Emerging Trends in Digital Transformation

ISSN: 3107-4308 | Conference Issue 2025

International Conference on Emerging Trends in Engineering, Technology & Management (ICETM – 2025)

ETDT

ETDT

Emerging Trends in Digital Transformation

Special Issue - 2025 ISSN: **3107-4308** *Paper ID:* ***ETDT-SI-08***

International Conference on Emerging Trends in Engineering, Technology & Management (ICETM-2025)
Conducted by *Viswam Engineering College (UGC—Autonomous Institution)* held on 11th & 12th, April- 2025

UTILIZING MACHINE LEARNING ALGORITHMS FOR TRAFFIC FLOW DETECTION AND PREDICTION IN SMART TRAFFIC LIGHT SYSTEMS

G S Gowthami Kumari, Assistant professor, I .Deepika, Assistant professor, K. Vijaya Lakshmi, Assistant professor,
Department of CSE, Viswam Engineering College, Madanapalle.

ABSTRACT: In present days, numerous urban areas are facing challenges related to traffic congestion during specific peak periods, resulting in increased pollution, noise, and stress for residents. Neural networks (NN) and machine-learning (ML) techniques are being increasingly utilized to address practical issues, surpassing traditional analytical and statistical methods due to their capability to handle dynamic behavior over time and a vast number of parameters in extensive datasets. This study introduces machine-learning (ML) and deep-learning (DL) algorithms for the prediction of traffic flow at an intersection, paving the way for adaptive traffic management through either remote control of traffic signals or the implementation of an algorithm that adjusts timing based on anticipated flow. Consequently, the focus of this research is solely on traffic flow prediction. Two public datasets are employed to train, validate, and assess the proposed ML and DL models. The initial dataset comprises vehicle counts recorded every five minutes at six intersections over a span of 56 days using various sensors. In this paper, four out of the six intersections were utilized for training the ML and DL models. The Multilayer Perceptron Neural Network (MLP-NN) demonstrated superior performance with an R-Squared and EV score of 0.93, requiring less training time. Gradient Boosting followed closely behind, with Recurrent Neural Networks (RNNs) showing good metric results but longer training times. Lastly, Random Forest, Linear Regression, and Stochastic Gradient exhibited their own performance metrics. All ML and DL algorithms displayed promising results, suggesting their suitability for integration into smart traffic light controllers. The intelligent transportation system (ITS) plays a crucial role in the development of smart cities. One of the key components of ITS is traffic flow prediction, which helps in optimizing traffic management. Traffic lights are an integral part of this system, and their efficient control is essential for smooth traffic flow. Machine learning techniques, such as deep learning and recurrent neural networks, are employed to analyze traffic patterns and make accurate predictions. Artificial intelligence and regression algorithms are also utilized to enhance the performance of ITS. Overall, the integration of these technologies and algorithms in ITS contributes to the advancement of smart cities and efficient traffic management.

DOI : https://doi.org/10.5281/zenodo.17276775

Keywords: Intelligent Transportation System (ITS), machine-learning (ML), Sequential Minimal Optimization (SMO) Regression, Multilayer Perceptron (MLP), M5P model tree, and Random Forest (RF), Gaussian Process Regressor (GPR), Multilayer Perceptron Neural Network (MLP-NN), Autoregressive Integrated Moving Average (ARIMA).

1. INTRODUCTION

In today's world, the rapid increase in population and the subsequent rise in the number of vehicles in urban areas, combined with the limitations of traditional traffic control signs, have turned vehicular traffic into a major issue of contemporary society. This problem has adverse effects on the environment, public health, and the economy. The concept of Intelligent Transportation System (ITS) plays a crucial role in addressing this challenge, serving as a key element of smart city infrastructure. By leveraging big data and communication technology, ITS enables real-time analysis of road infrastructure and more effective traffic management. Central to this system are traffic predictions, which help anticipate future traffic conditions on a transportation network based on past data. Such information is valuable for ITS applications like traffic congestion management and traffic light control. For instance, it can assess the probability of congestion on a specific road segment and take proactive measures to address it.

