

## Article

# Analysis of the Travelling Time According to Weather Conditions Using Machine Learning Algorithms

Gülçin Canbulut 

Industrial Engineering Department, Nuh Naci Yazgan University, Erkilet 38030, Kayseri, Turkey;  
gcanbulut@nny.edu.tr

## Abstract

A large share of the global population now lives in urban areas, which creates growing challenges for city life. Local authorities are seeking ways to enhance urban livability, with transportation emerging as a major focus. Developing smart public transit systems is therefore a key priority. Accurately estimating travel times is essential for managing transport operations and supporting strategic decisions. Previous studies have used statistical, mathematical, or machine learning models to predict travel time, but most examined these approaches separately. This study introduces a hybrid framework that combines statistical regression models and machine learning algorithms to predict public bus travel times. The analysis is based on 1410 bus trips recorded between November 2021 and July 2022 in Kayseri, Turkey, including detailed meteorological and operational data. A distinctive aspect of this research is the inclusion of weather variables—temperature, humidity, precipitation, air pressure, and wind speed—which are often neglected in the literature. Additionally, sensitivity analyses are conducted by varying  $k$  values in the K-nearest neighbors (KNN) algorithm and threshold values for outlier detection to test model robustness. Among the tested models, CatBoost achieved the best performance with a mean squared error (MSE) of approximately 18.4, outperforming random forest (MSE = 25.3) and XGBoost (MSE = 23.9). The empirical results show that the CatBoost algorithm consistently achieves the lowest mean squared error across different preprocessing and parameter settings. Overall, this study presents a comprehensive and environmentally aware approach to travel time prediction, contributing to the advancement of intelligent and adaptive urban transportation systems.

**Keywords:** travel time prediction; machine learning; regression; KNN; CatBoost



Academic Editor: Suchao Xie

Received: 7 October 2025

Revised: 3 November 2025

Accepted: 10 November 2025

Published: 19 December 2025

**Copyright:** © 2025 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Urbanization has intensified mobility challenges worldwide, particularly in large metropolitan areas where congestion, passenger density, and environmental variability jointly influence daily commuting patterns. The growing number of people living in cities increases traffic density and travel time, which directly reduces the quality of urban life. Local authorities and public transport operators therefore seek effective strategies to enhance transport efficiency, improve service quality, and support sustainable mobility. Among these strategies, accurate travel time prediction plays a crucial role in managing operations, optimizing schedules, and improving passenger satisfaction.

Public bus travel time estimation has become a key component of intelligent transportation systems (ITSs), providing data-driven support for decision-making and dynamic planning. A wide variety of methods—ranging from traditional mathematical and statistical

models to advanced machine learning algorithms—have been developed for this purpose. Statistical regression models are easy to interpret but often fail to capture the non-linear and dynamic nature of urban transportation data. Conversely, machine learning models such as random forest, gradient boosting, or deep neural networks offer higher predictive accuracy but tend to require large datasets and may suffer from limited interpretability.

Despite the growing research in this area, most existing studies focus primarily on operational and traffic-related variables, such as route geometry, passenger count, or departure intervals, while environmental factors such as weather conditions remain largely underexplored. Yet, weather variables—temperature, humidity, precipitation, air pressure, and wind speed—significantly affect vehicle performance, road friction, and passenger behavior. The neglect of these factors limits the robustness and generalizability of many predictive models.

Recent developments in travel time prediction have increasingly relied on ensemble and deep learning models designed to enhance temporal accuracy and dynamic adaptability. These approaches primarily focus on short-term traffic fluctuations and sequential patterns, while environmental variables such as weather conditions are often overlooked. Earlier regression-based approaches, though more interpretable and computationally efficient, tend to lose accuracy under conditions of high variability or nonlinear relationships. Consequently, there remains a methodological gap between models that prioritize interpretability and those that achieve higher predictive flexibility, highlighting the need for an integrated framework that balances both dimensions.

To address this gap, the present study proposes a hybrid modeling framework that integrates statistical regression and machine learning algorithms for predicting public bus travel times. The study explicitly incorporates meteorological variables—temperature, humidity, air pressure, precipitation, and wind speed—to assess their impact on travel time estimation. Furthermore, model robustness is tested through sensitivity analyses by varying  $k$  values in the K-nearest neighbors (KNN) algorithm and adjusting threshold values for outlier detection.

This integrated framework contributes to both methodological development and practical decision-making in urban transport planning. By combining interpretability with high predictive accuracy, the study offers a more comprehensive and environmentally aware approach to travel time prediction.

From a theoretical perspective, meteorological variability can influence travel time through multiple mechanisms. Temperature and humidity affect engine efficiency and passenger comfort, influencing vehicle acceleration and boarding durations. Precipitation and wind speed alter road surface friction, reduce visibility, and lower driving speeds, while changes in air pressure are often correlated with weather fronts that modify traffic flow dynamics. These factors interact with operational variables—such as route geometry and departure intervals—to produce nonlinear fluctuations in travel time. Therefore, integrating meteorological parameters into predictive frameworks provides a physically and behaviorally grounded representation of transportation system performance under varying environmental conditions.

Accordingly, this study aims to answer the following research question: Does the inclusion of meteorological variables significantly improve the accuracy and robustness of public bus travel time prediction models?

Based on this, the research hypothesis is formulated as follows:

**H<sub>1</sub>:** Integrating weather-related parameters such as temperature, humidity, precipitation, air pressure, and wind speed enhances model performance compared to models relying solely on operational and temporal data.

This explicit formulation clarifies the analytical focus of the study and establishes a clear link between the research objective, methodological framework, and empirical evaluation.

The remainder of this paper is organized as follows: Section 2 reviews the related literature, Section 3 explains the methodology and modeling framework, Section 4 presents the application and results, and Section 5 discusses findings and future research directions.

## 2. Literature Review

Travel time prediction has been a widely explored topic in the field of intelligent transportation systems (ITSs). Numerous studies have investigated different modeling approaches—ranging from traditional statistical methods to advanced machine learning (ML) algorithms—to improve the accuracy and robustness of travel time estimation. Table 1 summarizes representative studies in this field.

**Table 1.** Travel time prediction studies.

| Authors                | Methods                                                                                                                                                                                                                                                                                        | Main Conclusion/Weakness                                                                                                                      |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Serin vd. [1]          | AdaBoost Regression, Gradient Boosted Regression, Random Forest Regression, Extra-Tree Regression, KNN Regression, Support Vector Machine                                                                                                                                                      | Ensemble learning achieved higher accuracy than single algorithms, but no environmental or meteorological variables were included.            |
| Gal vd. [2]            | Queueing Theory, Machine Learning                                                                                                                                                                                                                                                              | Proposed a hybrid queueing-ML model improving predictive reliability; however, weather effects and external disturbances were not considered. |
| Peterson vd. [3]       | Long Short-Term Memory (LSTM) Neural Network                                                                                                                                                                                                                                                   | Captured temporal and spatial dependencies effectively; interpretability was limited, and environmental variables were excluded.              |
| Bai vd. [4]            | Support Vector Machines, Kalman Filtering                                                                                                                                                                                                                                                      | Combined SVM and Kalman filter for dynamic prediction; effective for real-time data but sensitive to noise and parameter tuning.              |
| Treethidtaphat vd. [5] | Deep Neural Network, Ordinary Least Square (OLS) Regression                                                                                                                                                                                                                                    | DNN outperformed OLS in travel time prediction, yet the model required large datasets and lacked contextual (e.g., weather) factors.          |
| Chen vd. [6]           | Linear Regression, The Least Absolute Shrinkage and Selection Operator, K-Nearest Neighbors Regression, Support Vector Regression, Gradient Boosting Regression, The Long Short Term Memory Network, Bi-Directional Long Short-Term Memory, Seasonal Auto-Regressive Integrated Moving Average | Comprehensive comparison of multiple ML and statistical models; improved accuracy but ignored environmental influence and interpretability.   |
| Ashwini et al. [7]     | Linear Regression, Ridge Regression, Least Absolute Shrinkage and Selection Operator Regression, Support Vector Regression, K-Nearest Neighbors, Regression Trees, Random Forest Regression, Gradient Boosting Regression                                                                      | Showed temporal and route-direction variables improve performance; however, no weather or external conditions were included.                  |

Table 1. Cont.

| Authors              | Methods                                                                                   | Main Conclusion/Weakness                                                                                                                          |
|----------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Servos et al. [8]    | Extremely Randomized Trees, Adaptive Boosting (AdaBoost), Support Vector Regression (SVR) | Ensemble algorithms outperformed mean-based approaches; dataset limited to freight transport and lacked meteorological diversity.                 |
| Ceylan and Özcan [9] |                                                                                           | Optimized bus frequencies using a meta-heuristic algorithm; not a predictive model and did not assess real-time variability.                      |
| Reddy et al. [10]    | Support Vector Regression (SVR)                                                           | SVR improved prediction under variable traffic conditions; applicability restricted by small-scale data and absence of weather inputs.            |
| Moosavi et al. [11]  | Chi-Square Automatic Interaction Detection, Random Forest, Gradient Boost Tree            | Tree-based methods performed well across routes with differing frequencies; however, interpretability remained limited.                           |
| Wu et al. [12]       | ConvLSTM, LSTM                                                                            | Integrated convolutional and recurrent structures enhanced temporal precision; required high computational cost and excluded weather factors.     |
| Lee et al. [13]      | ConvLSTM                                                                                  | Utilized spatio-temporal features for improved prediction; model complexity hindered practical deployment, and no environmental inputs were used. |
| He et al. [14]       | Interval-Based Historical Average Model, The Long Short Term Memory Network               | Separated riding and waiting times for greater granularity; still limited by small-scale experiments and no weather analysis.                     |

Machine learning models have become dominant in recent years due to their ability to capture nonlinear and complex relationships among traffic and temporal variables.

Serin et al. [1] applied ensemble learning models—AdaBoost, gradient boosting, random forest, Extra-Tree Regression, KNN, and SVM—to predict bus travel times in Istanbul, demonstrating that ensemble methods outperform single algorithms. Gal et al. [2] developed a hybrid model combining queueing theory and ML to enhance predictive reliability using real-world city data. Peterson et al. [3] introduced an LSTM-based framework capable of learning non-static spatio-temporal dependencies in bus networks, offering improved accuracy compared to traditional methods. Bai et al. [4] integrated support vector machines and Kalman filtering, where SVM predicted travel times and Kalman filtering refined the estimates using real-time data. Treethidtapath et al. [5] proposed a deep neural network (DNN) model to predict bus arrival times and benchmarked it against the Ordinary Least Squares (OLS) regression, achieving superior results. Chen et al. [6] designed an intelligent transport system using nine algorithms, including linear regression, KNN, SVR, gradient boosting, and deep architectures (LSTM and Bi-LSTM). Ashwini et al. [7] compared eight ML algorithms for bus travel time prediction, incorporating temporal and directional variables (time of day, day of week, and route direction). Servos et al. [8] applied Extremely Randomized Trees and AdaBoost methods, concluding that ensemble algorithms outperform mean-based approaches. Reddy et al. [10] showed that ML-based methods, especially SVR, improve prediction accuracy compared to traditional model-based approaches.

