Adaptive-XGBoost for Improved Long-Term Multi-Step Traffic Flow Prediction

Mas Omar
Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia

Fitri Yakub
Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia

Andika Aji Wijaya
Department of Mechanical Engineering, University of Business & Technology

Azizul Azizan
Faculty of Artificial Intelligence, Universiti Teknologi Malaysia

他

https://doi.org/10.5109/7323333

出版情報：Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES). 10, pp.666-671, 2024-10-17. International Exchange and Innovation Conference on Engineering & Sciences
バージョン：

Adaptive-XGBoost for Improved Long-Term Multi-Step Traffic Flow Prediction

Mas Omar[1], Fitri Yakub[1], Andika Aji Wijaya[2], Azizul Azizan[3], Shamsul Faisal Mohd Hussein[4], Niranjana Govindan[5]
[1]Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, 54100, Kuala Lumpur, Malaysia,
[2]Department of Mechanical Engineering, University of Business & Technology, 21448, Jeddah, Saudi Arabia,
[3]Faculty of Artificial Intelligence, Universiti Teknologi Malaysia, 54100, Kuala Lumpur, Malaysia,
[4]Jabatan Teknologi Kejuruteraan Mekanikal, Universiti Teknikal Malaysia Melaka, 76100, Melaka, Malaysia,
[5]School of Computing, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamil Nadu, India.
Corresponding author email: masomar@graduate.utm.my

Abstract: *Long-term multi-step traffic flow prediction is crucial for effective traffic management. Recently, XGBoost has demonstrated its capability in multi-step prediction applications. However, optimal hyperparameter tuning using adaptive intelligent methods like the Particle Swarm Optimization (PSO) algorithm has yet to be explored. This paper presents an Adaptive eXtreme Gradient Boosting (Adaptive-XGBoost) model utilising the direct method with the combination of Grid Search and PSO algorithms for adaptive hyperparameter tuning for multi-step prediction. This approach aims to demonstrate the potential of Adaptive-XGBoost compared to existing deep learning models like Long-Short Term Memory (LSTM) and Transformer in terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Experimental results show that the Adaptive-XGBoost model achieved a 3.10% and 4.98% improvement in MAE over the Transformer and LSTM models, respectively, and a 0.26% and 5.69% improvement in RMSE. These findings highlight the potential of Adaptive-XGBoost for improved long-term multi-step traffic flow prediction.*

Keywords: Long-Term Multi-Step, Traffic Flow Prediction, XGBoost, Transformer, Long-Short Term Memory.

1. INTRODUCTION

In recent years, with the rising number of vehicles in people's daily lives, traffic congestion in densely populated urban areas has become unavoidable. This issue substantially affects the economy and the environment, worsening air pollution and posing a possible threat to human health [1, 2]. One solution to these problems is to mitigate traffic congestion using the Intelligent Transportation System (ITS), which enables traffic management to plan for a more effective urban transportation plan [3]. Accurate traffic flow prediction models are at the core of ITS, providing critical insights for anticipating traffic conditions and making informed decisions to mitigate congestion [4].

Multi-step predictions are crucial for practical traffic management applications, offering more utility than single-step prediction models. Current research predominantly focuses on enhancing deep learning models, such as Long-Short Term Memory (LSTM) and Transformer-based models. These models are especially effective for capturing temporal dependencies in sequential data, making them popular choices for time series predicting tasks, including traffic prediction. For instance, Doğan [5] analysed the performance of LSTM models with various multi-step ahead prediction strategies for traffic flow prediction. Wei and Liu [6] enhance the LSTM capability by proposing a convolutional LSTM network integrated with a multi-head attention mechanism, demonstrating significant improvements in traffic flow prediction accuracy. Fernandes et al. [7] explored input variables, time frames, and multi-step approaches in LSTM models, highlighting the superior performance of the LSTM model for multi-step, multi-variate traffic flow prediction. On the other hand, Transformer-based models, which have revolutionised natural language processing [8], are now being adapted for traffic prediction due to their ability to manage long-range dependencies and parallelise computations. Reza et al. [9] presented a multi-head attention-based transformer model for traffic flow prediction, outperforming traditional Recurrent Neural Network models in traffic flow prediction tasks. Xing et al. [10] introduced the Spatial Linear Transformer with Temporal Convolution Network to optimise the self-attention mechanism and reduce computational costs. Similarly, Wang et al. [11] developed the Spatiotemporal Fusion Transformer, incorporating Seasonality Encoding, Tubelet Embedding, and Token Permutator modules to enhance traffic flow prediction performance. Lim et al. [12] introduced the Temporal Fusion Transformers for multi-horizon time series prediction, using attention mechanisms for high-performance prediction and interpretable insights.