There are two main types of techniques used for traffic prediction: parametric and non-parametric. Parametric techniques include stochastic and temporal methods, while non-parametric techniques involve machine-learning (ML) models. Recent studies have shown that non-parametric algorithms outperform parametric algorithms because they can handle a large number of parameters in massive data. In a study conducted in Porto city, five ML algorithms were evaluated for forecasting the total volume of traffic flow. These algorithms included Linear Regression, Sequential Minimal Optimization (SMO) Regression, Multilayer Perceptron (MLP), M5P model tree, and Random Forest (RF). The results of the experiment revealed that the M5P regression tree performed better than the other regression models. Another study explored multi-model ML methods for traffic flow estimation using floating car data. Specifically, the capacity of Gaussian Process Regressor (GPR) was evaluated in addressing this issue. Deep learning (DL), which is a subset of ML, utilizes multilayered neural networks and extensive data to train itself. The ability of DL models to extract knowledge from complex systems has made them a robust and viable solution in the field of Intelligent Transportation Systems (ITS).

A Multilayer Perceptron Neural Network (MLP-NN) has been introduced in references, utilizing a mutual information technique for predicting traffic flow. The simulations demonstrated a decrease in forecast error when compared to the mean and Autoregressive Integrated Moving Average (ARIMA) models that relied on past traffic data. The Back-Propagation Neural Network (BPNN) is a common architecture in Neural Networks, frequently employed in prediction and classification tasks. In reference, a traffic signal control system for urban areas is proposed, based on traffic flow prediction using BPNN. Additionally, reference applied BPNN to forecast future traffic volumes in a traffic light control system design, in conjunction with a genetic algorithm for timing optimization. This

DOI : https://doi.org/10.5281/zenodo.17276775

approach resulted in an almost 30 percent reduction in the average waiting rate compared to fixed-time traffic light control systems. By combining a genetic algorithm and neural network, reference introduced the Genetic Neural Network. Reference introduces a deep learning neural network method to optimize traffic flow and alleviate congestion at critical intersections by utilizing historical data from all movements at a specific intersection, with time series and environmental variables as input features. The output is processed through a delay equation to determine the optimal green times for managing traffic delays. In reference, a short-term traffic flow prediction method based on an enhanced wavelet neural network (WNN) is suggested. They incorporated an improved particle swarm optimization (IPSO) to prevent getting stuck in a local extremum. The outputs of the IPSO are the relevant wavelet neural network parameters, and experimental results indicate that this algorithm is more effective than WNN and PSO-WNN algorithms alone. The prediction outcomes are more consistent and precise, with an error reduction of nearly 15 percent compared to the traditional wavelet neural network.

Traffic control in large cities poses significant challenges. In order to address the financial burdens associated with traffic congestion, numerous countries worldwide have implemented Intelligent Transportation Systems (ITS). These systems rely on predictive models for traffic flow (TF) to aid in their development. ITS, which utilizes integrated communication and data processing technology, aims to enhance the transportation of people and goods by improving safety, reducing road congestion, and effectively managing traffic incidents. These objectives align with transportation policy goals such as demand management and prioritizing public transportation. TFP, or traffic flow prediction, finds extensive applications in city transportation and area management. It involves estimating future urban road traffic flow based on data collected from previous time periods at one or more observation points. This research endeavors to train a system that utilizes a TFP algorithm to forecast traffic. By analyzing user searches, the system can provide recommendations tailored to individual needs. Traffic congestion arises from the complex interplay of various factors, including road design, fluctuating traffic volume, weather conditions, accidents, and road maintenance activities. The implementation of this system will benefit the public by providing real-time TF and weather data, thereby minimizing the risk of urban road accidents and improving overall road safety.

2. ITS

The integration of communication, information, transportation, and urban transport systems is known as Intelligent Transportation Systems (ITS). The primary objectives of ITS are to improve traffic safety and efficiency. Some benefits of ITS include a) Minimized intersection congestion and delays, b) Speed control and optimization, c) Improved travel time, d) Capacity management, and e) Incident management. Academic and professional communities have shown increasing interest in ITS in recent times. Future ITS developments and the range of tasks falling under ITS are illustrated in Figures 1 and 2.

DOI : https://doi.org/10.5281/zenodo.17276775

Fig. 1: Future Intelligent Transportation system overview [10].