Moosavi et al. [11] evaluated tree-based ML models—Chi-Square Automatic Interaction Detection, random forest, and Gradient Boost Tree—and found them effective for routes with varying service frequencies. Wu et al. [12] introduced a hybrid ConvLSTM model integrating convolutional and recurrent structures, achieving enhanced temporal prediction performance. Lee et al. [13] developed a geo-convolutional LSTM model using both dwelling and transit times as features. He et al. [14] proposed a model that separately estimated riding and waiting times, combining LSTM with an interval-based historical average approach to improve temporal granularity.

In addition to ML-based models, several studies have relied on conventional statistical and optimization methods. Ceylan and Özcan [9] formulated a timetable-based transit assignment approach and optimized service frequencies using the harmony search algorithm. Bai et al. [4] and Reddy et al. [10] also represent early efforts to use regression-based prediction and hybrid statistical models. While these methods are computationally efficient and interpretable, they struggle to capture the nonlinear effects of urban dynamics and external influences such as weather variability.

Overall, studies on travel time prediction can be broadly categorized into statistical methods and machine learning methods, with most researchers employing one of these approaches in isolation. Statistical techniques offer interpretability, whereas machine learning algorithms provide flexibility in complex modeling and nonlinear relationships. However, the majority of existing studies rely mainly on operational or temporal variables—such as route geometry, passenger load, or departure intervals—while environmental and meteorological factors are largely ignored. Variables such as temperature, humidity, air pressure, precipitation, and wind speed, which directly affect vehicle performance and travel behavior, are rarely integrated into predictive models.

Furthermore, only a limited number of studies have combined statistical and machine learning techniques within a single framework or performed comparative robustness analyses under different preprocessing and parameter settings. Consequently, there remains a need for a hybrid modeling approach that merges the interpretability of regression-based models with the adaptability and predictive strength of modern machine learning algorithms, while explicitly accounting for meteorological influences.

To address this gap, the present study introduces an integrated hybrid framework that incorporates both statistical regression and machine learning models to predict public bus travel times. Distinct from previous research, this framework systematically evaluates the effects of weather variables—including temperature, humidity, wind speed, air pressure, and precipitation—on travel time estimation. In addition, sensitivity analyses are conducted by varying the  $k$  values in the K-nearest neighbors (KNN) algorithm and adjusting the threshold levels for outlier detection to assess model robustness. Using 1410 real-world observations, the empirical results demonstrate that weather-related variables significantly improve prediction accuracy and that the CatBoost algorithm consistently achieves the lowest mean squared error across different scenarios. This dual-layer methodology thus contributes both methodologically and practically to the advancement of intelligent, data-driven, and environmentally aware urban transportation systems.

### 3. Materials and Methods

There are many studies carried out using machine learning and regression models in the field of forecasting, as shown in Table 2. In this study, the methods used in machine learning and regression model are multiple regression, principle component regression, ridge regression, lasso regression, elastic net regression, K-nearest neighbor, multilayer perceptron, bagging trees, random forest, gradient boost machine, XGboost algorithm, and CatBoost algorithm.

**Table 2.** Machine learning and regression algorithm studies.

| Authors                  | Methods                                                                                      | Subject                                  | Main Conclusion/Weakness                                                                                       |
|--------------------------|----------------------------------------------------------------------------------------------|------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| Arslan and Ertuğrul [15] | Multiple Regression Models, Artificial Neural Networks Models                                | Electricity Consumption                  | Compared to regression and ANN; ANN achieved better fit but model lacked robustness for non-linear volatility. |
| Fumo and Biswas [16]     | Simple Linear Regression Model, Multiple Linear Regression Model                             | Energy Consumption                       | Regression captured linear dependence on temperature; ignored nonlinearity and multivariable interaction.      |
| Jang et al. [17]         | Linear Multiple Regression, Nonlinear Multiple Regression, Artificial Neural Networks (ANNs) | Geological Parameters                    | ANN performed better than regression for complex relationships; interpretability was weak.                     |
| Nguyen and Cripps [18]   | Multiple Regression Models, Artificial Neural Networks Models                                | House Sales                              | ANN outperformed regression; however, limited transparency and potential overfitting noted.                    |
| Talaat and Gamel [19]    | Correlation Coefficient, Multiple Linear Regression                                          | No. of Authors and No. of Citations      | Found strong correlation between authorship and citation; regression model lacked causal interpretation.       |
| Sun et al. [20]          | Random Forest Algorithm                                                                      | Research Octane Number                   | RF effectively modeled nonlinear fuel properties; dataset domain-specific, limiting generalizability.          |
| Adami et al. [21]        | Principal Component Analysis (PCA), Principal Component Regression (PCR)                     | Rheumatoid Arthritis                     | PCA/PCR identified key clinical factors; medical focus unrelated to transport forecasting.                     |
| Yan et al. [22]          | Principal Component Regression, Partial Least Squares Regression                             | Flight Load Analysis                     | Demonstrated efficiency in handling multicollinearity; limited transferability to dynamic datasets.            |
| Effendi et al. [23]      | Principal Component Regression                                                               | Farmer Exchange Rate                     | Showed PCR's strength in dimensionality reduction; agricultural data context only.                             |
| Sing et al. [24]         | Principal Component Regression, Partial Least Squares Regression                             | Piperine Contents in Black Pepper        | Compared regression methods for spectroscopy; not focused on time-dependent prediction.                        |
| Tahir and Ilyas [25]     | Robust Correlation-Based Regression, Robust Correlation Scaled Principal Regression          |                                          | Proposed robust approach for high-dimensional data; computationally heavy and untested in forecasting.         |
| Lettink et al. [26]      | Ridge Regression                                                                             | Health Indicator                         | Ridge regression reduced overfitting; performance limited by small sample.                                     |
| Zandi et al. [27]        | Locally Weighted Linear Regression Method                                                    | Three Large-Scale Precipitation Products | Achieved high spatial accuracy; model sensitive to regularization parameter.                                   |
| Zhang et al. [28]        | Polynomial Ridge Regression (RR) Algorithm                                                   | Atomic Nuclei                            | Efficient for physics-based prediction; domain-specific with limited cross-field relevance.                    |

Table 2. Cont.

| Authors            | Methods                                                                                  | Subject           | Main Conclusion/Weakness                                                                      |
|--------------------|------------------------------------------------------------------------------------------|-------------------|-----------------------------------------------------------------------------------------------|
| Zheng et al. [29]  | Kernel Ridge Regression, Support Vector Machine (SVM), Artificial Neural Networks (ANNs) | Wind Speed        | Kernel methods showed lowest error; required careful kernel tuning and large data.            |
| Song et al. [30]   | Lasso Regression, Long Short-Term Memory Model (LSTM)                                    | Gas Concentration | Lasso improved variable selection; LSTM achieved higher accuracy but at higher complexity.    |
| Li et al. [31]     | Lasso Regression, ARIMA, NARNN, LSTM                                                     | Carbon Price      | Combined classical and deep models; limited by temporal instability of financial data.        |
| Sharma et al. [32] | Linear Regression, Lasso Regression, Ridge Regression                                    | Strength of GGBFS | Regression performed well; lacks external validation and generalizability.                    |
| Didari et al. [33] | Lasso Regression                                                                         | Wheat Yield       | Identified key meteorological variables; effective locally but not tested on larger datasets. |
| Malakouti [34]     | Lasso Regression, Elastic Net Algorithms                                                 | Carbon Dioxide    | Elastic net provided stable results; small dataset constrained robustness assessment.         |

Arslan and Ertuğrul [15] analyzed the electricity price and compared the performance of multiple regression models and artificial neural network models.

Fumo and Biswas [16] performed a simple and multiple linear regression model along with a quadratic regression analysis. Multiple linear regression models using outdoor temperature and solar radiation offered an improved coefficient of determination.

Jang et al. [17] focused on the effects of geological parameters to the overbreak phenomenon by applying linear and nonlinear multiple regression and artificial neural networks (ANNs). The performance of these algorithms was evaluated by comparing correlations.

Nguyen and Cripps [18] compared the performance of two methods called artificial neural network and multiple regression analysis for predicting house sales. Artificial neural network solutions are better than multiple regression analysis solutions.

Talaat and Gamel [19] examined the effects of co-author count on the number of citations using Pearson's correlation coefficient and multiple linear regression. They compared the correlations between number of authors, number of countries, citation count, venue category, and year-from.

Sun et al. [20] used the random forest algorithm to predict the research octane number of gasoline. They indicated that the properties of gasoline effect the research octane number of gasoline, and complex relations can be modeled by using a random forest algorithm.

Adami et al. [21] introduced the clinical and bone metabolic risk factors using principal component analysis (PCA) and principal component regression (PCR). They found that some factors such as age, GC treatment, ACPA titer, etc. were negatively affected.

Yan et al. [22] compared different regression methods such as principal component regression and partial least squares regression for analyzing flight load. Then they demonstrate the efficiency and capabilities of these methods.

Effendi et al. [23] used principal component regression to predict the farmer exchange rate. Their analysis showed that harvest area, production, and the human development index affect the farmer exchange rate.

Sing et al. [24] compared principal component regression and partial least squares regression methods for estimating the piperine content in black pepper. The efficiency of these methods was evaluated by root mean square error and correlation coefficient.

Tahir and Ilyas [25] compared the result of the robust correlation-based regression and robust correlation scaled principal regression. Also, they proposed a model called as macro robust correlation scaled principal component regression, which can deal with missing values, cellwise outliers, row-wise outliers, high dimensions, and multicollinearity.

Lettink et al. [26] investigated the two-dimensional fused ridge estimator of the linear and logistic regression models. Then they developed an implementation of the cross-validation method. They showed the use of this method to predict health indicators.

Zandi et al. [27] used a locally weighted linear regression method for evaluating the potential of three large-scale products. They also used the L2 regularization for overcoming the multicollinearity problem.

Zhang et al. [28] introduced an algorithm which reduced the subspace and obtained optimal statistical machine learnings models. For simplicity, the polynomial ridge regression (RR) algorithm was used to learn the norm and Hamiltonian kernels of axially deformed configurations.

Zheng et al. [29] predicted wind speed using kernel ridge regression and then compared the results using methods such as support vector machine (SVM) and artificial neural networks (ANNs). The root means square error (RMSE) and root mean square ( $R^2$ ) were used to calculate the effectiveness of these algorithms.

Song et al. [30] proposed a model based on lasso regression to improve the applicability and effectivity of the model. They predicted the mine face gas concentration. After that, they compared the model with a long short-term memory prediction model.

Li et al. [31] combined a forecasting model using lasso regression and optimal integration. Also, ARIMA, NARNN, LSTM, and 11 other single forecasting models were used in their study.

Sharma et al. [32] compared compressive strength by using linear regression, lasso regression, and ridge regression. Some statistical measurements such as MSE, MAE, and RMSE were evaluated to calculate the performance of the methods.

Didari et al. [33] selected variables for predicting dryland wheat yield by using lasso regression. The result is that the lasso regression could be used reliably for each district considering the most effective meteorological parameters.

Malakouti [34] predicted the amount of carbon dioxide used by elastic net and lasso algorithms. After that, the methods were compared with the mean square error, mean absolute error, root mean square error, and mean absolute percentage error.

