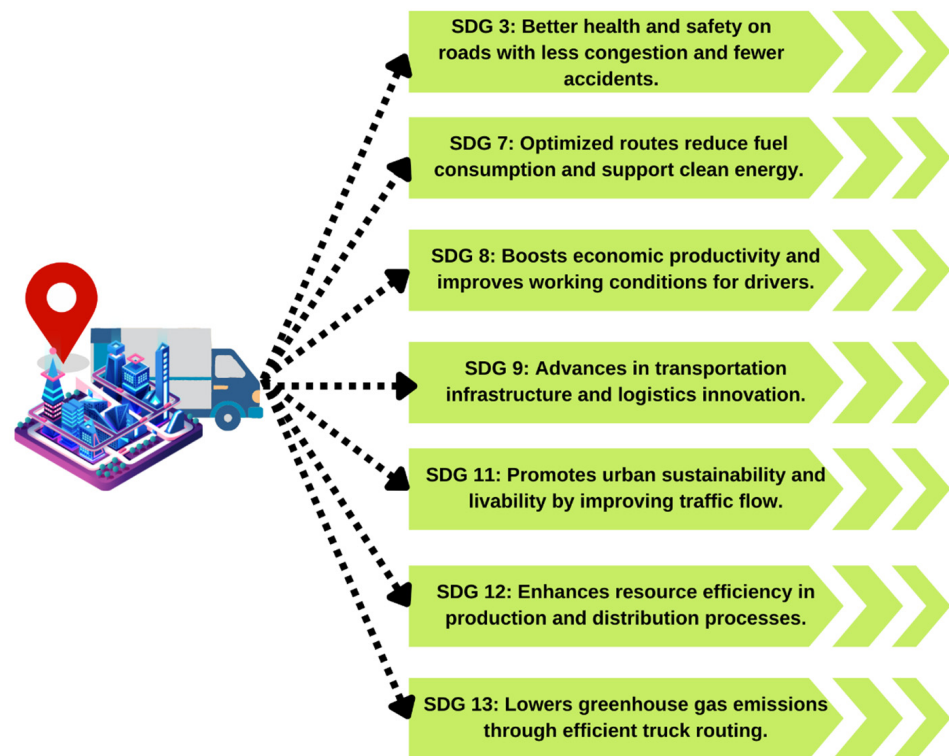


Figure 11. The assessment of SDGs aspects in the present study.

6. Conclusions, Limitations, and Future Works

The effective optimization of truck scheduling in cross-docking terminals and traffic management in large urban areas requires an insightful strategy. Restrictions on heavy-duty trucks, which limit them to specific routes, often lead to traffic congestion. Thus, predicting travel speeds on specific roads is crucial to providing customized information services to drivers. This study utilized truck-generated tracking data and regression algorithms, recognized for their predictive capabilities in estimating travel durations and vehicle locations. By employing a range of regression algorithms, including AdaBoost, GradientBoost, XGBoost, ElasticNet, RandomForest, KNeighbors, Linear, LinearSVR, and Lasso, our investigation provides insights across various scenarios. Among these, the XGBRegressor algorithm consistently stands out as a superior predictor, surpassing other algorithms in different time steps and situations. This finding opens exciting research opportunities for the future, including investigating different regression algorithms and performing comparative analyses to understand their effectiveness relative to our results.

Expanding the scope of this predictive model to include a broader range of vehicle types, such as buses, vans, and cars, presents a promising opportunity. These diverse vehicle types exhibit inherent variations in movement patterns, characteristics, and operational dynamics, making them rich subjects for investigation. By customizing and refining predictive methodologies to match their unique characteristics, we can develop a comprehensive toolkit for predicting the travel times and trajectories of various vehicles. This effort is crucial for enhancing transportation management strategies that address both freight logistics and the efficient movement of passengers, promoting a holistic approach to optimizing mobility.

Our research established a solid foundation for improved predictive models in estimating truck travel time. Future research can focus on ongoing exploration, method improvement, and expanding applications to various vehicle types. This advancement will contribute significantly to transportation management, achieving effective and sustainable mobility solutions [52,53].

Author Contributions: Conceptualization and design: A.G. and S.D.; Data collection: M.G. and R.M.; Analysis and interpretation of the results: M.G., A.G. and A.M.F.-F.; Draft manuscript preparation: A.G., S.D., M.G. and R.M.; Writing—review and editing: R.M.; Supervision: A.M.F.-F. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank Technical University of Liberec for their support through the Student Grant Competition SGS-2023–3401. The research also was supported by Research Infrastructure NanoEnviCz, via Czech Republic’s Ministry of Education, Youth, and Sports under Project No. LM2023066.

Data Availability Statement: The datasets examined in this research are not accessible to the public because of privacy considerations, but the data can be made accessible upon a reasonable request to the corresponding author.