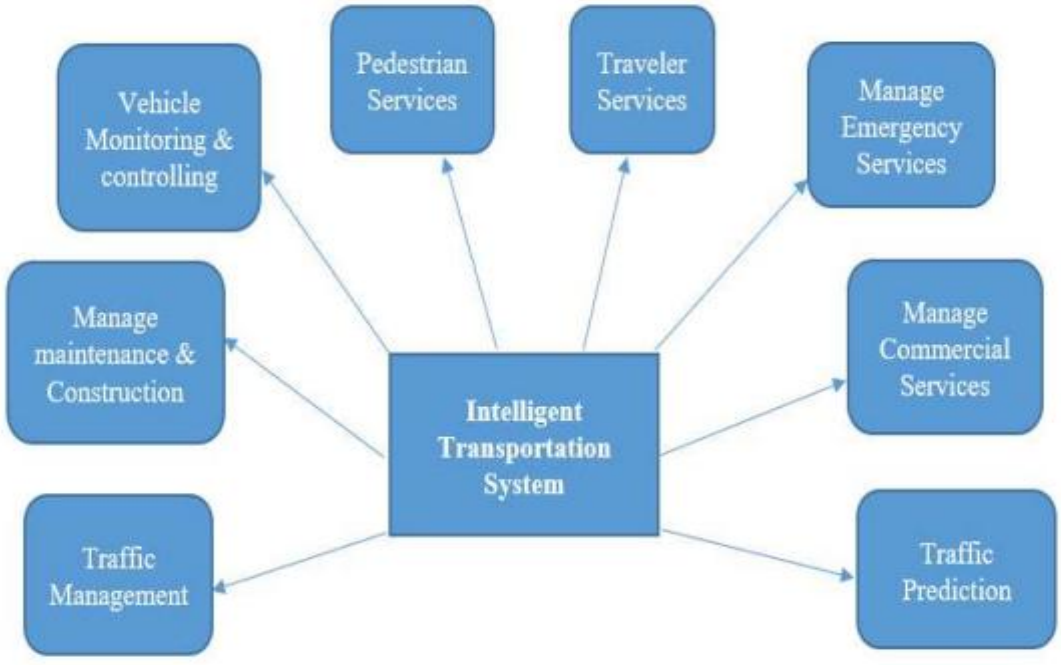


Fig. 2: Various tasks performed by an Intelligent Transportation Systems.

Machine Learning Predictions in ITS:

ML techniques have demonstrated remarkable proficiency in forecasting tasks within Intelligent Transportation Systems (ITS), particularly in predicting TF, travel time, vehicle behavior, user behavior, and road occupancy. Table 1 provides an overview of the diverse prediction categories encompassed by traffic forecasting.

DOI : https://doi.org/10.5281/zenodo.17276775

Table 2.1: Prediction Categories under Traffic Forecasting

Prediction Category	Description	Role of ML
Traffic Flow	Utilizing spatiotemporal dependencies for TFP analysis.	Traffic pattern analysis can incorporate various data sources such as weather data, time-series data, historical data, accident-prone area data, and road maintenance work information.
Travel Time	Forecasting the duration of travel for automobiles, buses, bicycles, and other modes of transportation.	Studying traffic flow using time-based information. Identifying characteristics and understanding trends in travel times.
The behavior of Vehicle & User	Predicting lane changes, vehicle steering angles, and pedestrian actions.	Analyzing driver behavior, identifying pedestrian trends, and predicting future vehicle movements.
Road Occupancy	Forecasting road density in urban areas to anticipate parking availability.	Investigating parking occupancy patterns and developing models for both short-term and long-term predictions.

In this document, five machine learning models (MLP-NN, Gradient Boosting Regressor, Random Forest Regressor, Linear Regressor, and Stochastic Gradient Regressor) and two deep learning models based on RNNs (GRU and LSTM) are assessed for their effectiveness in predicting traffic flow at an intersection. The aim is to enhance traffic light controllers without the need for a complete overhaul, thus making implementation more practical. Results show that all models perform well in predicting traffic flow and can be integrated into a smart traffic light controller. The remainder of the document is structured as follows: Section 2 outlines the materials and methods used to train the machine learning models. Section 3 presents the results of various metrics used to assess performance and compare ML and DL models. Section 4 details the proposed real-world application scenario. Conclusions and future work are discussed in Section 5.