The originality of this study lies in its integration of statistical regression and advanced machine learning models in a single framework, with a specific focus on the role of meteorological variables in urban travel time prediction. Unlike previous studies, which often neglected environmental factors or focused solely on model performance, this research provides a systematic analysis of weather impacts and model sensitivity. Therefore, it contributes not only to methodological improvement but also to the practical deployment of smart transportation infrastructure.

## 4. Methods and Applications

The framework of the study is given in Figure 1. The study is divided into five phases: data collecting, data preprocessing, modeling, model evaluation, and model selection. A detailed description of each phase is presented in this section.

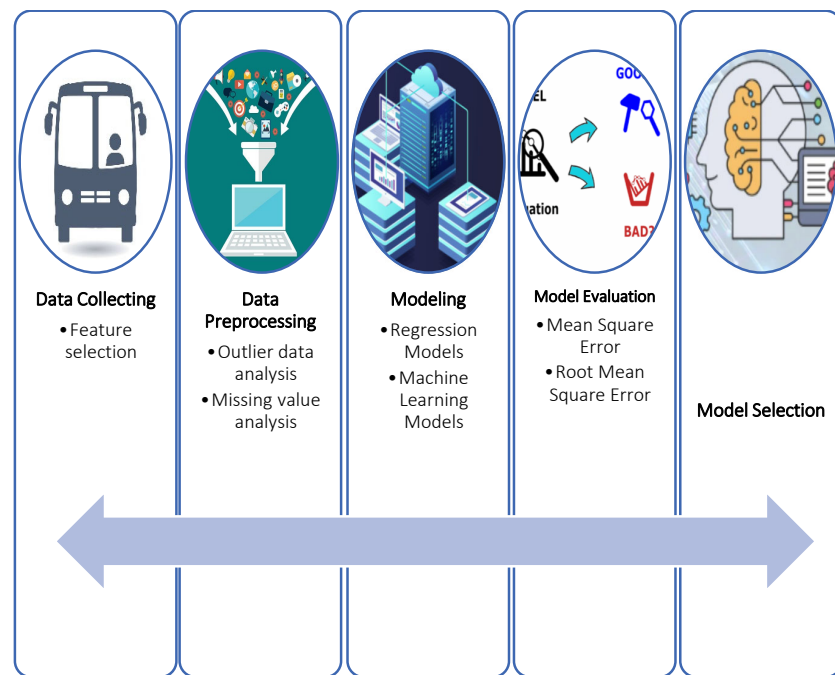


**Figure 1.** The framework of the study.

#### 4.1. Data Collecting

This stage is the first stage of the study, where the data is obtained. It can be described as the stage in which it is decided which characteristics the obtained data should have. In the study, the data of a city bus line belonging to a public transportation company operating in Turkey were used.

According to interviews with transportation company officials and experts, route number “R049 Organize Sanayi RSİ-Hörmetçi Aktarma” was deemed appropriate to carry out the works. The selected R049 route represents one of the longest and most frequently used lines in the Kayseri public transport network, connecting residential zones with the Organized Industrial Area. This route was chosen because it captures diverse traffic and meteorological conditions—ranging from urban congestion near the city center to open-road segments in suburban areas—thus offering representative variability for model training.

The study period (November 2021–July 2022) was deliberately selected to include both winter and summer months, allowing the analysis to capture the seasonal influence of weather conditions on travel time. Although the dataset is limited to a single route, it reflects operational and environmental heterogeneity typical of mid-sized Turkish cities, providing a meaningful foundation for generalization to similar contexts. The route map along with the bus-stop location is shown in Figure 2.

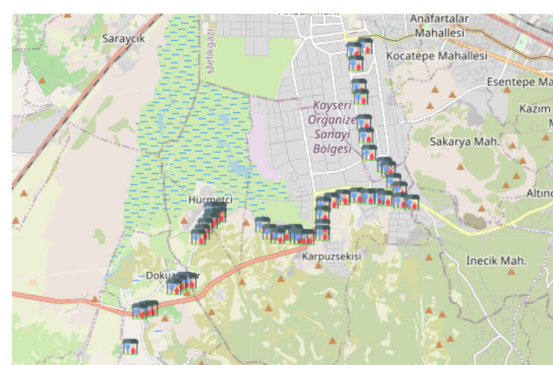


**Figure 2.** Route map of R049 (The symbols indicate the bus stop locations along the selected route).

In order to examine the effects of weather conditions on bus services, weather conditions of the period in question were also collected. Data between 5 November 2021 and 28 July 2022 are used in the study. A total of 1410 trips were observed for the specified time period. Meteorological data were retrieved from the Turkish State Meteorological Service (MGM) and synchronized with each bus trip according to its date and scheduled departure time. For each record, daily averages of temperature, humidity, air pressure, wind speed, and total daily precipitation were used to represent the environmental conditions during the corresponding day. In the study, we tried to predict the values that the travel time variable will take depending on the day, average temperature, humidity, precipitation, air pressure, average wind speed, scheduled starting time, and number of passengers variables.

The variables and definitions are as shown in Table 3.

**Table 3.** The variables and definitions.

| Variables               | Definitions                                                                                                         |
|-------------------------|---------------------------------------------------------------------------------------------------------------------|
| DAY                     | Shows the days of the week.<br>1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday, 6: Saturday, 7: Sunday. |
| AVERAGE_TEMPERATURE     | Shows the average temperature during the day.                                                                       |
| HUMIDITY                | Shows the humidity rate during the day.                                                                             |
| PRECIPITATION           | Shows the amount of precipitation per square meter during the day.                                                  |
| AIR_PRESSURE            | Shows the air pressure.                                                                                             |
| AVERAGE_WIND_SPEED      | Shows the average wind speed during the day.                                                                        |
| SCHEDULED_STARTING_TIME | Shows the bus departure time scheduled by the transportation company.                                               |
| NUMBER_OF_PASSENGERS    | Shows the number of passengers who boarded the bus on the specified expedition.                                     |

A snapshot of the data collected is given in Figure 3.

| DAY | AVERAGE_TEMPERATURE | HUMIDITY | PRECIPITATION | AIR_PRESSURE | AVERAGE_WIND_SPEED | SCHEDULED_STARTING_TIME | NUMBER_OF_PASSENGERS | TRAVEL_TIME |
|-----|---------------------|----------|---------------|--------------|--------------------|-------------------------|----------------------|-------------|
| 4   | 24.5                | 38.3     | 0.0           | 890.5        | 1.5                | 13                      | 19                   | 85          |
| 4   | 24.5                | 38.3     | 0.0           | 890.5        | 1.5                | 14                      |                      |             |
| 4   | 24.5                | 38.3     | 0.0           | 890.5        | 1.5                | 8                       | 24                   | 80          |
| 4   | 24.5                | 38.3     | 0.0           | 890.5        | 1.5                | 8                       |                      |             |
| 4   | 24.5                | 38.3     | 0.0           | 890.5        | 1.5                | 6                       | 29                   | 55          |
| 4   | 24.5                | 38.3     | 0.0           | 890.5        | 1.5                | 8                       | 30                   | 80          |

**Figure 3.** Data Sample of the City Bus Line.

For transportation companies, it is important that the travel times are estimated correctly, and the planning is carried out according to these estimations. The transportation company stated that the travel time should be 45 min. However, as shown in Figure 4, in practice, travel time was found to be 60.21 min.

|                                | count   | mean   | std   | min    | 25%    | 50%    | 75%    | max    |
|--------------------------------|---------|--------|-------|--------|--------|--------|--------|--------|
| <b>DAY</b>                     | 1409.00 | 3.60   | 1.82  | 1.00   | 2.00   | 4.00   | 5.00   | 7.00   |
| <b>AVERAGE_TEMPERATURE</b>     | 1410.00 | 9.01   | 9.36  | -13.30 | 1.10   | 8.00   | 18.60  | 27.30  |
| <b>HUMIDITY</b>                | 1410.00 | 62.31  | 15.61 | 14.30  | 51.30  | 64.30  | 73.80  | 94.80  |
| <b>PRECIPITATION</b>           | 1410.00 | 1.36   | 4.09  | 0.00   | 0.00   | 0.00   | 0.60   | 31.40  |
| <b>AIR_PRESSURE</b>            | 1410.00 | 891.48 | 4.15  | 880.30 | 888.60 | 891.20 | 894.80 | 902.30 |
| <b>AVERAGE_WIND_SPEED</b>      | 1403.00 | 1.41   | 0.83  | 0.30   | 1.00   | 1.30   | 1.60   | 6.60   |
| <b>SCHEDULED_STARTING_TIME</b> | 1410.00 | 12.70  | 4.82  | 6.00   | 8.00   | 13.00  | 17.00  | 19.00  |
| <b>NUMBER_OF_PASSENGERS</b>    | 1296.00 | 18.68  | 13.92 | 0.00   | 8.00   | 15.00  | 27.00  | 59.00  |
| <b>TRAVEL_TIME</b>             | 1296.00 | 60.21  | 27.20 | 3.00   | 40.00  | 53.00  | 81.00  | 337.00 |

Figure 4. The average travel time.

In addition, there were 841 observations that exceeded the standard travel time determined by the transportation company. The examples of these observations are shown in Figure 5.

|      | DAY  | AVERAGE_TEMPERATURE | HUMIDITY | PRECIPITATION | AIR_PRESSURE | AVERAGE_WIND_SPEED | SCHEDULED_STARTING_TIME | NUMBER_OF_PASSENGERS | TRAVEL_TIME |
|------|------|---------------------|----------|---------------|--------------|--------------------|-------------------------|----------------------|-------------|
| 0    | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 13                      | 19.00                | 85.00       |
| 2    | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8                       | 24.00                | 80.00       |
| 4    | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 6                       | 29.00                | 55.00       |
| 5    | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8                       | 30.00                | 80.00       |
| 7    | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 6                       | 29.00                | 53.00       |
| ...  | ...  | ...                 | ...      | ...           | ...          | ...                | ...                     | ...                  | ...         |
| 1395 | 1.00 | 9.20                | 49.50    | 0.00          | 897.40       | 0.50               | 19                      | 25.00                | 51.00       |
| 1398 | 7.00 | 8.10                | 51.00    | 0.00          | 899.20       | 0.70               | 18                      | 7.00                 | 46.00       |
| 1400 | 6.00 | 8.60                | 57.30    | 0.00          | 899.50       | 0.70               | 19                      | 12.00                | 53.00       |
| 1404 | NaN  | 8.60                | 57.30    | 0.00          | 899.50       | 0.70               | 16                      | 13.00                | 80.00       |
| 1405 | 5.00 | 8.70                | 67.00    | 0.00          | 900.20       | 0.60               | 8                       | 4.00                 | 46.00       |

841 rows × 9 columns

Figure 5. Examples of observations that exceeded the standard travel time.

#### 4.2. Data Preprocessing

The data of the observations obtained must be preprocessed before being used in any regression model or machine learning algorithm. Thus, more meaningful results can be obtained by removing noisy, erroneous, and redundant data from the dataset [7]. The data preprocessing phases carried out are as shown in Figure 6.

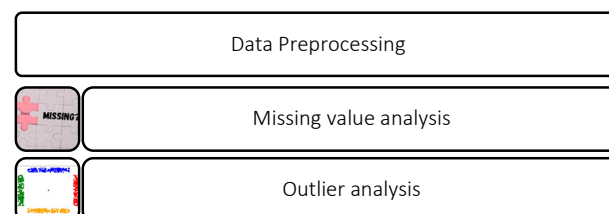


Figure 6. The data preprocessing phases.