While deep learning models like LSTM and Transformer-based models have shown great promise for multi-step prediction applications, the potential of models inherently built for single-step prediction, such as XGBoost [13], remains underexplored. Historically, XGBoost was only used for single-step prediction, as demonstrated by Cao et al. [14]. This capability made it suitable for predicting the next single-step traffic flow value in the near future. Subsequently, its application was extended by integrating XGBoost with LSTM for multi-step traffic flow predictions, a hybrid approach proposed by Zhang et al. [15], enhancing accuracy and robustness. However, this research was tested only on short-term predictions of less than a 1-hour interval. Recent evidence suggests that the XGBoost model can also adapt for long-term multi-step predictions for hourly interval data, showing superior results to LSTM, as revealed by Tsalikidis et al. [16]. These capabilities were evaluated using the Grid Search algorithm for hyperparameter tuning. However, applying more adaptive methods like Particle Swarm Optimization (PSO) remains unexplored.

Furthermore, to the best of current knowledge, the comparison of the XGBoost model with the Transformer model has also not been investigated.

Therefore, this paper addresses this gap by presenting a novel approach for long-term multi-step traffic flow prediction using the direct method with the fusion of Grid Search and Particle Swarm Optimization (Direct-PSOGS) algorithm introduced by Omar et al. [17] for proposing an Adaptive eXtreme Gradient Boosting (Adaptive-XGBoost) model. The proposed Adaptive-XGBoost model aims to capture complex temporal patterns in long-term traffic data, demonstrating the model potential in a multi-step prediction scenario on a single road. This research also includes a comparative analysis with LSTM and Transformer models. The paper results show that the proposed model is superior to LSTM and Transformer models in terms of accuracy, making it suitable as an alternative improved long-term multi-step traffic flow prediction model for the ITS.

The structure of the paper is as follows. Section 2 covers the fundamental concepts necessary for comprehension, including the proposed prediction model architecture principle and a general overview of the Transformer and LSTM model. Section 3 details the dataset and the experimental simulation settings. Section 4 presents the results and discussion. Finally, Section 5 offers conclusions and suggestions for future research directions.

2. METHODOLOGY

2.1 Principles of XGBoost

XGBoost is one of the most popular, powerful, and effective gradient-boosting methods created by Chen and Guestrin [13], based on the Gradient Boosting Decision Tree proposed by Friedman [18]. The principal basis of XGBoost can be understood by first looking at its objective function, Γ which is given by:

$$\Gamma = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) +, \sum_{K=1}^{K} \Omega\left(f_k\right) \tag{1}$$

In this formula, y_i represents the actual value, and $\hat{y}_i$ represents the predicted value. The first part of the objective function is the training loss, l which measures how well the model fits the training data. The second part is the sum of the complexities of each tree, where the complexity of the k-th tree is given by:

$$\Omega\left(f_k\right) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{2}$$

Here, T indicates the number of leaf nodes; w is the leaf weight, γ is the penalty coefficient for the number of leaf nodes, and λ is the penalty coefficient for the leaf weight. The methodology for XGBoost can be broken down into the following steps. Given a dataset of $D = \{(x_i, y_i): i = 1,2,\dots,m, x_i \in R^n, y_i \in R\}$ containing m samples with n-dimensional features, a model composed of K decision trees can be represented as $\hat{y}_i$:

$$\hat{y}_i = \sum_{K=1}^{K} f_k(x_i), f_k \in F \tag{3}$$

Where F denotes the function space consisting of all tree models, while f_k represents each regression tree.