3. MATERIALS AND METHODS

The research paper utilizes the Road Traffic Prediction Dataset provided by Huawei Munich Research Center. This dataset is a publicly available resource specifically designed for traffic prediction purposes. It encompasses data collected from various traffic sensors, such as induction loops. It is worth mentioning that there are currently only a limited number of public datasets available. The dataset enables the forecasting of traffic patterns and the adjustment of stop-light control parameters. It comprises recorded data from six intersections within an urban area, spanning duration of 56 days. The data is presented in the form of flow time series, representing the number of vehicles passing through every five minutes throughout the entire day. This dataset is particularly suitable for short-term predictions. In this study, four out of the six intersections are utilized to simulate four lanes of an intersection.

DOI : https://doi.org/10.5281/zenodo.17276775

Data Preprocessing

Missing values are often found in databases, typically represented by zeros, which can be attributed to sensor failures. In order to address this issue, the article referenced as proposes imputing the missing data points by using the historical average value. On the other hand, [20] suggests substituting these missing values with the mean of the entire column. Although these substituted values may not accurately reflect reality, researchers have determined that they are still preferable to having no data at all. However, it is important to note that this imputation procedure can lead to spikes in the real values during predictions. This can result in increased error, as there are instances where the trend indicates that the zero values were actually valid. These spikes are visually highlighted in red rectangles, as depicted in Figure 1a. To mitigate this issue, a moving average approach is employed, utilizing the 12 previous readings. This helps to smooth out the abrupt changes caused by the general average, as demonstrated in Figure 1b. Following the data preprocessing steps, the database is then divided into two portions: 75% of the data (equivalent to 42 days) is allocated for training, while the remaining 25% (equivalent to 14 days) is reserved for testing purposes.

RECURRENT NEURAL NETWORKS

RNNs Design

Missing values in databases are often represented by zeros, potentially caused by sensor failures. one study, the missing data points were filled using the historical average value, while another study replaced these values with the mean of the entire column. Although these substitutions may not reflect realistic values, researchers found them to be preferable to having no data at all. However, this method can lead to spikes in the actual values during predictions, resulting in increased errors. In some cases, zero values were indeed valid, causing these spikes. To address this issue, a moving average based on the 12 previous readings was applied. This approach helped to smooth out abrupt changes caused by the general average, as demonstrated in the database was divided into 75% training data (42 days) and 25% testing data (14 days).

RNNs Training

During the model compilation, the mean squared error (MSE) serves as the loss function, while the optimizer utilized is RMSprop from the Keras library with default parameters. The mean absolute error (MAE) is employed as the metric function. The training process involves a batch size of 128 and 50 epochs, with five percent of the training data allocated for validation. The experiments are conducted in Google Collaboratory and tracked using Weights & Biases . The training performance of the two architectures is depicted in Figure 3, showcasing the loss and evaluation metrics in both training and validation phases.

Machine Learning Methods

Five regression models from the scikit-learn library in Python have been utilized: Linear Regression, Gradient Boosting Regressor, Multilayer Perceptron Regressor, Stochastic Gradient Descendent Regressor, and Random Forest Regressor. All models were implemented with default parameters and a random state set to zero to ensure reproducibility. 'X' was reshaped for both training and testing to convert the 3D array into a 2D array required

DOI : https://doi.org/10.5281/zenodo.17276775

by these models, not RNN's. Subsequently, the models were trained using the training split. The methodology summary, depicted as a flowchart, can be seen in Figure.