Missing values occur when no data value is stored for the variable in an observation. Missing values have a significant effect on the conclusions. For this reason, researchers must try to eliminate this problem. Many approaches are available to tackle the problems imposed by missing values in data preprocessing. For example, the instances which have

missing values can be discarded; can be assigned a value such as mean or mode; or can be obtained by using a prediction method like KNN, random forest, etc. [34].

Outliers are patterns in data that do not conform to a well-defined notion of normal behavior [35]. While it is easy to identify outliers in datasets with a single variable, it is more difficult in datasets with multiple variables. For this reason, researchers have developed different methods. The local outlier factor (LOF) method is one of these methods. The basic idea of the LOF is to define the degree of an outlier of the point by assigning an object's deviation factor in the dataset.

By applying these steps, data is prepared for analysis. The travel time was the dependent variable and day, average temperature, humidity, precipitation, air pressure, average wind speed, scheduled starting time, and number of passengers were the independent variables.

Firstly, the missing values in the dataset were obtained for each variable. The bar chart is as shown in Figure 7.

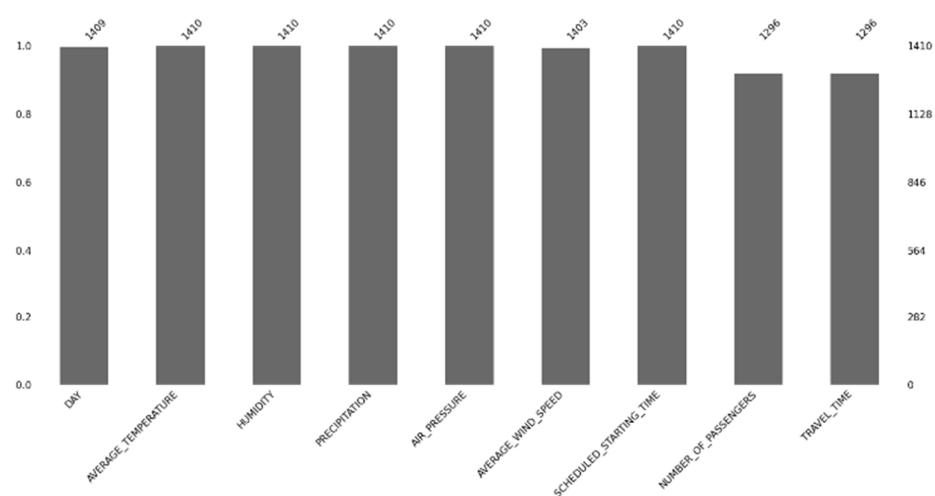


**Figure 7.** The bar chart for missing values.

Then the instances which had missing values were assigned a value obtained by using a prediction method which is called KNN. The k value, which is the model parameter, was accepted as 4. The missing values were completed according to the estimated values with this algorithm. The old data and the new data obtained by using the KNN method are shown in Figures 8 and 9.

A preliminary analysis was conducted to understand the correlation between the independent variables and the dependent variables. A heat map using the seaborn library in Python 3.10 (Python Software Foundation, Wilmington, DE, USA) is shown in Figure 10.

| Output View |      |                     |          |               |              |                    |                         |                      |             |
|-------------|------|---------------------|----------|---------------|--------------|--------------------|-------------------------|----------------------|-------------|
|             | DAY  | AVERAGE_TEMPERATURE | HUMIDITY | PRECIPITATION | AIR_PRESSURE | AVERAGE_WIND_SPEED | SCHEDULED_STARTING_TIME | NUMBER_OF_PASSENGERS | TRAVEL_TIME |
| 0           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 13                      | 19.00                | 85.00       |
| 1           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 14                      | NaN                  | NaN         |
| 2           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8                       | 24.00                | 80.00       |
| 3           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8                       | NaN                  | NaN         |
| 4           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 6                       | 29.00                | 55.00       |
| 5           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8                       | 30.00                | 80.00       |
| 6           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8                       | NaN                  | NaN         |
| 7           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 6                       | 29.00                | 53.00       |
| 8           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 13                      | 23.00                | 88.00       |
| 9           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 14                      | NaN                  | NaN         |

**Figure 8.** The old data in datasets.

| Output View |      |                     |          |               |              |                    |                         |                      |             |
|-------------|------|---------------------|----------|---------------|--------------|--------------------|-------------------------|----------------------|-------------|
|             | DAY  | AVERAGE_TEMPERATURE | HUMIDITY | PRECIPITATION | AIR_PRESSURE | AVERAGE_WIND_SPEED | SCHEDULED_STARTING_TIME | NUMBER_OF_PASSENGERS | TRAVEL_TIME |
| 0           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 13.00                   | 19.00                | 85.00       |
| 1           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 14.00                   | 20.63                | 82.86       |
| 2           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8.00                    | 24.00                | 80.00       |
| 3           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8.00                    | 27.00                | 80.00       |
| 4           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 6.00                    | 29.00                | 55.00       |
| 5           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8.00                    | 30.00                | 80.00       |
| 6           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 8.00                    | 27.00                | 80.00       |
| 7           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 6.00                    | 29.00                | 53.00       |
| 8           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 13.00                   | 23.00                | 88.00       |
| 9           | 4.00 | 24.50               | 38.30    | 0.00          | 890.50       | 1.50               | 14.00                   | 20.63                | 82.86       |

**Figure 9.** The new data in datasets.

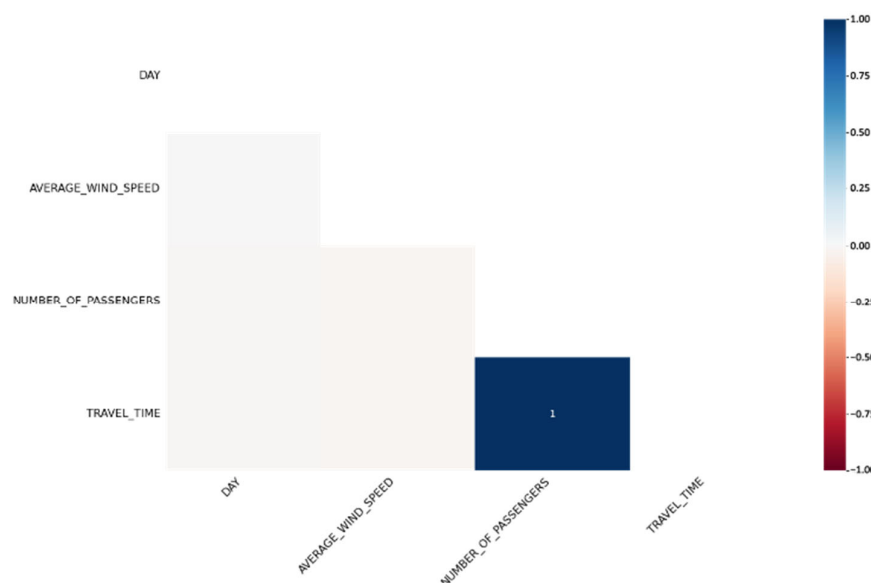


**Figure 10.** The heatmap for the correlation between the independent and dependent variables.

A heat map depicts values for a main variable of interest across two axis variables as a grid of colored squares. According to Figure 10, there is a very low correlation between almost all variables. However, the observed perfect correlation (1.0) between number\_of\_passengers and travel\_time suggests a potential data leakage issue. This may indicate that the passenger count is either directly derived from or strongly collinear with the actual travel time. Although this variable was retained for analysis due to its operational relevance, future studies should investigate its causal direction and consider alternative feature engineering strategies to mitigate overfitting risks.

In this study, one of the most prevalent outlier detection algorithms, LOF, was used. In order to apply this method, density values were calculated for each data point using the KNN algorithm. The data were sorted according to these density values, and the outliers were converted to the threshold value according to the order. Accordingly, 14 outliers were found in the dataset, as shown in Figure 11. In this context, each observation was assigned a local density score based on the local outlier factor (LOF) method. The data points were then sorted in ascending order of their density values, and those below the 13th-ranked density were defined as outliers. Therefore, the phrase “threshold = 13th data” indicates that the 13th lowest density value was used as the cutoff for outlier detection.

The threshold value obtained is as shown in Figure 12.

These 14 observations had the following data and were replaced by the threshold value. The outliers are as shown in Figure 13.

After the preprocessing phase, the information in the dataset is as shown in Figure 14.

```
aykirilar.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 14 entries, 48 to 1236
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                ---
0   DAY                                  14 non-null     float64
1   AVERAGE_TEMPERATURE                14 non-null     float64
2   HUMIDITY                           14 non-null     float64
3   PRECIPITATION                      14 non-null     float64
4   AIR_PRESSURE                       14 non-null     float64
5   AVERAGE_WIND_SPEED                14 non-null     float64
6   SCHEDULED_STARTING_TIME            14 non-null     float64
7   NUMBER_OF_PASSENGERS               14 non-null     float64
8   TRAVEL_TIME                        14 non-null     float64
```

Figure 11. Outlier detection results using the LOF method.

|     | DAY  | AVERAGE_TEMPERATURE | HUMIDITY | PRECIPITATION | AIR_PRESSURE | AVERAGE_WIND_SPEED | SCHEDULED_STARTING_TIME | NUMBER_OF_PASSENGERS | TRAVEL_TIME |       |
|-----|------|---------------------|----------|---------------|--------------|--------------------|-------------------------|----------------------|-------------|-------|
| 298 | 4.00 |                     | 19.10    | 47.50         | 23.50        | 888.60             | 1.60                    | 8.00                 | 13.00       | 44.00 |

Figure 12. Threshold value used for outlier classification (13th ranked density).

|      | DAY  | AVERAGE_TEMPERATURE | HUMIDITY | PRECIPITATION | AIR_PRESSURE | AVERAGE_WIND_SPEED | SCHEDULED_STARTING_TIME | NUMBER_OF_PASSENGERS | TRAVEL_TIME |
|------|------|---------------------|----------|---------------|--------------|--------------------|-------------------------|----------------------|-------------|
| 48   | 4.00 | 18.90               | 57.20    | 0.00          | 890.30       | 1.50               | 14.00                   | 6.00                 | 123.00      |
| 75   | 7.00 | 22.00               | 51.20    | 0.00          | 891.70       | 1.90               | 18.00                   | 20.00                | 337.00      |
| 84   | 5.00 | 19.90               | 54.90    | 0.00          | 888.80       | 1.50               | 6.00                    | 34.00                | 220.00      |
| 148  | 1.00 | 18.80               | 44.40    | 0.00          | 890.90       | 1.30               | 8.00                    | 2.00                 | 3.00        |
| 178  | 4.00 | 20.70               | 64.60    | 0.00          | 888.50       | 1.10               | 19.00                   | 42.00                | 309.00      |
| 179  | 4.00 | 20.70               | 64.60    | 0.00          | 888.50       | 1.10               | 19.00                   | 41.99                | 308.84      |
| 247  | 6.00 | 21.60               | 51.90    | 0.00          | 890.50       | 1.10               | 6.00                    | 52.00                | 210.00      |
| 297  | 4.00 | 19.10               | 47.50    | 23.50         | 888.60       | 1.60               | 8.00                    | 11.00                | 39.00       |
| 298  | 4.00 | 19.10               | 47.50    | 23.50         | 888.60       | 1.60               | 8.00                    | 13.00                | 44.00       |
| 328  | 1.00 | 18.70               | 58.20    | 31.40         | 888.10       | 1.80               | 8.00                    | 6.00                 | 38.00       |
| 329  | 1.00 | 18.70               | 58.20    | 31.40         | 888.10       | 1.80               | 8.00                    | 11.00                | 44.00       |
| 330  | 1.00 | 18.70               | 58.20    | 31.40         | 888.10       | 1.80               | 6.00                    | 47.00                | 117.00      |
| 334  | 6.00 | 23.50               | 30.00    | 0.00          | 888.30       | 1.10               | 6.00                    | 43.00                | 216.00      |
| 1236 | 1.00 | 4.30                | 71.50    | 0.00          | 894.70       | 0.80               | 17.00                   | 0.00                 | 5.00        |

