Content list available at ITC

Techno-Science Research Journal

Journal Homepage: http://techno-srj.itc.edu.kh/

Techno-Science Research Journal

Bus Arrival Time Prediction Using Machine Learning Approaches

Phannet Pov[1*], Sokkhey Phauk[1], Dona Valy[2], Narith Saum[3]

[1] *Department of Applied Mathematics and Statistics, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

[2] *Department of Information and Communication Engineering, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

[3] *Department of Transport and Infrastructure Engineering, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, Phnom Penh, Cambodia*

Received: 12 July 2024; Revised: 20 August 2024; Accepted: 31 August 2024; Available online: 30 April 2025

Abstract: *Efficient public transportation systems are critical for urban mobility, requiring precise bus arrival time predictions to enhance service quality and passenger satisfaction. This study thoroughly investigates predictive modeling utilizing machine learning methodologies to forecast bus arrival times along a specific route in Phnom Penh, Cambodia. The dataset contains historical bus arrival data from a specific route, including journey duration, hour of day, day of the week, distance, speed, current bus stop, next bus stop, and weather conditions. The predictive models of this research are built using machine learning methods such as linear regression, XGBoost, support vector machine (SVM), k-nearest neighbors (KNN), and artificial neural network (ANN). These algorithms are used to create prediction models, which are fine-tuned to yield the most accurate estimates of bus arrival times. The usefulness of these models is systematically tested using common performance metrics such as the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), offering a full assessment of predicting. Extensive experimentation data shows the best performance of XGBoost model, which consistently outperforms other machine learning methods in accurately estimating bus arrival times, with an MAE of 16 seconds, an RMSE of 28.03 seconds, and a MAPE of 2.61%. Our findings contribute to the development of predictive modeling tools in urban transportation planning and management. This study provides a vital tool for improving the performance and sustainability of Phnom Penh's public transit systems by using the power of machine learning and considering a wide range of parameters, including those not covered by traditional datasets.*

Keywords: Bus arrival time prediction; Machine Learning; Predictive modeling; Public transportation

1. INTRODUCTION

The increasing effectiveness of Intelligent Transport System (ITS) solutions underscores the necessity for a real-time transportation information system, which, by providing commuters with timely information, enhances travel planning, reduces bus waiting times, and optimizes the use of public transport [1]. The precise estimation of link travel time holds paramount importance in ITS transit applications, particularly with the advancement of Advanced Travelers Information Systems (ATIS), which has significantly increased the importance of short-term travel time prediction, leading to the development of a variety of prediction models such as historical data-based models, regression models, time series models, and neural network models by diverse transit agencies over the years [2]. This research addresses the critical need for accurate bus arrival time forecasting by employing comprehensive predictive modeling methods based on machine learning techniques.

Advanced technologies now enable transit agencies to acquire real-time bus information, reducing passenger journey times and improving service levels, leading to a growing interest in using electronic information and communication technologies to provide passengers with real-time arrival information for more efficient and informed travel planning [3]. The study aims to provide transit agencies with valuable data to optimize bus schedules and routes, ultimately improving service quality and passenger satisfaction in Phnom Penh, Cambodia. Phnom Penh City Bus, a public transportation network serving urban areas,

[*] Corresponding author: Phannet Pov

E-mail: phannetpov@gmail.com; Tel: +855-71 828 2736

grew quickly in the following years, covering 21 routes by 2024 [4]. Furthermore, this research seeks to enhance the efficiency and reliability of public transportation such as Phnom Penh City Bus.

Predictive modeling for bus arrival time prediction is a growing field of study in transportation engineering and data science. This field involves using a variety of machine learning methods, each with unique characteristics, to capture the complex dynamics of urban traffic and transit networks. A diverse range of methodologies, such as Time Series Regression (TSR), Artificial Neural Network (ANN), Kalman Filter (KF), Support Vector Machine (SVM), and others [5] were utilized to construct a suite of regression models designed to predict the arrival time of buses traveling between two specified points along a route [5,6]. The regression model assumption of independence among different factors is often unrealistic [7]. A study by Yin et al. 2017 considered Kalman Filter model for predicting travel time [8], but we did not consider it in this study due to its limited ability to handle non-linear relationships and its reliance on assumptions that may not hold in complex. The Support Vector Machine method offers clear advantages in addressing challenges such as small sample sizes, nonlinearity, and multivariable classification and regression problems [9], yet its performance is profoundly influenced by the parameters governing the training process [10,11]. The k-NN method was formulated, with findings indicating its capacity to enhance the accuracy of bus arrival time prediction [6]. The prediction outcomes are contrasted with those of the gradient boosting model, revealing that the XGBoost model demonstrates superior performance in terms of both accuracy and efficiency [12]. The reliability analysis demonstrated that enhanced ANNs exhibit accurate performance for predicting both single and multiple stops, with the stop-based ANN being preferred in scenarios involving multiple intersections between stops, while the link-based ANN is better suited for pairs of stops with fewer intersections [13]. The ANN model utilized arrival time, dwell time, schedule adherence, and distance as primary predictors [14], which tends to result in high prediction error rates during adverse traffic conditions due to the model's lack of adaptation to changing traffic conditions [2]. The ANN achieved better prediction accuracy than both the historical data-based model and the regression model, which presented a freeway travel time prediction framework that integrated a state-space neural network with preprocessing strategies employing imputation [3,15].