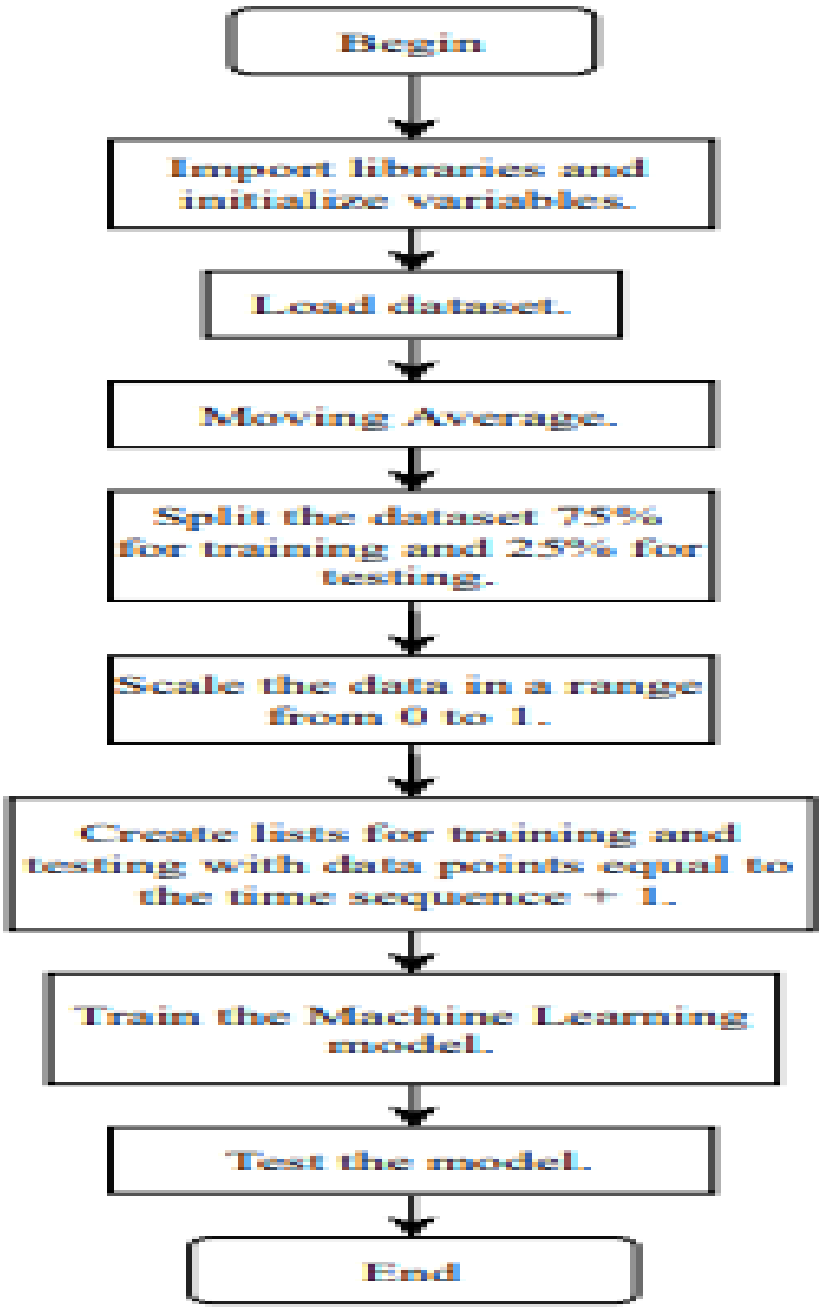


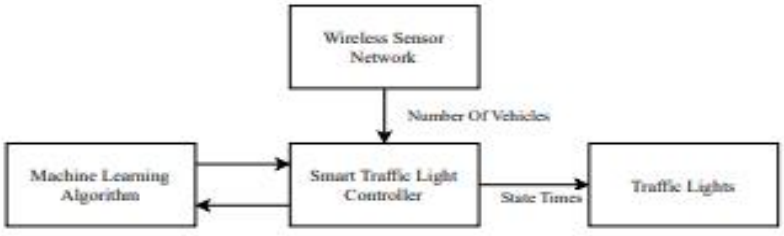
Fig 3: Flow chart of proposed method

4. PROPOSED USAGE SCENARIO

The models are suitable for integration into an intelligent traffic light controller, which is supplied with data from traffic sensors that monitor the number of vehicles passing through a lane at regular intervals. This data can be used to create a database similar to the one described in this study. Once the database is established, machine learning models can be trained for each intersection. These models can then predict traffic flow for the upcoming

DOI : https://doi.org/10.5281/zenodo.17276775

period based on historical data. After making predictions, the timing of each traffic light state can be optimized either manually by an operator or automatically using an algorithm. This entire process can be facilitated through wireless communication between the traffic lights and a central station or directly at the controller. A block diagram of the key components of the proposed system is illustrated in Figure 8a, while Figureb depicts its real-world application.