Figure 13. Data points identified as outliers (indices shown as IDs).

|                         | count   | mean   | std   | min    | 25%    | 50%    | 75%    | max    |
|-------------------------|---------|--------|-------|--------|--------|--------|--------|--------|
| DAY                     | 1410.00 | 3.61   | 1.80  | 1.00   | 2.00   | 4.00   | 5.00   | 7.00   |
| AVERAGE_TEMPERATURE     | 1410.00 | 9.01   | 9.36  | -13.30 | 1.10   | 8.00   | 18.60  | 27.30  |
| HUMIDITY                | 1410.00 | 62.25  | 15.63 | 14.30  | 51.05  | 64.15  | 73.80  | 94.80  |
| PRECIPITATION           | 1410.00 | 1.49   | 4.36  | 0.00   | 0.00   | 0.00   | 0.60   | 31.40  |
| AIR_PRESSURE            | 1410.00 | 891.47 | 4.15  | 880.30 | 888.60 | 891.20 | 894.80 | 902.30 |
| AVERAGE_WIND_SPEED      | 1410.00 | 1.42   | 0.83  | 0.30   | 1.00   | 1.30   | 1.60   | 6.60   |
| SCHEDULED_STARTING_TIME | 1410.00 | 12.67  | 4.81  | 6.00   | 8.00   | 13.00  | 17.00  | 19.00  |
| NUMBER_OF_PASSENGERS    | 1410.00 | 18.88  | 13.40 | 0.00   | 8.21   | 16.00  | 26.99  | 59.00  |
| TRAVEL_TIME             | 1410.00 | 61.23  | 24.71 | 6.00   | 41.00  | 56.00  | 82.00  | 215.00 |

Figure 14. The information in the dataset.

#### 4.3. Modeling

The travel time prediction was obtained by using regression models and machine learning models for the selected route, such as multiple linear regression (MLR), principal component regression (PCR), ridge regression, lasso regression, elastic net regression, K-nearest neighbors, multilayer perceptron (MLP), classification and regression tree (CART), bagging trees regression, random forest regression, gradient boost machine (GBM), light GBM, and XGboost regression, as shown in Figure 15.

Since each bus trip is an independent record rather than a continuous time series, recurrent and sequential neural network architectures such as LSTM or GRU were not applicable to this dataset.

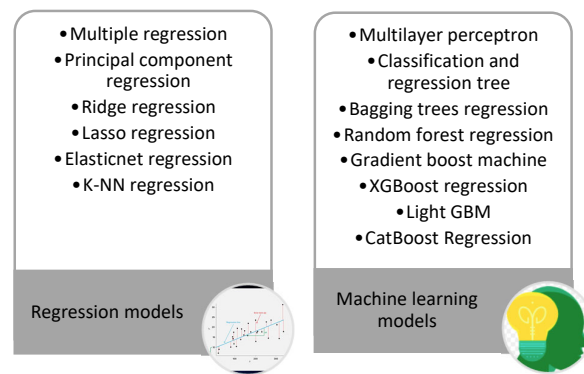


**Figure 15.** The regression models and machine learning models used in this study.

#### 4.3.1. Regression Models

Regression analysis is a methodology that allows for finding a functional relationship among response or dependent variables and predictor, explanatory, or independent variables. For complex systems, the regression analysis should be viewed as an iterative process [16]. The regression models used in this study are described below.

##### Multiple Linear Regression (MLR)

Multiple linear regression is the generalization of the simple linear regression model. The model in multiple linear regression allows for more than one predictor variable.

$$Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \dots + \alpha_p X_p + \epsilon \quad (1)$$

where  $Y$  is the response variable;  $X_1, X_2, \dots, X_p$  are the predictor variables with  $p$  as the number of variables;  $\alpha_0, \alpha_1, \dots, \alpha_p$  are the regression coefficients; and  $\epsilon$  is an error to account for the gap between predicted data and observed data.

##### Principal Component Regression (PCR)

Principal component regression is a basic, but very powerful, multivariate calibration method. When discussing multivariate analysis techniques, including PCR, three terms are often used: variance, vector, and projection. The term “vector” is used to describe a line segment in a coordinate system with a specific direction, and the term “projection” is used to describe the distance of a point along a vector.

##### Ridge Regression

Ridge regression is a popular parameter estimation method used to address collinearity problems frequently arising in multiple linear regression. The aim is to find the coefficients that minimize the mean absolute error by applying a penalty to the coefficients. It uses the  $L_2$  approach as shown in Equation (2).

$$SSE_{L_2} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \beta_j^2 \quad (2)$$

where  $\lambda$  is the setting parameter and  $\sum_{j=1}^p \beta_j^2$  is the penalty term.

The  $\lambda$  parameter is important, so the cross-validation method is used to find a good value.

##### Lasso Regression

The lasso is a method for regularizing a least squares regression. Suppose we have predictor measurements  $x_{ij}, j = 1, 2, \dots, p$  and an outcome measurement  $y_i$ , observed for cases  $i = 1, 2, \dots, N$ . The lasso fits a linear regression model by solving the optimization problem

$$\min_{\beta} \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 \quad (3)$$

subject to

$$\sum_{j=1}^p |\beta_j| \leq s \text{ [36].}$$

Ridge and lasso regression are not superior to each other. It uses the  $L_1$  approach as shown in Equation (3).

#### Elastic Net Regression

The elastic net regression is a form of regularized optimization for linear regression that provides a bridge between ridge regression and the lasso [37]. It uses both the  $L_1$  approach and  $L_2$  approach as shown in Equation (4).

$$SSE_{Enet} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^p \beta_j^2 + \lambda_2 \sum_{j=1}^p |\beta_j| \quad (4)$$

#### K-Nearest Neighbors

The K-nearest neighbor (KNN) method has widely been used in the applications of data mining and machine learning due to its simple implementation and distinguished performance [38]. KNN regression is also known as a lazy learner; it is a basic non-linear model which works on the principle of similarity [7]. KNN methods are as shown [39]:

1. Assigning an optimal k value with a fixed expert predefined value for all test samples.
2. Assigning different optimal k values for different test samples.

#### 4.3.2. Machine Learning Models

Machine learning (ML) is the scientific area of algorithms and statistical models that computer systems use to execute a specific duty without being explicitly programmed. In recent years, researchers have used machine learning methods in many areas. The machine learning models used in this study are described below.

##### Multilayer Perceptron (MLP)

Multilayer perceptron neural network models are a type of artificial neural network that use nonlinear functions. These models consist of three types: the input layer, the hidden layer(s), and the output layer. The input layer receives the input signals. There are weights connecting the input and hidden layers and weights connecting the hidden layers and output layer. Multilayer perceptron learns and makes predictions using these components [40].

##### Classification And Regression Tree (CART)

Classification and regression tree algorithms extract decision rules from features, which may include either numerical or categorical values, and builds a model to predict the target values [41]. CART produces the most explicit information compared to other machine learning algorithms [42].

CART algorithms include four basic steps; in the first step, a tree is built using recursive splitting of nodes. Each node is assigned to a class. In the second step, a maximal tree is produced and obtains information using the learning dataset. The next step consists of tree “pruning”, and the last step consists of optimal tree selection [43].

##### Bagging Trees Regression

A bagging tree chooses several subsets of data randomly with replacement from the training sample [44]. This tree is also known as bootstrap aggregation. When compared to boosting technique, a bagging tree has advantages of minimizing the decision tree training error. The aim is to reduce the variance of the predictor and avoid overfitting.

##### Random Forest Regression

Random forest regression is a machine learning model which is developed with the intention of modeling the output variables according to the inputs. This regression model is

aimed at predicting the structure of each tree and computing to utilize the training dataset. In this model, a lot of decision trees are obtained and tested [45].

#### **Gradient Boost Machine, Light GBM, and XGBoost Regression**

The gradient boost algorithm utilizes an ensemble, which is used to simplistically eliminate bias, noise, and variance, which dilute the effectiveness of the prediction model. The ensemble employs bagging and boosting techniques, which pre-build many independent models and which is tested and then implemented sequentially to allow the new models to learn from the mistakes of the previous models [46].

In machine learning algorithms, the light gradient boosting machine model is a more efficient ensemble learning model than other existing ensemble learning models. The performance of this algorithm is related to selecting the hyper parameter configuration. Thus, the hyper parameters must be chosen correctly. The hyperparameter choosing algorithms are grid search (Grid), random search, covariance matrix adaptive evolution strategy, and the tree-structured Parzen estimator [47].

An extreme gradient boost algorithm is a flexible model, and its hyperparameters can be tuned using soft computer algorithms. Thus, it is more accurate and faster than the other algorithms. In every step, the loss function can be reduced using the residual of the previous tree [48].

#### **4.3.3. The Applications of Algorithms in Prediction the Travel Time**

For all machine learning algorithms, hyperparameter optimization was performed using a grid search method with K-fold cross-validation ( $k = 5$ ). The hyperparameters tuned for each algorithm included the learning rate, maximum depth, the number of estimators for ensemble models (e.g., CatBoost, XGBoost), and the number of neurons and layers for neural networks. The best-performing parameters were selected based on the lowest mean squared error (MSE) on validation folds.

Feature importance was calculated for tree-based models using built-in importance metrics. In CatBoost, the most influential variables were number\_of\_passengers, scheduled\_starting\_time, and precipitation. This analysis highlights the role of weather-related parameters in travel time prediction. The travel time prediction was obtained by using the regression models and machine learning models for the selected route, such as multiple linear regression (MLR), principal component regression (PCR), ridge regression, lasso regression, elastic net regression, K-nearest neighbors, multilayer perceptron (MLP), classification and regression tree (CART), bagging trees regression, random forest regression, gradient boost machine (GBM), light GBM, and XGboost regression.

In this study, the hybrid modeling framework was implemented in two complementary stages. In the first stage, statistical regression models (MLR, PCR, ridge, lasso, and elastic net) were employed to establish baseline relationships and identify the relative influence of each explanatory variable. These models provide interpretable coefficients and assist in detecting multicollinearity or redundancy among predictors. In the second stage, machine learning algorithms (e.g., random forest, gradient boosting, XGBoost, CatBoost, and MLP) were trained on the same standardized dataset to capture nonlinear and interaction effects that cannot be represented by traditional regressions. The comparative evaluation of these two model groups formed the basis of the proposed hybrid framework, allowing both interpretability and high predictive performance to be jointly assessed.