2. METHODOLOGY

2.1 Raw Dataset

The raw dataset contains historical bus arrivals for route 11A from Sangkat Pong Tuek to Sangkat Tual Svay Prey II. It stretches around 16.4 kilometers and serves 19 bus stations. It provides essential information on the route's operational features. GPS tracking was used on city buses to collect data for two months, from July to August 2023. Table 1 describes the raw dataset, which combines data from GPS and weather sources. The original dataset contains many attributes, but significant attributes have been selected for this research. The key attributes used are latitude, longitude, date, time, and weather conditions.

Table 1. Raw dataset

Latitude	Longitude	Date	Time	Weather
11.46428	104.81640	2023-07-20	17:04:16	Cloudy
11.46484	104.82040	2023-07-20	17:05:22	Cloudy
11.46437	104.81637	2023-07-21	09:53:17	Light rain
11.48765	104.85140	2023-07-21	09:59:05	Light rain
11.46430	104.81648	2023-08-01	11:03:06	Cloudy
11.55199	104.90270	2023-08-01	11:22:32	Cloudy

2.2 Data Preprocessing

The data preprocessing involves amalgamating historical GPS data with meteorological information. To enrich the dataset and offer deeper insights into travel dynamics, new variables are derived to expand the information on trip characteristics. The Haversine formula is used to calculate distances between successive Global Positioning System (GPS) coordinates [16], calculated using Eq. 1. However, bus route maps are not always straightforward due to the complexity of corners and curves, so to ensure accurate distance estimation, several additional points are included throughout the route by interpolating longitude and latitude values.

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2-\phi_1}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \text{(Eq.1)}$$

where:

d : the distance between two points

ϕ : the longitude

λ : the latitude

R : the radius of the earth

The average speed between stops is determined, providing useful information about travel velocity along various segments of the route. Finally, weather data is fully concatenated with the information using timestamps, adding contextual depth with variables like cloudy and rainy conditions. By methodically following these preprocessing steps, a comprehensive dataset is created, setting the groundwork for the construction of accurate predictive models customized to bus arrival times.

Table 2 presents a processed dataset focusing on bus arrival times, consolidating various features derived from Table 1. The dataset includes key attributes following preprocessing, such as current bus stop identifiers, next bus stop identifiers, geographical coordinates (current latitude, current longitude, next latitude, next longitude), distance between stops, average speed, trip identifiers, hour of day, day of the week, weather conditions, and trip durations. This consolidation ensures comprehensive data readiness for subsequent analyses related to bus route optimization and operational efficiency. The dataset is separated into features and a target variable, which is the arrival time. A method is employed to randomly partition the dataset, allocating 80% of the arrival time dataset for training and 20% for testing.

Table 2. Processed dataset of bus arrival time

Current Bus Stop	Next Bus Stop	Current Latitude	Current Longitude	Next Latitude	Next Longitude	Distance	Speed	Trip	Hour of Day	Day of Week	Weather Condition	Trip duration
11A_1	11A_2	11.46453	104.81657	11.46483	104.82052	0.4961	0.744	1	6	15	Cloudy	40
11A_1	11A_3	11.46453	104.81657	11.46325	104.82602	1.1667	0.707	1	6	15	Cloudy	99
11A_1	11A_4	11.46453	104.81657	11.46250	104.83288	2.0589	0.730	1	6	15	Cloudy	169
11A_2	11A_3	11.46483	104.82052	11.46325	104.82602	0.6706	0.682	1	6	15	Cloudy	93
11A_2	11A_4	11.46483	104.82052	11.46250	104.83288	1.5628	0.726	1	6	15	Cloudy	129
11A_2	11A_5	11.46483	104.82052	11.46986	104.84461	3.1717	0.734	1	6	15	Cloudy	259
11A_3	11A_4	11.46325	104.82602	11.462506	104.83288	11.549	0.772	1	6	15	Cloudy	65

2.3 Prediction Models

The study investigates the application of various machine learning techniques such as linear regression, XGBoost, Support Vector Machine, K-Nearest Neighbors, and Artificial Neural Network to develop predictive models. The study focuses on analyzing historical data to train and test these models. Each algorithm's performance in predicting bus arrival times is evaluated, providing insights into their effectiveness in this specific context. This research contributes to the advancement of predictive modeling in transportation systems, particularly in urban environments such as Phnom Penh, by exploring the efficacy of different machine learning approaches.