Conflicts of Interest: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Moniruzzaman, M.; Maoh, H.; Anderson, W. Short-term prediction of border crossing time and traffic volume for commercial trucks: A case study for the Ambassador Bridge. *Transp. Res. Part C Emerg. Technol.* **2016**, *63*, 182–194. [[CrossRef](#)]
2. Golob, T.F.; Regan, A.C. Impacts of highway congestion on freight operations: Perceptions of trucking industry managers. *Transp. Res. Part A Policy Pract.* **2001**, *35*, 577–599. [[CrossRef](#)]
3. Theophilus, O.; Dulebenets, M.A.; Pasha, J.; Lau, Y.Y.; Fathollahi-Fard, A.M.; Mazaheri, A. Truck scheduling optimization at a cold-chain cross-docking terminal with product perishability considerations. *Comput. Ind. Eng.* **2021**, *156*, 107240. [[CrossRef](#)]
4. Fathollahi-Fard, A.M.; Ranjbar-Bourani, M.; Cheikhrouhou, N.; Hajjaghahi-Keshteli, M. Novel modifications of social engineering optimizer to solve a truck scheduling problem in a cross-docking system. *Comput. Ind. Eng.* **2019**, *137*, 106103. [[CrossRef](#)]
5. Shi, L.; Liu, M.; Liu, Y.; Zhao, Q.; Cheng, K.; Zhang, H.; Fathollahi-Fard, A.M. Evaluation of Urban Traffic Accidents Based on Pedestrian Landing Injury Risks. *Appl. Sci.* **2022**, *12*, 6040. [[CrossRef](#)]