In order to produce predictions with machine learning methods, it is necessary to divide the data into training and test data. There are various methods in the literature for this process. In this study, separation of training and test data was performed with the K-fold cross-validation method. The data were divided into training and test datasets

with a 0.75/0.25 split. The algorithm that gave the best results was obtained, and then, sensitivity analyses were carried out by changing various parameters.

The model tuning phase was also performed to find the optimal values of hyperparameters to maximize model performance. Hyperparameters in the machine learning algorithms are the set of variables whose values cannot be estimated by the model from the training data. Thus, model tuning can be called hyperparameter optimization. Thanks to model tuning, different hyperparameter values were obtained for each algorithm, and the options that gave the best results were obtained by using these hyperparameter values.

The main hyperparameters used for each algorithm are summarized as follows:

- KNN:  $k = 4$ –8;
- Random forest:  $n\_estimators = 200$ ,  $max\_depth = 10$ ;
- XGBoost:  $learning\_rate = 0.1$ ,  $n\_estimators = 300$ ,  $max\_depth = 8$ ;
- CatBoost:  $learning\_rate = 0.05$ ,  $depth = 8$ ,  $iterations = 500$ ;
- MLP:  $hidden\_layer\_sizes = (50, 25)$ ,  $activation = 'relu'$ ,  $solver = 'adam'$ .

All models were optimized using a grid search with 5-fold cross-validation.

#### 4.4. Modeling Evaluation and Model Selection

In the study, in addition to model tuning, sensitivity analyses were also carried out according to the change in the  $k$  value in the KNN algorithm and the change in the threshold values used to determine outliers.

Analysis 1: The Change in the Threshold Values Used to Determine Outliers.

According to the change in the threshold value used to determine the outliers, the mean squared error values of each algorithm are as seen in Figure 16. In this situation, the instances which had missing values were assigned a value obtained by using a prediction method called KNN, where the  $k$  value was accepted as 4.



**Figure 16.** Graphical representation of MSE values for each algorithm ( $k = 4$ ).

As seen in Figure 16, according to algorithms, the results are similar to each other. The values of each algorithm are shown in Table 4.

**Table 4.** Numerical mean squared error (MSE) values of each algorithm for  $k = 4$ . Red font indicates the lowest MSE value in each column (best model performance). Shaded background highlights the best-performing algorithm for each parameter setting.

| K Values | Threshold Value | Multiple Linear Regression | Principal Component Regression | Ridge Regression | Lasso Regression | Elastic Net Regression | K-Nearest Neighbors | Multilayer Perceptron | Classification and Regression Tree | Bagging Trees Regression | Random Forest Regression | Gradient Boost Machine | Xgboost Regression | Light GBM | Catboost |
|----------|-----------------|----------------------------|--------------------------------|------------------|------------------|------------------------|---------------------|-----------------------|------------------------------------|--------------------------|--------------------------|------------------------|--------------------|-----------|----------|
| 4        | 10              | 25.10                      | 26.47                          | 26.44            | 26.42            | 26.46                  | 26.26               | 24.06                 | 24.11                              | 23.12                    | 22.95                    | 23.19                  | 23.90              | 23.84     | 22.39    |
| 4        | 20              | 21.57                      | 23.12                          | 22.71            | 22.73            | 22.73                  | 20.41               | 20.51                 | 21.07                              | 19.68                    | 19.12                    | 19.13                  | 19.51              | 19.43     | 18.86    |
| 4        | 30              | 21.66                      | 22.10                          | 22.44            | 22.50            | 22.44                  | 20.56               | 18.86                 | 20.31                              | 18.50                    | 17.86                    | 18.11                  | 18.26              | 18.37     | 17.73    |
| 4        | 50              | 19.13                      | 20.04                          | 19.77            | 19.79            | 19.83                  | 18.96               | 18.57                 | 19.45                              | 18.65                    | 17.83                    | 17.89                  | 18.05              | 18.07     | 17.62    |
| 4        | 100             | 18.78                      | 21.07                          | 19.14            | 19.15            | 19.17                  | 18.23               | 18.03                 | 18.68                              | 17.88                    | 17.19                    | 17.26                  | 17.39              | 17.46     | 17.20    |

As shown in Table 4, nearly all of the minimum error values were obtained by using the CatBoost algorithm. Also, the minimum error values of each algorithm were obtained when the threshold value was 100.

According to the change in the threshold value in determining the outliers, the mean squared error values of each algorithm are as seen in Figure 17. In this situation, the instances which had missing values were assigned a value obtained by using a prediction method called KNN, where the  $k$  value was accepted as 5.

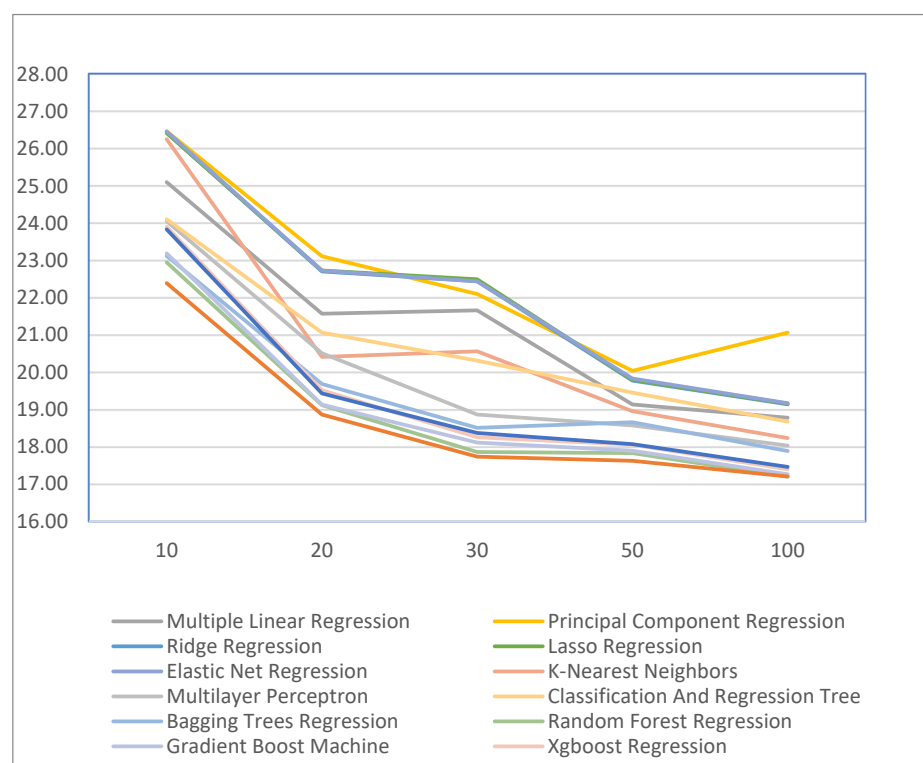


**Figure 17.** Graphical representation of MSE values for each algorithm ( $k = 5$ ).

As seen in Figure 17, according to the algorithms, the results are similar to each other. The values for each algorithm are shown in Table 5.

**Table 5.** Numerical mean squared error (MSE) values of each algorithm for  $k = 5$ . Red font indicates the lowest MSE value in each column (best model performance). Shaded background highlights the best-performing algorithm for each parameter setting.

| K Values | Threshold Value | Multiple Linear Regression | Principal Component Regression | Ridge Regression | Lasso Regression | Elastic Net Regression | K-Nearest Neighbors | Multilayer Perceptron | Classification and Regression Tree | Bagging Trees Regression | Random Forest Regression | Gradient Boost Machine | Xgboost Regression | Light GBM | Catboost |
|----------|-----------------|----------------------------|--------------------------------|------------------|------------------|------------------------|---------------------|-----------------------|------------------------------------|--------------------------|--------------------------|------------------------|--------------------|-----------|----------|
| 5        | 10              | 22.70                      | 25.79                          | 25.15            | 25.17            | 25.14                  | 25.27               | 24.20                 | 23.25                              | 23.24                    | 23.03                    | 22.88                  | 23.96              | 23.35     | 22.52    |
| 5        | 20              | 20.24                      | 25.14                          | 21.60            | 21.62            | 21.63                  | 20.47               | 20.66                 | 20.93                              | 20.08                    | 19.27                    | 19.44                  | 19.60              | 19.65     | 19.09    |
| 5        | 30              | 20.01                      | 21.62                          | 20.52            | 20.50            | 20.54                  | 19.70               | 18.64                 | 19.27                              | 18.67                    | 17.84                    | 17.96                  | 17.83              | 17.99     | 17.95    |
| 5        | 50              | 20.81                      | 21.77                          | 21.33            | 21.29            | 21.36                  | 19.21               | 18.44                 | 19.33                              | 18.64                    | 17.86                    | 18.05                  | 18.09              | 18.05     | 17.59    |
| 5        | 100             | 20.17                      | 22.26                          | 19.94            | 19.93            | 19.98                  | 18.33               | 18.18                 | 18.37                              | 17.82                    | 17.02                    | 17.10                  | 17.07              | 17.20     | 16.80    |

As shown in Table 5, nearly all of the minimum error values were obtained by using the CatBoost algorithm. Also, nearly all of the minimum error values of each algorithm were obtained when the threshold value was 100.

Analysis 2: The Change in the K Value in the KNN Algorithm.

Thus, while the threshold values were constant, the error values were analyzed according to the change in the  $k$  values. The results are as shown in Table 6.