2.3.1 Linear Regression

Linear regression (LR), a fundamental technique in both statistical analysis and machine learning, serves to estimate the dependent variable using a linear combination of independent factors, essentially optimizing a line to minimize prediction disparities [17]. The equation of linear regression is expressed as:

$$y = \varepsilon + \beta_0 + \sum_{i=1}^{n} \beta_i x_i \qquad \text{(Eq. 2)}$$

The linear regression equation encompasses the predicted arrival time y, an error term ε, an intercept β_0, and a coefficients β_i for each independent variable x_i. The intercept β_0 represents the expected arrival time when all independent variables are zero, while the coefficients β_i indicate the impact of each independent variable on the arrival time. By estimating the coefficients that minimize the error term, linear regression provides a model that can be used to forecast bus arrival times based on the given independent variables.

2.3.2 XGBoost

Gradient boosting, a machine learning technique, constructs predictive models by aggregating weak prediction models, typically decision trees, to enhance performance across regression and classification tasks [18,19]. The main objective function is defined as follows [20]:

$$\mathcal{L} = \sum_{i=1}^{n} L\big(y_i, F(x_i)\big) + \sum_{k=1}^{t} \Omega(f_k) + C \qquad \text{(Eq. 3)}$$

where $\Omega(f_k)$ denotes the regularization term, with C being a constant that can be optionally removed.

The regularization term $\Omega(f_k)$ is

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \qquad \text{(Eq. 4)}$$

In this context, γ represents the penalty associated with the complexity of the tree's leaves, while T denotes the total number of leaf nodes. The term λ is the regularization parameter that adds a penalty to prevent overfitting, and w_j signifies the output values assigned to each leaf node. Leaf nodes, which are terminal nodes in the tree, correspond to the final predicted categories based on the classification criteria and cannot be further split.

In XGBoost, the loss function is enhanced by incorporating the second-order Taylor expansion, which includes both first-order and second-order derivatives to improve optimization accuracy [20]. When using the mean squared error (MSE) as the loss function [20], the primary function is expressed as:

$$\mathcal{L} = \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2}\left(h_i w_{q(x_i)}^2 \right) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \qquad \text{(Eq. 5)}$$

where $q(x_i)$ is a function that maps data points to leaves, g_i and h_i represents loss function's first and second derivatives, respectively.

The overall loss value in a decision tree model is determined by summing the contributions from all individual leaf nodes [20]. Since each sample in the decision tree corresponds to a leaf node, the final loss is derived from aggregating the loss values across these leaf nodes. Consequently, the loss function for the model can be expressed as follows:

$$\mathcal{L} = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2\right] + \gamma T \qquad \text{(Eq. 6)}$$

where $G_j = \sum_{i \in I_j} g_i$ $H_j = \sum_{i \in I_j} h_i$, and I_j are the total number of samples in leaf node j.

2.3.3 Support Vector Machine

Support Vector Machine, in short SVM, a learning theory grounded in statistics and employing the principle of structural risk minimization, exhibits superior generalization prowess through the utilization of high-dimensional linear functions, facilitating the capture of intricate data patterns with greater ease compared to alternative models [10,21,22].

$$y = f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x) + b \qquad \text{(Eq. 7)}$$

Eq. 7 presents the function $f(x)$, which is defined as a summation over n terms, each representing the weighted similarity between an input data point x and a set of data points x_i. These similarities are computed using a kernel function denoted by K, with the linear kernel being specified in Eq. 8. In this equation, $f(x)$ is expressed as the sum of the products of the differences between corresponding coefficients α_i and α_i^*, and linear kernel $K(x_i, x)$, along with an additional offset term b.

$$K(x_i, x) = x_i \cdot x \qquad \text{(Eq. 8)}$$

This mathematical representation aids in understanding the function approximation process in support vector machine, particularly in discerning between different classes of data points in classification tasks.

2.3.4 K-Nearest Neighbors

The prediction method using K-Nearest Neighbors (KNN) involves selecting past sequences from the time series that closely resemble the current sequence and integrating their future values to forecast the next value in the current sequence [6,23].

$$d(x, x_i) = \sum_{j=1}^{n}\left|x_j - x_{ij}\right| \qquad \text{(Eq. 9)}$$

As shown in Eq. 9, $d(x, x_i)$ is a distance metric measuring the dissimilarity between an input vector x and a reference vector x_j across n dimensions. The metric computes the sum of absolute differences between corresponding components of x and x_j, crucial for determining proximity in feature space.

$$y = \frac{1}{k}\sum_{i=1}^{k} y_i \qquad \text{(Eq. 10)}$$

Additionally, Eq. 10 outlines how KNN predicts y, the output variable of interest. By averaging the target values y_i from the k nearest neighbors of x, the algorithm determines y. This averaging process reflects KNN's principle of predicting based on the average value of nearby data points, emphasizing its simplicity and effectiveness in handling non-linear decision boundaries and diverse data distributions.