(a)

(b)

Fig 4 (a)Block diagram, 4 (b)Representation of real world scenario

5. RESULTS

In order to assess the effectiveness of ML and DL algorithms, we initially employed an inverse scaler on the 'y' test. Subsequently, we utilized the metrics provided by the scikit-learn library, including the mean absolute error (MAE), root mean square error (RMSE) , mean absolute percent error (MAPE) , R-squared (R 2) , and explained variance (EV).
They are defined as:

$$\mathrm{MAE}(y,\hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}|y_i - \hat{y}_i| \quad (1)$$

$$\mathrm{MAPE}(y,\hat{y}) = \frac{100\%}{n}\sum_{i=0}^{n-1}\frac{|y_i - \hat{y}_i|}{y_i} \quad (2)$$

$$\mathrm{RMSE}(y,\hat{y}) = \left[\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2\right]^{\frac{1}{2}} \quad (3)$$

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (4)$$

$$\mathrm{explained\_variance}(y,\hat{y}) = 1 - \frac{Var[y - \hat{y}]}{Var[y]} \quad (5)$$

The performance of different ML and DL models was evaluated using various metrics. MAE and RMSE measure absolute prediction errors, while MAPE measures relative prediction

DOI : https://doi.org/10.5281/zenodo.17276775

errors. Smaller values indicate better prediction performance for these metrics. R^2 and EV values range from zero to one, with values closer to one indicating a better fit for the regression model. Table 1 presents the performance metrics for each model. The Multilayer Perceptron and Gradient Boosting models achieved R-Squared and Explained Variance above 0.93, MAE of 10.8, MAPE of 21%, and RMSE of 15.4. On the other hand, the Random Forest model had slightly lower R-Squared and Explained Variance values, with an MAE of 10.88, MAPE of 21%, and RMSE of 15.5. The GRU and LSTM models obtained R-Squared and Explained Variance near 0.92, with an MAE of 10.88, MAPE of 22%, and RMSE of 15.6. The Linear Regression model had R-Squared and Explained Variance values of 0.926, an MAE of 11.2, MAPE of 24%, and RMSE of 15.85. Lastly, the Stochastic Gradient model had R-Squared and Explained Variance of 0.9, an MAE of 12.8, MAPE of 29%, and RMSE of 18. To ensure reliable results, RNNs were trained iteratively ten times, and the average of each metric was calculated. For the ML models (scikit-learn), the random state was used to obtain consistent results each time.

Table 1. Comparison of performance metrics using the first dataset [30].

ML/DL Model	MAE	MAPE	RMSE	R^2	EV Score
MLP-NN	10.8281	21.1593%	15.4202	0.9304	0.9307
Gradient Boosting	10.8508	21.9493%	15.4121	0.9305	0.9306
Random Forest	10.8827	21.8392%	15.5481	0.9296	0.9297
GRU	10.8843	22.8492%	15.6191	0.9278	0.9295
LSTM	10.8806	22.3244%	15.6771	0.9267	0.9287
Linear Regression	11.2010	24.3238%	15.8545	0.9263	0.9264
Stochastic Gradient	12.8230	29.0075%	18.3727	0.9003	0.9004

Table 2. Performance metrics using the second dataset (PeMS) [25].

ML/DL Model	MAE	MAPE	RMSE	R^2	EV Score
MLP-NN	7.2427	18.2176	9.8096	0.9393	0.9395
Gradient Boosting	7.12151	17.6224	9.6648	0.941	0.941
Random Forest	7.05046	17.3788	9.5799	0.9421	0.9421
GRU	7.64266	18.5307	10.2406	0.9338	0.9381
LSTM	7.32852	19.0923	9.8816	0.9384	0.9388
Linear Regression	7.51693	20.3822	10.1914	0.9344	0.9344
Stochastic Gradient	8.39243	23.7443	11.3199	0.9191	0.9194

Upon further examination, the cost-benefit analysis of implementing the models involved calculating the average training time for each one. The experiments were conducted in the Google Colaboratory execution environment, with time tracking done using.Illustrates the training time for the seven ML models tested in the study. Notably, scikit learn models require less training time compared to RNNs. LSTM and GRU models had the longest training times at 321 and 250 seconds, respectively. Among the scikit learn models, Random Forest, Gradient Boosting, Linear Regression, Stochastic Gradient, and MLP-NN had training

DOI : https://doi.org/10.5281/zenodo.17276775

times of 46, 28, 20, 20, and 18 seconds, respectively. MLP-NN was the fastest model and also demonstrated superior performance metrics, as indicated in Table 1.