**Table 6.** The mean squared error values of each algorithm (changing  $k$  values). Shaded background highlights the best-performing algorithm for each parameter setting.

| K Values | Threshold Value | Multiple Linear Regression | Principal Component Regression | Ridge Regression | Lasso Regression | Elastic Net Regression | K-Nearest Neighbors | Multilayer Perceptron | Classification and Regression Tree | Bagging Trees Regression | Random Forest Regression | Gradient Boost Machine | Xgboost Regression | Light GBM | Catboost |
|----------|-----------------|----------------------------|--------------------------------|------------------|------------------|------------------------|---------------------|-----------------------|------------------------------------|--------------------------|--------------------------|------------------------|--------------------|-----------|----------|
| 4        | 100             | 18.7758                    | 21.0668                        | 19.1381          | 19.1539          | 19.1665                | 18.2300             | 18.0274               | 18.6750                            | 17.8805                  | 17.1935                  | 17.2576                | 17.3929            | 17.4580   | 17.1993  |
| 5        | 100             | 20.1716                    | 22.2621                        | 19.9354          | 19.9318          | 19.9821                | 18.3324             | 18.1830               | 18.3747                            | 17.8195                  | 17.0247                  | 17.1034                | 17.0672            | 17.1977   | 16.7975  |
| 6        | 100             | 19.9567                    | 26.8876                        | 20.3238          | 20.3413          | 20.3111                | 18.1531             | 18.0175               | 18.6016                            | 18.3199                  | 17.2081                  | 17.3279                | 17.1931            | 17.3498   | 16.9579  |
| 7        | 100             | 18.9454                    | 19.1356                        | 19.1208          | 19.147           | 19.1613                | 18.1592             | 17.7727               | 18.6512                            | 18.0104                  | 17.1203                  | 17.1721                | 17.3995            | 17.2529   | 16.9975  |
| 8        | 100             | 18.9375                    | 20.9714                        | 19.1898          | 19.2011          | 19.2376                | 18.0917             | 17.8998               | 18.3897                            | 18.1027                  | 17.0496                  | 17.1558                | 17.0621            | 17.4066   | 16.7994  |
| 9        | 100             | 20.8862                    | 21.3898                        | 21.3553          | 21.4084          | 21.3990                | 18.3022             | 18.0237               | 18.2731                            | 17.8038                  | 16.9951                  | 17.1753                | 17.1409            | 17.2827   | 16.7951  |
| 10       | 100             | 20.8755                    | 21.3905                        | 21.3571          | 21.4095          | 21.3993                | 18.5103             | 18.1871               | 18.7065                            | 17.3708                  | 17.0292                  | 17.0780                | 17.1868            | 17.1771   | 16.722   |
| 15       | 100             | 18.9470                    | 19.6539                        | 19.2751          | 19.2633          | 19.2960                | 18.2728             | 17.5958               | 18.6351                            | 17.4969                  | 17.0611                  | 16.9917                | 17.2837            | 17.1807   | 16.9158  |
| 20       | 100             | 20.7664                    | 21.2142                        | 21.2089          | 21.2622          | 21.2595                | 18.5742             | 18.2516               | 18.6117                            | 17.2602                  | 16.9403                  | 17.4187                | 17.179             | 17.2928   | 16.7505  |
| 30       | 100             | 18.9485                    | 24.6539                        | 19.3945          | 19.4052          | 19.3943                | 18.1641             | 17.9939               | 17.9480                            | 17.6267                  | 17.1106                  | 17.0197                | 17.0441            | 16.9391   | 16.7295  |
| 40       | 100             | 19.5432                    | 21.0539                        | 19.8611          | 19.8998          | 19.9326                | 18.3028             | 18.176                | 18.0005                            | 17.1836                  | 17.1334                  | 16.9885                | 17.1486            | 17.1024   | 16.6134  |
| 50       | 100             | 19.2573                    | 19.5114                        | 19.2834          | 19.2550          | 19.2904                | 18.1027             | 17.7176               | 18.1323                            | 17.5025                  | 16.8580                  | 17.1984                | 17.1723            | 16.9679   | 16.5454  |
| 100      | 100             | 18.7193                    | 18.7824                        | 18.8567          | 18.8676          | 18.8943                | 18.0978             | 18.0318               | 17.7718                            | 17.6758                  | 16.9005                  | 16.9857                | 17.1425            | 17.1054   | 16.7129  |

As illustrated in Table 6 and Figure 18, the CatBoost algorithm consistently produced the lowest mean squared error (MSE) values across all parameter settings. This superior performance indicates that gradient boosting-based approaches, particularly CatBoost, exhibited higher robustness and generalization capability compared to other regression and machine learning models under varying preprocessing conditions.

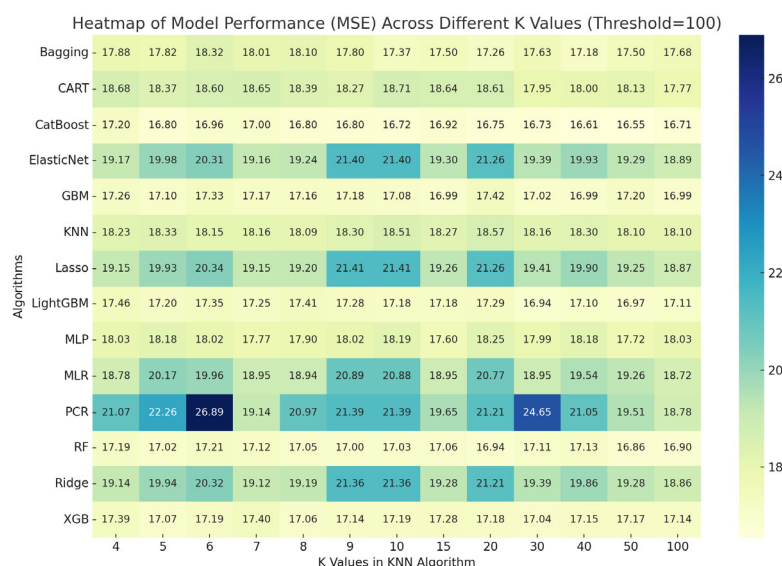


**Figure 18.** Heatmap of model performance (MSE) across different  $k$  values.

To statistically validate these performance differences, a Friedman test was conducted based on the MSE values of all algorithms. The test results as shown in Table 7 confirmed that the differences among the models were statistically significant ( $\chi^2 = 148.54$ ,  $df = 12$ ,  $p < 0.001$ ), rejecting the null hypothesis of equal treatment effects. The test was conducted at a 95% confidence level.

**Table 7.** Results of the Friedman test for model performance comparison.

| Hypothesis             | Description                                |            |
|------------------------|--------------------------------------------|------------|
| Null hypothesis        | $H_0$ : All treatment effects are zero     |            |
| Alternative hypothesis | $H_1$ : Not all treatment effects are zero |            |
| DF                     | Chi-Square                                 | $p$ -Value |
| 12                     | 148.54                                     | 0.000      |

Considering the values as shown in Table 6, the CatBoost algorithm achieved the lowest median error value among all models. Therefore, it was selected as the reference model in the Wilcoxon signed-rank test. The test results ( $p = 0.002$  for all comparisons) confirmed that CatBoost performed statistically better than the other algorithms.

## 5. Conclusions

This study investigates the impact of weather conditions on public bus travel times using both traditional regression and modern machine learning algorithms. Unlike prior studies, it combines statistical and machine learning techniques in a unified framework while incorporating weather variables often overlooked in the literature.

The analysis reveals that the CatBoost algorithm consistently yielded the lowest mean squared error across different preprocessing scenarios, including variations in KNN  $k$  values and outlier threshold settings. The inclusion of meteorological variables yielded significant improvements in model performance. These results underline the predictive power of environmental features in urban transportation modeling. Conceptually, these findings support a causal framework where meteorological factors act as exogenous drivers of variability in travel time through both mechanical (e.g., road friction, engine load) and behavioral (e.g., driver caution, passenger flow) pathways. This aligns with transport system resilience theory, which emphasizes the sensitivity of urban mobility performance to

environmental disturbances. Thus, the inclusion of weather parameters not only enhances predictive accuracy but also enriches the explanatory depth of the model. Additionally, the robustness of the proposed approach was validated through sensitivity analyses, where CatBoost maintained superior accuracy across varying  $k$ -values and outlier thresholds. This suggests that gradient boosting-based models are more resilient to noise and variability in transport-related data.

To further validate these findings, non-parametric statistical analyses were conducted. The Friedman test confirmed that the performance differences among algorithms were statistically significant ( $\chi^2 = 148.54$ ,  $df = 12$ ,  $p < 0.001$ ), rejecting the null hypothesis of equal treatment effects. Following this, pairwise Wilcoxon signed-rank tests ( $p = 0.002$  for all comparisons) revealed that CatBoost significantly outperformed the other algorithms. These statistical results substantiate the robustness of the proposed approach and strengthen the conclusion that CatBoost offers the most reliable performance for travel time prediction under varying preprocessing and parameter settings.

The hybrid modeling strategy adopted in this study bridges the interpretability of regression-based models with the predictive flexibility of machine learning algorithms, offering a balanced and transparent framework for travel time estimation.

Beyond the empirical improvement in predictive performance, the study also provides a theoretical contribution by clarifying how meteorological variability systematically influences travel time dynamics through physical, operational, and behavioral mechanisms.

The selected R049 route was intentionally used because it connects industrial and residential regions under varying meteorological conditions and traffic densities, offering a representative testbed for the proposed modeling framework.

However, a critical limitation is the relatively small dataset ( $N = 1410$ ), which corresponds to a single bus route observed over a nine-month period. Although the dataset size was sufficient for cross-validation and sensitivity analyses across multiple algorithms, future research should employ larger, multi-route datasets to enhance robustness and generalizability.

From a practical standpoint, the developed framework can guide transport authorities in designing data-driven scheduling and monitoring systems. By integrating such predictive models into real-time control centers, transit agencies could dynamically adjust route frequencies or provide accurate arrival information to passengers under varying weather conditions. These applications would enhance service reliability and support sustainable urban mobility policies.

Future research should expand the dataset over multiple cities and seasons, incorporate real-time traffic data, and apply more advanced time-series models such as LSTM or Transformer architectures. In addition, feature importance analyses and SHAP-based interpretability tools could be used to uncover the relative influence of each weather parameter. These improvements would contribute to the development of robust, real-time, and adaptive public transport management systems.

In this study, the performance of regression and machine learning algorithms were analyzed. In addition to this study, not only for the  $k$  values and threshold values but also the different parameter values of each algorithm, sensitivity analyses were performed. Thus, we aimed to achieve lower error values.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available in the article.

**Conflicts of Interest:** The author declares no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

|         |                                           |
|---------|-------------------------------------------|
| ANN     | Artificial Neural Network                 |
| ARIMA   | Auto-Regressive Integrated Moving Average |
| CART    | Classification and Regression Tree        |
| GBM     | Gradient Boosting Machine                 |
| KNN     | K-Nearest Neighbors                       |
| LOF     | Local Outlier Factor                      |
| LSTM    | Long Short-Term Memory                    |
| MAE     | Mean Absolute Error                       |
| ML      | Machine Learning                          |
| MLP     | Multilayer Perceptron                     |
| MLR     | Multiple Linear Regression                |
| MSE     | Mean Squared Error                        |
| PCR     | Principal Component Regression            |
| RMSE    | Root Mean Square Error                    |
| SVR     | Support Vector Regression                 |
| XGBoost | Extreme Gradient Boosting                 |