2.3.5 Artificial Neural Network

The Artificial Neural Network (ANN) consists of an input layer, hidden layers, and an output layer with a single neuron, as illustrated in Fig. 1. This research explores variations in the number of hidden layers to assess how the depth of the network influences its ability to capture complex patterns and relationships in the data. By increasing the number of hidden layers, the model gains additional capacity to represent intricate functions, which may enhance its predictive accuracy. However, this also introduces the risk of overfitting, making it essential to carefully balance model complexity with generalization. The study aims to determine the optimal number of hidden layers that offers the best trade-off between model performance and computational efficiency.

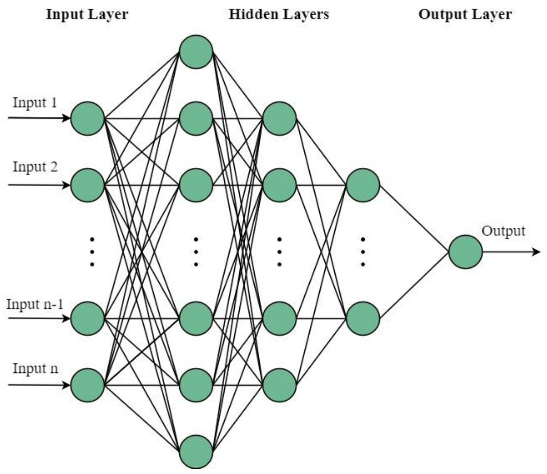

Fig. 1. Artificial Neural Network architecture

In the proposed ANN models for predicting bus arrival times in this study, the input features are normalized, consisting of twelve variables (i.e., i=12), while the target output is a single variable representing the bus arrival time. The models differ in the number of hidden layers, with configurations of one, two, and three hidden layers (i.e., h=1, 2, 3). The sigmoid function is used as the activation function within the hidden layers.

$$y = f_{sig}\left\{b_0 + \sum_{k=1}^{h}\left[w_k \times f_{sig}\left(b_{hk} + \sum_{i=1}^{m} w_{ik}X_i\right)\right]\right\} \qquad \text{(Eq. 11)}$$

Eq. 11 presents a foundational formulation of a neural network model, where y signifies the network's output, shaped by the application of a sigmoid activation function $f_{sig}(z)$ to a weighted sum. This sum incorporates biases b_0 and b_{hk} along with inputs X_i weighted by coefficients w_{ik} and w_k.

$$f_{sig}(z) = \frac{1}{1 + e^{-z}} \qquad \text{(Eq. 12)}$$

The sigmoid function $f_{sig}(z)$ as defined in Eq. 12 operates to confine outputs within the (0, 1) interval, facilitating the modeling of non-linear relationships and the capture of intricate data patterns.

2.4 Hyperparameter Tuning

The hyperparameters tuning is critical in improving the performance of machine learning models. Grid search cross-validation (GridSearchCV) was utilized to select the optimal model for each machine learning approach, and the parameters yielding the best cross-validation performance were then used to automatically fit a new model to the entire training dataset [24]. GridSearchCV was used to identify the best hyperparameters value for this study. Table 3 shows the optimal hyperparameters of proposed machine learning models such as XGBoost, KNN, SVM, and ANN were optimized using grid search within a defined search space. Each algorithm's hyperparameters were systematically tuned to achieve optimal predictive performance and generalization. This process highlights the importance of thorough hyperparameter tuning in enhancing the effectiveness of machine learning models.

Table 3. The optimal hyperparameters of proposed machine learning models

Machine Learning	Hyperparameters	Search Space	Optimal Hyperparameters Value
XGBoost	learning_rate	[0.01, 0.05, 0.1]	0.05
	max_depth	[3, 4, 5]	5
	num_estimators	[100, 200, 300]	300
	gamma	[0, 0.1, 0.5]	0.1
	colsample_bytree	[0.8, 1.0]	1.0
Support Vector Machine (SVM)	C	[0.1, 1, 10, 100]	100
	gamma	[0.01, 0.1, 1]	0.1
	epsilon	[0.01, 0.1, 0.2, 0.5]	0.5
K-Nearest Neighbors (KNN)	n_neighbors	[3, 5, 7, 9]	7
	weights	['Uniform', 'distance']	Uniform
	algorithm	['auto', 'ball_tree', 'kd_tree', 'brute']	Auto
	p	[1, 2]	1
Artificial Neural Network (ANN)	learning_rate	[0.01, 0.001, 0.001]	0.001
	batch_size	[4, 8, 16, 32]	4
	epochs	[50, 100, 200]	200

2.4 Evaluation metrics

Evaluation metrics are vital tools for assessing the effectiveness of predictive models. Three commonly used metrics for this purpose are Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|T_{actual,i} - T_{predict,i}\right| \qquad \text{(Eq. 13)}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(T_{actual,i} - T_{predict,i}\right)^2} \qquad \text{(Eq. 14)}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|T_{actual,i} - T_{predict,i}\right|}{T_{actual,i}} \cdot 100\% \qquad \text{(Eq. 15)}$$

T_{actual} represents the actual value of arrival time, and $T_{predict}$ signifies the predicted arrival time, generated by the predictive model. MAE computed using Eq. 13 quantifies the average magnitude of the differences between the actual and predicted arrival times, providing a straightforward measure of prediction accuracy. RMSE is expressed in Eq. 14 and is particularly useful for capturing the overall variability in prediction errors. MAPE offers insights into the relative accuracy of predictions, considering the scale of actual arrival times, computed using Eq. 15.