6. Nadi, A.; Sharma, S.; Snelder, M.; Bakri, T.; van Lint, H.; Tavasszy, L. Short-term prediction of outbound truck traffic from the exchange of information in logistics hubs: A case study for the port of Rotterdam. *Transp. Res. Part C Emerg. Technol.* **2021**, *127*, 103111. [[CrossRef](#)]
7. Borowska-Stefańska, M.; Kowalski, M.; Kurzyk, P.; Sahebgharani, A.; Sapińska, P.; Wiśniewski, S.; Goniewicz, K.; Dulebenets, M.A. Assessing the impacts of sunday trading restrictions on urban public transport: An example of a big city in central Poland. *J. Public Transp.* **2023**, *25*, 100049. [[CrossRef](#)]
8. Sun, Z.; Ban, X.J. Vehicle classification using GPS data. *Transp. Res. Part C Emerg. Technol.* **2013**, *37*, 102–117. [[CrossRef](#)]
9. Gingerich, K.; Maoh, H.; Anderson, W. Classifying the purpose of stopped truck events: An application of entropy to GPS data. *Transp. Res. Part C Emerg. Technol.* **2016**, *64*, 17–27. [[CrossRef](#)]
10. Li, F.; Feng, J.; Yan, H.; Jin, G.; Yang, F.; Sun, F.; Jin, D.; Li, Y. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 1–21. [[CrossRef](#)]
11. Zhou, M.; Kong, N.; Zhao, L.; Huang, F.; Wang, S.; Campy, K.S. Understanding urban delivery drivers' intention to adopt electric trucks in China. *Transp. Res. Part D Transp. Environ.* **2019**, *74*, 65–81. [[CrossRef](#)]
12. Bai, R.; Xue, N.; Chen, J.; Roberts, G.W. A set-covering model for a bidirectional multi-shift full truckload vehicle routing problem. *Transp. Res. Part B Methodol.* **2015**, *79*, 134–148. [[CrossRef](#)]
13. Yang, Y.; Jia, B.; Yan, X.Y.; Jiang, R.; Ji, H.; Gao, Z. Identifying intracity freight trip ends from heavy truck GPS trajectories. *Transp. Res. Part C Emerg. Technol.* **2022**, *136*, 103564. [[CrossRef](#)]
14. Gheibi, M.; Karrabi, M.; Latifi, P.; Fathollahi-Fard, A.M. Evaluation of traffic noise pollution using geographic information system and descriptive statistical method: A case study in Mashhad, Iran. In *Environmental Science and Pollution Research*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–14.
15. Bombelli, A.; Fazi, S. The ground handler dock capacitated pickup and delivery problem with time windows: A collaborative framework for air cargo operations. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *159*, 102603. [[CrossRef](#)]
16. Ni, L.; Wang, X.C.; Zhang, D. Impacts of information technology and urbanization on less-than-truckload freight flows in China: An analysis considering spatial effects. *Transp. Res. Part A Policy Pract.* **2016**, *92*, 12–25. [[CrossRef](#)]
17. Popken, D.A. An analytical framework for routing multiattribute multicommodity freight. *Transp. Res. Part B Methodol.* **1996**, *30*, 133–145. [[CrossRef](#)]
18. Li, N.; Wu, Y.; Wang, Q.; Ye, H.; Wang, L.; Jia, M.; Zhao, S. Underground mine truck travel time prediction based on stacking integrated learning. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105873. [[CrossRef](#)]
19. Sharman, B.W.; Roorda, M.J. Multilevel modelling of commercial vehicle inter-arrival duration using GPS data. *Transp. Res. Part E Logist. Transp. Rev.* **2013**, *56*, 94–107. [[CrossRef](#)]
20. Zhao, J.; Gao, Y.; Yang, Z.; Li, J.; Feng, Y.; Qin, Z.; Bai, Z. Truck traffic speed prediction under non-recurrent congestion: Based on optimized deep learning algorithms and GPS data. *IEEE Access* **2019**, *7*, 9116–9127. [[CrossRef](#)]
21. Zhao, W.; Goodchild, A.V. Truck travel time reliability and prediction in a port drayage network. *Marit. Econ. Logist.* **2011**, *13*, 387–418. [[CrossRef](#)]
22. Morgul, E.F.; Ozbay, K.; Iyer, S.; Holguin-Veras, J. Commercial vehicle travel time estimation in urban networks using GPS data from multiple sources. In Proceedings of the Transportation Research Board 92nd Annual Meeting (No. 13–4439), Washington, DC, USA, 13–17 January 2013.
23. Jiang, F. Bus Transit Time Prediction Using GPS Data with Artificial Neural Networks. 2017. Available online: <https://www.semanticscholar.org/paper/Bus-Transit-Time-Prediction-using-GPS-Data-with-Jiang/fd4d5ffba0471cffe9b0045b3b4407b26ef160> (accessed on 20 May 2023).
24. Wang, S.; Zhao, J.; Shao, C.; Dong, C.; Yin, C. Truck traffic flow prediction based on LSTM and GRU methods with sampled GPS data. *IEEE Access* **2020**, *8*, 208158–208169. [[CrossRef](#)]
25. Demissie, M.G.; Kattan, L. Estimation of truck origin-destination flows using GPS data. *Transp. Res. Part E Logist. Transp. Rev.* **2022**, *159*, 102621. [[CrossRef](#)]
26. Pani, C.; Fadda, P.; Fancello, G.; Frigau, L.; Mola, F. A data mining approach to forecast late arrivals in a transshipment container terminal. *Transport* **2014**, *29*, 175–184. [[CrossRef](#)]
27. Bhattacharya, A.; Kumar, S.A.; Tiwari, M.K.; Talluri, S. An intermodal freight transport system for optimal supply chain logistics. *Transp. Res. Part C Emerg. Technol.* **2014**, *38*, 73–84. [[CrossRef](#)]
28. Li, X.; Bai, R. Freight vehicle travel time prediction using gradient boosting regression tree. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; pp. 1010–1015.
29. Van der Spoel, S.; Amrit, C.; van Hillegersberg, J. Predictive analytics for truck arrival time estimation: A field study at a European distribution centre. *Int. J. Prod. Res.* **2017**, *55*, 5062–5078. [[CrossRef](#)]
30. Salleh NH, M.; Riahi, R.; Yang, Z.; Wang, J. Predicting a containership's arrival punctuality in liner operations by using a fuzzy rule-based bayesian network (frbnn). *Asian J. Shipp. Logist.* **2017**, *33*, 95–104. [[CrossRef](#)]
31. Alcoba, R.D.; Ohlund, K.W. Predicting on-Time Delivery in the Trucking Industry. Ph.D. Dissertation, Supply Chain Management Program, Massachusetts Institute of Technology, Cambridge, MA, USA, 2017. Available online: <https://dspace.mit.edu/handle/1721.1/112870> (accessed on 20 May 2023).