## References

1. Serin, F.; Alisan, Y.; Erturkler, M. Predicting bus travel time using machine learning methods with three-layer architecture. *Measurement* **2022**, *198*, 111403. [\[CrossRef\]](#)
2. Gal, A.; Mandelbaum, A.; Schnitzler, F.; Senderovich, A.; Weidlich, M. Traveling time prediction in scheduled transportation with journey segments. *Inf. Syst.* **2017**, *64*, 266–280. [\[CrossRef\]](#)
3. Peterson, N.C.; Rodrigues, F.; Pereira, F.C. Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Syst. Appl.* **2019**, *120*, 426–435. [\[CrossRef\]](#)
4. Bai, C.; Peng, Z.R.; Lu, Q.C.; Sun, J. Dynamic bus travel time prediction models on road with multiple bus routes. *Comput. Intell. Neurosci.* **2015**, *2015*, 1–10. [\[CrossRef\]](#)
5. Treethidaphat, W.; Atikom, W.P.; Khaimook, S. Bus Arrival Time Prediction at Any Distance of Bus Route Using Deep Neural Network Model. In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC): Workshop, Yokohama, Japan, 16–19 October 2017; pp. 757–762. [\[CrossRef\]](#)
6. Chen, M.Y.; Chiang, H.S.; Yang, K.J. Constructing cooperative intelligent transport systems for travel time prediction with deep learning approaches. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16590–16599. [\[CrossRef\]](#)
7. Ashwini, B.P.; Sumathi, R.; Sudhira, H.S. Bus Travel Time Prediction: A Comparative Study of Linear and Non-Linear Machine Learning Models. *J. Phys. Conf. Ser.* **2022**, *2161*, 012053. [\[CrossRef\]](#)
8. Servos, N.; Liu, X.; Teucke, M.; Freitag, F. Travel Time Prediction in a Multimodal Freight Travel Time Prediction in a Multimodal Freight Learning Algorithms. *Logistics* **2020**, *4*, 1. [\[CrossRef\]](#)
9. Ceylan, H.; Özcan, T. Optimization of Service Frequencies in Bus Networks with Harmony Search Algorithm: An Application on Mandl's Tet Network. *Pamukkale Univ. J. Eng. Sci.* **2018**, *24*, 1107–1116. [\[CrossRef\]](#)
10. Reddy, K.K.; Kumar, B.; Vanajakshi, L. Bus travel time prediction under high variability conditions. *Curr. Sci.* **2016**, *111*, 700–711. [\[CrossRef\]](#)
11. Mossavi, S.M.H.; Aghaabbasi, M.; Yuen, C.W. Evaluation of Applicability and Accuracy of Bus Travel Time Prediction in High and Low Frequency Bus Routes Using Tree-Based ML Techniques. *J. Soft Comput. Civ. Eng.* **2023**, *7*, 74–97.
12. Wu, J.; Wu, Q.; Cai, C. Towards Attention-Based Convolutional Long Short-Term Memory for Travel Time Prediction of Bus Journeys. *Sensors* **2020**, *20*, 3354. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Lee, G.; Choo, S.; Choi, S.; Lee, H. Does the Inclusion of Spatio-Temporal Feature Improve Bus Travel Time Predictions? A Deep Learning-Based Modelling Approach. *Sustainability* **2022**, *14*, 7431. [\[CrossRef\]](#)
14. He, P.; Jiang, G.; Lam, S.K.; Tang, D. Travel-Time Prediction of Bus Journey with Multiple Bus Trips. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 4192–4205. [\[CrossRef\]](#)
15. Arslan, B.; Ertuğrul, İ. Çoklu Regresyon, ARIMA ve Yapay Sinir Ağı Yöntemleri ile Türkiye Elektrik Piyasasında Fiyat Tahmin ve Analizi. *Yönetim Ve Ekon. Araştırmaları Derg.* **2022**, *20*, 331–353. [\[CrossRef\]](#)
16. Fumo, N.; Biswas, M. Regression analysis for prediction of residential energy consumption. *Renew. Sustain. Energy Rev.* **2015**, *47*, 332–343. [\[CrossRef\]](#)

17. Jang, H.; Topal, E. Optimizing overbreak prediction based on geological parameters comparing multiple regression analysis and artificial neural network. *Tunn. Undergr. Space Technol.* **2013**, *38*, 161–169. [\[CrossRef\]](#)
18. Nguyen, N.; Cripps, A. Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks. *J. Real Estate Res.* **2021**, *22*, 313–336. [\[CrossRef\]](#)
19. Talaat, F.M.; Gamel, S.A. Predicting the impact of no. of authors on no. of citations of research publications based on neural networks. *J. Ambient Intell. Humaniz. Comput.* **2022**, *14*, 8499–8509. [\[CrossRef\]](#)
20. Sun, X.; Zhang, F.; Liu, J.; Duan, X. Predicting of Gasoline Reseach Octane Number Using Multiple Feature Machine Learning Models. *Fuel* **2023**, *333*, 126510. [\[CrossRef\]](#)
21. Adami, G.; Orsolini, G.; Fassio, A.; Viapiana, O.; Sorio, E.; Benini, C.; Gatti, D.; Bertelle, D.; Rossini, M. POS0474 Factirs Associated with Erosive Rheumatoid Arthritis, A Multimarker Principal Component analysis (PCA) and Principal Component Regression (PCR) Analysis. *Ann. Rheum. Dis.* **2023**, *82*, 497–498. [\[CrossRef\]](#)
22. Yan, Q.; Yang, C.; Wan, Z. A Comparative Regression Analysis between Principal Component and Partial Least Squares Methods for Flight Load Calculation. *Appl. Sci.* **2023**, *13*, 8428. [\[CrossRef\]](#)
23. Effendi, M.; Ardhyatirta, R.; Angelina, S.G.; Ohyver, M. Predict Farmer Exchange Rate in the Food Crop Sector Using Principal Component Regression. *Enthusiastic Int. J. Appl. Stat. Data Sci.* **2023**, *3*, 74–84. [\[CrossRef\]](#)
24. Sing, D.; Dastidar, S.G.; Akram, W.; Guchhait, S.; Jana, S.N.; Banerjee, S.; Mukherjee, P.K.; Bandyopadhyay, R. A Comparative Study Between Partial Least Squares and Principal Component Regression for Nondestructive Quantification of Piperine Contents in Black Pepper by Raman Spectroscopy. In *Smart Sensors Measurement and Instrumentation*; Springer: Singapore, 2023.
25. Tahir, A.; Ilyas, M. Robust Correlation Scaled Principal Component Regression. *Hacet. J. Math. Stat.* **2023**, *52*, 459–486. [\[CrossRef\]](#)
26. Lettink, A.; Chinapaw, M.; Wieringen, W.N. Two-Dimensional Fused Targeted Ridge Regression for Health Indicator Prediction From Accelerometer Data. *J. R. Stat. Soc. Ser. C Appl. Stat.* **2023**, *72*, 1064–1078. [\[CrossRef\]](#)
27. Zandi, O.; Nasserli, M.; Zahraie, B. A Locally Weighted Linear Ridge Regression Framework for Spatial Interpolation of Monthly Precipitation over an Orographically Complex Area. *Int. J. Climatol.* **2023**, *43*, 2601–2622. [\[CrossRef\]](#)
28. Zhang, X.; Lin, W.; Yao, J.; Jiao, C.; Romero, A.; Rodriguez, T.; Hergert, H. Optimization of the generator coordinate method with machine-learning techniques for nuclear spectra and neutrinoless double-  $\beta$  decay: Ridge regression for nuclei with axial deformation. *Pyhsical Rev. C Cover. Nucl. Phys.* **2023**, *107*, 024304. [\[CrossRef\]](#)
29. Zheng, Y.; Ge, Y.; Muhsen, S.; Wang, S.; Elkamchouchi, D.H.; Ali, E.; Ali, H. New Ridge Regression, Artificial Neural Networks and Support Vector Machine for Wind Speed Prediction. *Adv. Eng. Softw.* **2023**, *179*, 103426. [\[CrossRef\]](#)
30. Song, S.; Chen, J.; Ma, L.; Zhag, L.; He, S.; Du, G.; Wang, J. Research on a Working Face Gas Concentration Prediction Model Based on LASS\_RNN time Series Data. *Heliyon* **2023**, *9*, e14864. [\[CrossRef\]](#)
31. Li, Y.; Yang, R.; Wang, X.; Zhu, J.; Song, N. Carbon Price Combination Forecasting Model Based on Lasso Regression And Optimal Integration. *Sustainability* **2023**, *15*, 9354. [\[CrossRef\]](#)
32. Sharma, U.; Gupta, N.; Verma, M. Prediction of Compressive Strength of GGBFS and Flyash-based Geopolymer Composite by Linear Regression, Lasso Regression, and Ridge Regression. *Asian J. Civ. Eng.* **2023**, *24*, 3399–3411. [\[CrossRef\]](#)
33. Didari, S.; Talebnejad, R.; Bahrami, M.; Mahmoudi, M.R. Dryland Farming Wheat Yield Prediction Using the Lasso Regression Model and Meteorological Variables in Dry and Semi-dry Region. *Stoch. Environ. Res. Risk Assess.* **2023**, *37*, 3967–3985. [\[CrossRef\]](#)
34. Garcia, S.; Gallego, S.R.; Luengo, J.; Benitez, J.M.; Herrera, F. Big Data Preprocessing: Methods and Prospects. *Big Data Anal.* **2016**, *1*, 9. [\[CrossRef\]](#)
35. Singh, K.; Upadhyaya, S. Outlier Detection: Applications and Techniques. *Int. J. Comput. Sci.* **2012**, *9*, 307–324.
36. Hastie, T.; Taylor, J.; Tibshirani, R.; Walther, G. Forward Stagewise Regression and The Monotone Lasso. *Electron. J. Stat.* **2007**, *1*, 1–29. [\[CrossRef\]](#)
37. Hans, C. Elastic Net Regression Modeling with the Orthant Normal Prior. *J. Am. Stat. Assoc.* **2011**, *106*, 1383–1393. [\[CrossRef\]](#)
38. Zhang, S.; Li, X.; Zong, M.; Zhu, X. Learning k for kNN Classification. *ACM Trans. Intell. Syst. Technol.* **2017**, *8*, 1–19. [\[CrossRef\]](#)
39. Mahesh, B. Machine Learning Algorithms-A Review. *Int. J. Sci. Res.* **2020**, *9*, 381–386. [\[CrossRef\]](#)
40. Afzal, S.; Ziapour, B.M.; Shokri, A.; Shakibi, H.; Sobhani, B. Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms. *Energy* **2023**, *282*, 128446. [\[CrossRef\]](#)
41. Ozcan, M.; Peker, S. Classification and regression tree algorithm for heart disease modeling and prediction. *Helathcare Anal.* **2023**, *3*, 100130. [\[CrossRef\]](#)
42. Kori, G.S.; Kakkasageri, M.S. Classification and regression tree (CART) based resource allocation scheme for wireless sensor networks. *Comput. Commun.* **2023**, *197*, 242–254. [\[CrossRef\]](#)
43. Lewis, R.J. An Introduction to Classification and Regression Tree (CART) Analysis. In Proceedings of the Annual Meeting of the Society for Academic Emergency Medicine, San Francisco, CA, USA, 22–25 May 2000.
44. Karthan, M.K.; Kumar, P.N. Prediction of IRNSS User Position using Regression Algorithms. In Proceedings of the International Conference on Machine Intelligence for GeoAnalytics and Remote Sensing, Hyderabad, India, 27–29 January 2023.

45. Rajkovic, D.; Jeromela, A.M.; Pezo, L.; Loncar, B.; Grahovac, N.; Spika, A.K. Artificial neural network and random forest regression models for modelling fatty acid and tocopherol content in oil of winter rapeseed. *J. Food Compos. Anal.* **2023**, *115*, 105020. [[CrossRef](#)]
46. Felix, A.Y.; Sasipraba, T. Flood Detection Using Gradient Boost Machine Learning Approach. In Proceedings of the 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 11–12 December 2019.
47. Guo, J.; Yun, S.; Meng, Y.; He, N.; Ye, D.; Zhao, Z.; Jia, L.; Yang, L. Prediction of heating and cooling loads based on light gradient boosting machine algorithms. *Build. Environ.* **2023**, *236*, 110252. [[CrossRef](#)]
48. Ali, Z.H.; Burhan, A.M. Hybrid machine learning approach for construction cost estimation: An evaluation of extreme gradient boosting model. *Asian J. Civ. Eng.* **2023**, *24*, 2427–2442. [[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.