3. RESULTS AND DISCUSSION

3.1 Optimal Hyperparameters Estimation

Table 3 shows optimal hyperparameters which were determined by machine learning models. XGBoost were found the best of parameters such as a learning rate of 0.05, a maximum depth of 5, 300 estimators, a gamma value of 0.1, and a column sample by tree of 1.0. SVM performed optimally with a regularization parameter C of 100, a gamma value of 0.1, and an epsilon of 0.5, enhancing classification accuracy and margin control. KNN showed best results with 7 neighbors, uniform weights, 'auto' algorithm, and a power parameter P of 1 for the Minkowski metric, ensuring effective local pattern recognition. ANN achieved optimal performance with a learning rate of 0.001, a batch size of 4, and training over 200 epochs, allowing for gradual learning and precise convergence on complex data patterns.

3.2 Selected Number of Hidden Layers of ANN

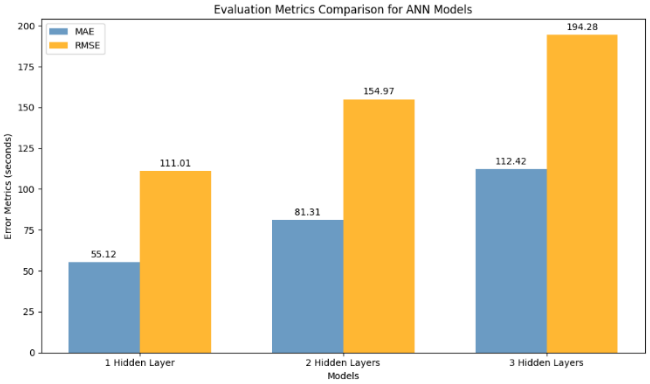

Fig. 2. Evaluation Metrics Comparison for ANN Models

The bar chart in Fig. 2 depicts a comparative examination of ANN models with one, two, and three hidden layers, evaluated using metrics such as MAE and RMSE in seconds. The findings show that the ANN configuration with one hidden layer delivers the highest predicted accuracy, with an MAE of 55.12 and RMSE of 111.01. On the other hand, models with 2 and 3 hidden layers have increasing MAE values 81.31 and 112.42, respectively and RMSE values 154.97 and 194.28, respectively, suggesting a diminishing return in accuracy as network complexity increases. The model with one hidden layer has been selected for comparison with other machine learning models.

3.3 Result of Predictive Models

In the experiment, machine learning models were used to predict bus arrival times, with 20% of the dataset reserved for testing. The comparison is made between the predicted arrival times and the actual values to assess prediction accuracy. The results highlight the errors in the machine learning predictions relative to the actual bus arrival times. The model with the smallest error, measured in seconds, is considered the best due to the importance of precise timing in bus arrival predictions. Moreover, we employed k-fold cross-validation to ensure the robustness of our models, which allowed us to validate the models across different subsets of the data. This approach assists in mitigating overfitting and provides a more reliable estimate of model performance.

Table 4 displays the refined performance metrics of machine learning models for bus arrival time prediction following parameter tuning from Table 3. XGBoost demonstrates significant improvement with an MAE of 16 seconds and an RMSE of 28.03 seconds. Although LR, SVM and KNN exhibit reduced errors compared to their initial configurations, they still lag behind XGBoost's performance. Remarkably, Artificial Neural Network exhibits substantial enhancement post-tuning, achieving an impressive MAE of 55 seconds, and an RMSE of 111.01 seconds.

Table 4. Results from the test set with hyperparameters tuning

Model	MAE (seconds)	RMSE (seconds)
Linear Regression (LR)	144.17	236.17
XGBoost	**16**	**28.03**
Support Vector Machine (SVM)	120.19	242.72
K-Nearest Neighbors (KNN)	109.28	166.93
Artificial Neural Network (ANN)	**55.12**	**111.01**

Fig. 3-7 show the prediction versus actual graphs for Linear regression, XGBoost, SVM, KNN, and ANN, respectively. In these figures, the x-axis shows the actual values, and the y-axis shows the expected values. Of these models, XGBoost has the best alignment between the expected and actual data, which proves more accurate. The figures show that the ANN estimates an arrival time only 2,500 seconds, while the estimates of other models typically reach 3,000 seconds. XGBoost can anticipate larger values, perhaps up to 3500 seconds of prediction. While the models demonstrated varying levels of success in predicting bus arrival times, it is important to acknowledge potential biases in our dataset, such as the disproportionate representation of data from specific hour of the day and weather conditions.