6. CONCLUSION

In this document, we presented various ML and DL models for predicting traffic flow at an intersection, setting the foundation for adaptive traffic control. We utilized two public datasets to train, validate, and test the models. The Multilayer Perceptron Regressor demonstrated superior performance with a shorter training time of 18 seconds. On the other hand, the Gradient Boosting Regressor performed similarly but required more processing time (28 seconds). Both RNNs and Random Forest Regressor achieved comparable scores, although RNNs had a longer training time (between 250 and 321 seconds). Linear Regression and Stochastic Gradient Regressor had efficient processing times (20 seconds) but exhibited the poorest performance among the models. All ML and DL models attained an EV Score and R-squared greater than 0.90, with MAE close to 10, RMSE near 15, and MAPE between 20 and 30 percent. Overall, the performance of the seven algorithms did not show significant differences. In summary, the results were satisfactory for predicting traffic flow in a four-lane intersection, indicating the potential for implementation in smart traffic light controllers.

REFERENCES

[1] Zhang, J., Wang, F. Y., Wang, K., Lin, W. H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 12(4), 1624-1639.

[2] Kim, Y. J., & Hong, J. S. (2015). Urban TFP system using a multifactor pattern recognition model. IEEE Transactions on Intelligent Transportation Systems, 16(5), 2744-2755.

[3] Boukerche, A., & Wang, J. (2020). Machine Learning-based traffic prediction models for Intelligent Transportation Systems. Computer Networks, 181(1), 1-21.

[4] Qureshi, K. N., & Abdullah, A. H. (2013). A survey on intelligent transportation systems. MiddleEast Journal of Scientific Research, 15(5), 629-642.

[5] Thomas, T., Weijermars, W., & Van Berkum, E. (2009). Predictions of urban volumes in single time series. IEEE Transactions on Intelligent Transportation Systems, 11(1), 71-80.

[6] Pan, T. L., Sumalee, A., Zhong, R. X., & Indra-Payoong, N. (2013). Short-term traffic state prediction based on temporal-spatial correlation. IEEE Transactions on Intelligent Transportation Systems, 14(3), 1242-1254.

[7] Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). TFP with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873.

[8] Mirzabeiki, V. (2013). An overview of freight intelligent transportation systems. International Journal of Logistics Systems and Management, 14(4), 473-489.

[9] Prabha, R., & Kabadi, M. G. (2016). Overview of data collection methods for intelligent transportation systems. The International Journal of Engineering and Science (IJES), 5(3), 16- 20. [10] Javed, M. A., Ben Hamida, E., & Znaidi, W. (2016). Security in intelligent transport systems for smart cities: From theory to practice. Sensors, 16(6), 879-904.

DOI : https://doi.org/10.5281/zenodo.17276775

[11] Sun, P., Aljeri, N., & Boukerche, A. (2020). Machine learning-based models for real-time TFP in vehicular networks. IEEE Network, 34(3), 178-185.
[12] Li, C., & Xu, P. (2021). Application on TFP of machine learning in intelligent transportation. Neural Computing and Applications, 33(2), 613-624.
[13] Yuan, T., Da Rocha, W., Rothenberg, C. E., Obraczka, K., Barakat, C., & Turletti, T. (2019). Machine learning for next-generation intelligent transportation systems: A survey. Transactions on Emerging Telecommunications Technologies,1(1),1-35.
[14] Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2021). A deep-learning model for urban TFP with traffic events mined from twitter. World Wide Web, 24(4), 1345-1368.
[15] Tian, Y., Zhang, K., Li, J., Lin, X., & Yang, B. (2018). LSTM-based TFP with missing data. Neurocomputing, 318(1), 297-305.
[16] Poonia, P., Jain, V. K., & Kumar, A. (2018). Short term TFP methodologies: a review. Mody University International Journal of Computing and Engineering Research, 2(1), 37-39.
[17] Z. Huang, Q. Li, F. Li and J. Xia. (2019). A Novel Bus-Dispatching Model Based on Passenger Flow and Arrival Time Prediction, IEEE Access, 7(1), 106453-106465.

DOI : https://doi.org/10.5281/zenodo.17276775