32. Barbour, W.; Samal, C.; Kuppa, S.; Dubey, A.; Work, D.B. On the data-driven prediction of arrival times for freight trains on us railroads. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2289–2296.
33. Wu, R.; Luo, G.; Shao, J.; Tian, L.; Peng, C. Location prediction on trajectory data: A review. *Big Data Min. Anal.* **2018**, *1*, 108–127. [[CrossRef](#)]
34. James, J.Q.; Yu, W.; Gu, J. Online vehicle routing with neural combinatorial optimization and deep reinforcement learning. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3806–3817.
35. Yu, J.; Tang, G.; Song, X.; Yu, X.; Qi, Y.; Li, D.; Zhang, Y. Ship arrival prediction and its value on daily container terminal operation. *Ocean Eng.* **2018**, *157*, 73–86. [[CrossRef](#)]
36. Balster, A.; Hansen, O.; Friedrich, H.; Ludwig, A. An ETA prediction model for intermodal transport networks based on machine learning. *Bus. Inf. Syst. Eng.* **2020**, *62*, 403–416. [[CrossRef](#)]
37. Servos, N.; Liu, X.; Teucke, M.; Freitag, M. Travel time prediction in a multimodal freight transport relation using machine learning algorithms. *Logistics* **2019**, *4*, 1. [[CrossRef](#)]
38. Verma, A.K.; Saxena, R.; Jadeja, M.; Bhateja, V.; Lin, J.C.W. Bet-GAT: An Efficient Centrality-Based Graph Attention Model for Semi-Supervised Node Classification. *Appl. Sci.* **2023**, *13*, 847. [[CrossRef](#)]
39. Liu, Y.; Zou, B.; Ni, A.; Gao, L.; Zhang, C. Calibrating microscopic traffic simulators using machine learning and particle swarm optimization. *Transp. Lett.* **2021**, *13*, 295–307. [[CrossRef](#)]
40. Antamis, T.; Medentzidis, C.R.; Skoumperdis, M.; Vafeiadis, T.; Nizamis, A.; Ioannidis, D.; Tzovaras, D. AI-supported forecasting of intermodal freight transportation delivery time. In Proceedings of the 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), Riga, Latvia, 14–15 October 2021; pp. 1–6.
41. Valatsos, P.; Vafeiadis, T.; Nizamis, A.; Ioannidis, D.; Tzovaras, D. Freight transportation route time prediction with ensemble learning techniques. In Proceedings of the 25th Pan-Hellenic Conference on Informatics, Volos, Greece, 26–28 November 2021; pp. 52–57.
42. Konečný, V.; Břidzиковá, M.; Marienka, P. Research of bus transport demand and its factors using multicriteria regression analysis. *Transp. Res. Procedia* **2021**, *55*, 180–187. [[CrossRef](#)]
43. Costa, M.; Félix, R.; Marques, M.; Moura, F. Impact of COVID-19 lockdown on the behavior change of cyclists in Lisbon, using multinomial logit regression analysis. *Transp. Res. Interdiscip. Perspect.* **2022**, *14*, 100609. [[CrossRef](#)]
44. Comi, A.; Zhuk, M.; Kovalyshyn, V.; Hilevyeh, V. Investigating bus travel time and predictive models: A time series-based approach. *Transp. Res. Procedia* **2020**, *45*, 692–699. [[CrossRef](#)]
45. Ma, T.; Antoniou, C.; Toledo, T. Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. *Transp. Res. Part C Emerg. Technol.* **2020**, *111*, 352–372. [[CrossRef](#)]
46. Li, M.; Qi, J.; Tian, X.; Guo, H.; Liu, L.; Fathollahi-Fard, A.M.; Tian, G. Smartphone-based straw incorporation: An improved convolutional neural network. *Comput. Electron. Agric.* **2024**, *221*, 109010. [[CrossRef](#)]
47. Wang, Q.; Zhang, R.; Wang, Y.; Lv, S. Machine learning-based driving style identification of truck drivers in open-pit mines. *Electronics* **2019**, *9*, 19. [[CrossRef](#)]
48. Rivera-Campoverde, N.D.; Arenas-Ramírez, B.; Muñoz Sanz, J.L.; Jiménez, E. GPS Data and Machine Learning Tools, a Practical and Cost-Effective Combination for Estimating Light Vehicle Emissions. *Sensors* **2024**, *24*, 2304. [[CrossRef](#)]
49. Behdadfar, E.; Samaei, S.R. Towards a Smart Tehran: Leveraging Machine Learning for Sustainable Development, Balanced Growth, and Resilience. *J. New Res. Smart City* **2024**, *2*, 53–67.
50. Alqahtani, H.; Kumar, G. Machine learning for enhancing transportation security: A comprehensive analysis of electric and flying vehicle systems. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107667. [[CrossRef](#)]
51. Kunieda, Y.; Suzuki, H. A Detection Method of Garbage Collection Status from Sound of Garbage Trucks. In Proceedings of the 2024 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 6–8 January 2024; pp. 1–6.
52. Zhan, C.; Zhang, X.; Yuan, J.; Chen, X.; Zhang, X.; Fathollahi-Fard, A.M.; Tian, G. A hybrid approach for low-carbon transportation system analysis: Integrating CRITIC-DEMATEL and deep learning features. *Int. J. Environ. Sci. Technol.* **2024**, *21*, 791–804. [[CrossRef](#)] [[PubMed](#)]
53. Fathollahi-Fard, A.M.; Woodward, L.; Akhrif, O. A distributed permutation flow-shop considering sustainability criteria and real-time scheduling. *J. Ind. Inf. Integr.* **2024**, *39*, 100598. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.