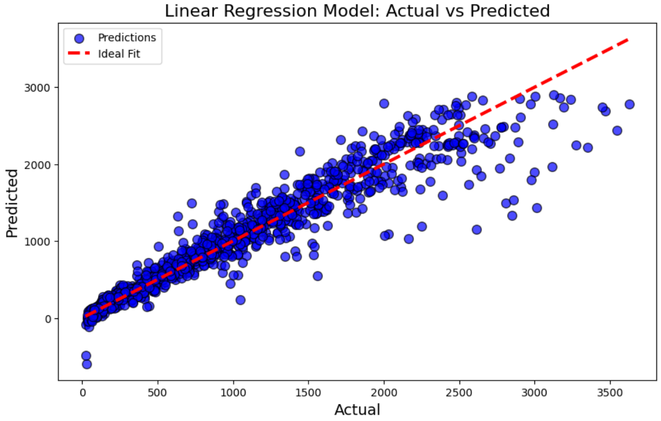

Fig 3. Predicted versus actual plot for Linear Regression

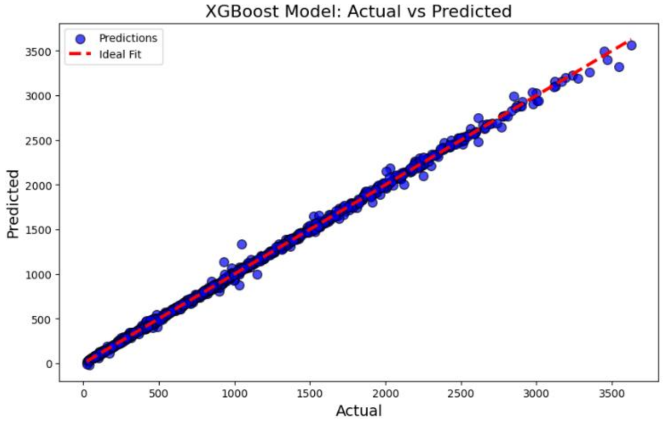

Fig 4. Predicted versus actual plot for XGBoost

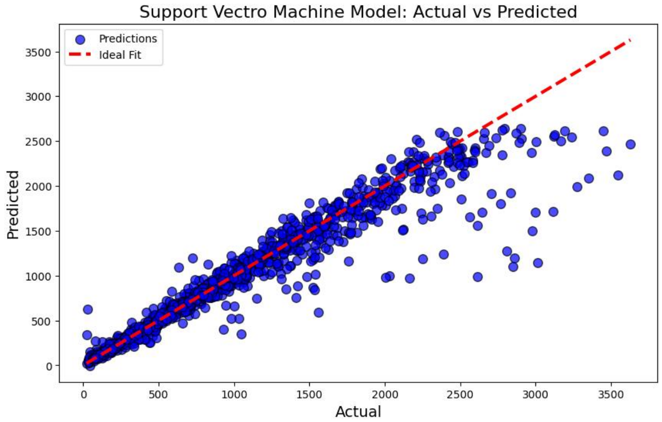

Fig 5. Predicted versus actual plot for SVM

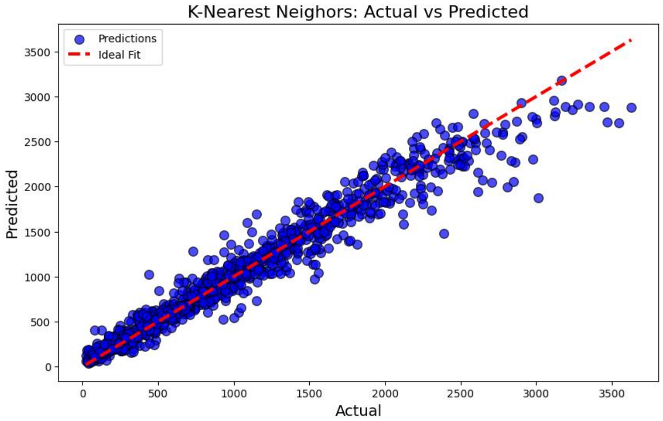

Fig 6. Predicted versus actual plot for KNN

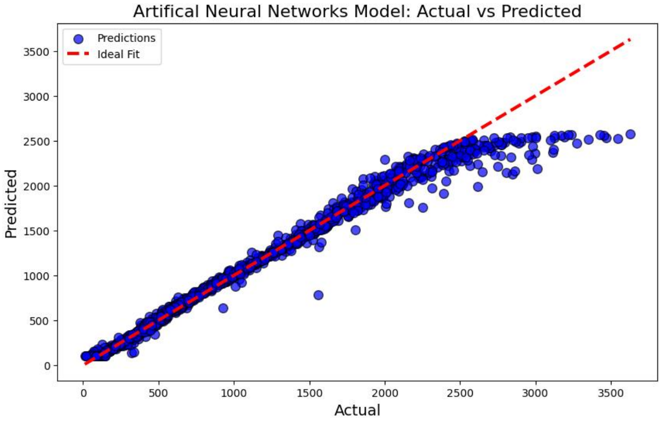

Fig 7. Predicted versus actual plot for ANN

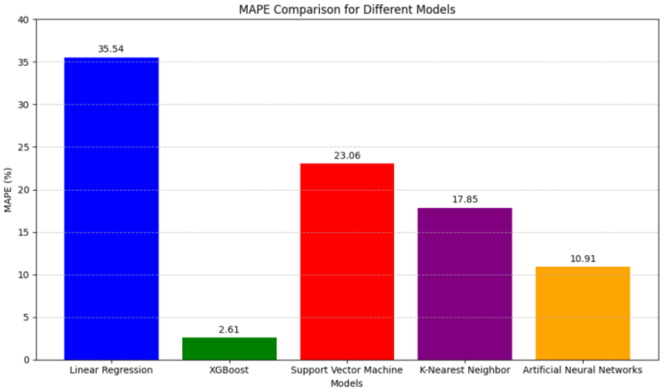

Fig 8. Model performance comparison results of MAPE

Fig. 8 shows a bar chart comparing results of MAPE among different models. Among the models analyzed, XGBoost demonstrated the lowest MAPE at 2.61%, indicating the highest accuracy in predicting bus arrival times. In contrast, LR, SVM, and KNN models have higher MAPE values of 35.54%, 23.06%, and 17.85%, indicating a poorer predictive precision in this context. These findings demonstrate that advanced machine learning algorithms, such as XGBoost and ANN, outperform classic regression and proximity-based methods in terms of predicting accuracy.

4. CONCLUSIONS

The study's examination of machine learning models for predicting bus arrival times in Phnom Penh, leveraging historical bus data and weather conditions, underscores the superiority of Artificial Neural Network and XGBoost algorithms over traditional methods. Notably, XGBoost emerges as the standout performer, showcasing exceptional predictive capabilities with minimal errors, including an MAE of 16 seconds, an RMSE of 28.03 seconds, and a MAPE of 2.61%. This highlights XGBoost's adeptness in capturing the nuances of bus arrival time dynamics, laying a foundation for transit management optimization and urban transportation system enhancements. The remarkable accuracy achieved by XGBoost can be attributed to its ability to discern intricate patterns and temporal dependencies within the data, surpassing the capabilities of linear regression and distance-based algorithms. Furthermore, the integration of weather conditions into the predictive modeling process enhances forecast precision by accounting for environmental factors that influence transit operations. As a result, XGBoost presents a robust solution for optimizing bus schedules, minimizing delays, and improving overall commuter satisfaction. By harnessing the power of data-driven insights, cities can pave the way for smarter, more efficient public transportation networks that cater to the evolving needs of urban commuters, ultimately fostering sustainable and accessible mobility solutions for Phnom Penh and beyond.

Future research will focus on developing specialized Internet of Things (IoT) devices equipped with sensors to capture real-time data directly from buses. These sensors would monitor essential factors, providing a continuous stream of data to the predictive models using machine learning. The integration of this sensor-driven IoT system with real-time data analytics will significantly improve the precision and reliability of bus arrival predictions. The system can deliver accurate information by processing the real-time dataset and up-to-the-minute arrival information to passengers via digital displays and mobile applications, ultimately enhancing the efficiency of public transportation systems and commuter satisfaction.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to Phnom Penh Capital Hall for approving the collection of essential data that has significantly contributed to my research. The cooperation and support from Phnom Penh City Bus in sharing this dataset have been invaluable. I am profoundly grateful for their assistance and collaboration.

REFERENCES

[1] S. Sharad, P. B. Sivakumar, and V. A. Narayanan, "The smart bus for a smart city — A real-time implementation," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Bangalore, India: IEEE, Nov. 2016, pp. 1–6. doi: 10.1109/ANTS.2016.7947850.

[2] R. Jeong and R. Rilett, "Bus arrival time prediction using artificial neural network model," in *Proceedings. The 7th international IEEE conference on intelligent transportation systems (IEEE Cat. No. 04TH8749)*, IEEE, 2004, pp. 988–993. Accessed: Aug. 28, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1399041/

[3] B. Yu, W. H. K. Lam, and M. L. Tam, "Bus arrival time prediction at bus stop with multiple routes," *Transp. Res. Part C Emerg. Technol.*, vol. 19, no. 6, pp. 1157–1170, Dec. 2011, doi: 10.1016/j.trc.2011.01.003.

[4] "Phnom Penh City Bus," *Wikipedia*. Mar. 29, 2024. Accessed: Apr. 17, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Phnom_Penh_City_Bus&oldid=1216137545

[5] J. Patnaik, S. Chien, and A. Bladikas, "Estimation of bus arrival times using APC data," *J. Public Transp.*, vol. 7, no. 1, pp. 1–20, 2004.

[6] T. Liu, J. Ma, W. Guan, Y. Song, and H. Niu, "Bus arrival time prediction based on the k-nearest neighbor method," in *2012 Fifth International Joint Conference on Computational Sciences and Optimization*, IEEE, 2012, pp. 480–483. Accessed: Apr. 22, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6274771/

[7] L. Vanajakshi, S. C. Subramanian, and R. Sivanandan, "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses," *IET Intell. Transp. Syst.*, vol. 3, no. 1, pp. 1–9, 2009.

[8] T. Yin, G. Zhong, J. Zhang, S. He, and B. Ran, "A prediction model of bus arrival time at stops with multi-routes," *Transp. Res. Procedia*, vol. 25, pp. 4623–4636, 2017, doi: 10.1016/j.trpro.2017.05.381.

[9] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, 1999.

[10] M. Yang, C. Chen, L. Wang, X. Yan, and L. Zhou, "Bus arrival time prediction using support vector machine with genetic algorithm," *Neural Netw. World*, vol. 26, no. 3, p. 205, 2016.

[11] B. Yao, P. Hu, M. Zhang, and M. Jin, "A support vector machine with the tabu search algorithm for freeway incident detection," *Int. J. Appl. Math. Comput. Sci.*, vol. 24, no. 2, pp. 397–404, Jun. 2014, doi: 10.2478/amcs-2014-0030.

[12] Z. Chen and W. Fan, “A Freeway Travel Time Prediction Method Based on an XGBoost Model,” *Sustainability*, vol. 13, no. 15, Art. no. 15, Jan. 2021, doi: 10.3390/su13158577.

[13] S. I.-J. Chien, Y. Ding, and C. Wei, “Dynamic Bus Arrival Time Prediction with Artificial Neural Networks,” *J. Transp. Eng.*, vol. 128, no. 5, pp. 429–438, Sep. 2002, doi: 10.1061/(ASCE)0733-947X(2002)128:5(429).

[14] W. Treethidtaphat, W. Pattara-Atikom, and S. Khaimook, “Bus arrival time prediction at any distance of bus route using deep neural network model,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama: IEEE, Oct. 2017, pp. 988–992. doi: 10.1109/ITSC.2017.8317891.

[15] J. W. C. Van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, “Accurate freeway travel time prediction with state-space neural networks under missing data,” *Transp. Res. Part C Emerg. Technol.*, vol. 13, no. 5–6, pp. 347–369, 2005.

[16] N. R. Chopde and M. Nichat, “Landmark based shortest path detection by using A* and Haversine formula,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 1, no. 2, pp. 298–302, 2013.

[17] B. P. Ashwini, R. Sumathi, and H. S. Sudhira, “Bus Travel Time Prediction: A Comparative Study of Linear and Non-Linear Machine Learning Models,” *J. Phys. Conf. Ser.*, vol. 2161, no. 1, p. 012053, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012053.

[18] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[19] N. Gaikwad and S. Varma, “Performance analysis of bus arrival time prediction using machine learning based ensemble technique,” in *Proceedings 2019: Conference on Technologies for Future Cities (CTFC)*, 2019. Accessed: Apr. 22, 2024. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3358828

[20] M. Amjad, I. Ahmad, M. Ahmad, P. Wróblewski, P. Kamiński, and U. Amjad, “Prediction of pile bearing capacity using XGBoost algorithm: modeling and performance evaluation,” *Appl. Sci.*, vol. 12, no. 4, p. 2126, 2022.

[21] B. Yao, J. Yao, M. Zhang, and L. Yu, “Improved support vector machine regression in multi-step-ahead prediction for rock displacement surrounding a tunnel,” *Sci. Iran. Trans. Civ. Eng.*, vol. 21, no. 4, p. 1309, 2014.

[22] B.-Z. Yao, C.-Y. Yang, J.-B. Yao, and J. Sun, “Tunnel Surrounding Rock Displacement Prediction Using Support Vector Machine,” *Int. J. Comput. Intell. Syst.*, vol. 3, no. 6, pp. 843–852, Dec. 2010, doi: 10.1080/18756891.2010.9727746.

[23] J. Arroyo and C. Maté, “Forecasting histogram time series with k-nearest neighbours methods,” *Int. J. Forecast.*, vol. 25, no. 1, pp. 192–207, 2009.

[24] Z. M. Alhakeem, Y. M. Jebur, S. N. Henedy, H. Imran, L. F. A. Bernardo, and H. M. Hussein, “Prediction of Ecofriendly Concrete Compressive Strength Using Gradient Boosting Regression Tree Combined with GridSearchCV Hyperparameter-Optimization Techniques,” *Materials*, vol. 15, no. 21, p. 7432, Oct. 2022, doi: 10.3390/ma15217432.