$$F = \{f(x) = w_{q(x)}\}(q: R^n \to T, w \in R^T) \tag{4}$$

Here, q represents the mapping relationship between x and the leaf node, whereas w represents the weight assigned to the leaf node. At the new iteration, the existing model predicted value is continuously improved by adding newly developed regression tree output to approach the actual value. Suppose the predicted result of the i-th sample in the t-th iteration is $\hat{y}_i^{(t)}$, and the newly added regression tree is $f_t(x_i)$, then:

$$\hat{y}_i^{(t)} = \sum_{K=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{5}$$

Where $\hat{y}_i^{(t-1)}$ is the predicted value of the model in the round $t-1$, and $f_t(x_i)$ is the function newly added in the round t. The initial state prediction is always set to $\hat{y}_i^{(0)} = 0$. By substituting this into the objective function, it becomes:

$$\Gamma^{(t)} = \sum_{i=1}^{m} l[y_i, \hat{y}_i^{(t-1)} + f_t(x_i)] + \Omega(f_t) + c \tag{6}$$

Where c is a constant term. Conducting the second-order Taylor expansion on the objective function and introducing regularisation terms, obtaining:

$$\Gamma^{(t)} = \sum_{i=1}^{m} \left[l y_i, \hat{y}_i^{(t-1)} + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) + c \tag{7}$$

Where $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$, $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ are the first and second derivatives of the objective function, respectively, and c is a constant term. Remove the constant term to get:

$$\tilde{\Gamma}^{(t)} \cong \sum_{i=1}^{m} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{8}$$

By leveraging the gradient boosting strategy, the iterative process of adding new decision trees minimizes the defined objective function and enhances predictive performance until the model's stopping criterion is reached.

2.2 Direct multi-step prediction strategy

In this paper, the direct multi-step prediction approach has been selected to predict the next 24-hour horizon. This approach directly predicts multiple future time steps rather than relying on iteratively predicting one step at a time. By independently modelling each future time step, the direct method can provide more stable and accurate predictions for longer horizons. By defining the predicted horizon time step data as $\{\hat{\phi}_{t+1}, \dots, \hat{\phi}_{t+H}\}$ and the available observations data as $\{\phi_t, \dots, \phi_{t-L}\}$, in which the $H > 1$ represent both the total number of horizon values and the independent number of models that are trained separately during the model training process and L is the total past values, the direct multi-step strategy can be modelled as:

$$\phi_{t+h} = f_h(\phi_t, \phi_{t-1}, \dots, \phi_{t-L}) + \omega_h \quad \text{for } h \in \{1, \dots, H\} \tag{9}$$

Here f_h and ω_h denotes the model learned function and additive noise, respectively, for every model's output, h. The present work will utilise the single training step strategy recently highlighted by [15], which is better than the horizon training step strategy. Here, H different models that also correspond to the total prediction horizon values are trained for predicting the output, $\hat{\phi}_{t+1}$ to $\hat{\phi}_{t+H}$ as

$$\hat{\phi}_{t+h} = \hat{f}_h(\phi_t, \phi_{t-1}, \dots, \phi_{t-L}) \quad \text{for } h \in \{1, \dots, H\} \tag{10}$$

The illustration of input and prediction output for the direct multi-step prediction strategy is shown in Fig. 1.

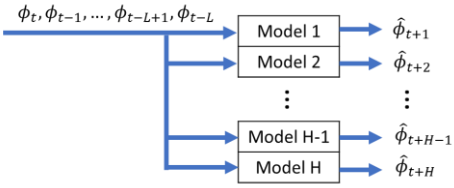

Fig. 1. Direct multi-step prediction strategy.

2.3 Grid Search Algorithm

Grid Search is a widely used hyperparameter tuning technique in machine learning to optimise model performance. Fig. 2 shows the overview of the Grid Search algorithm.

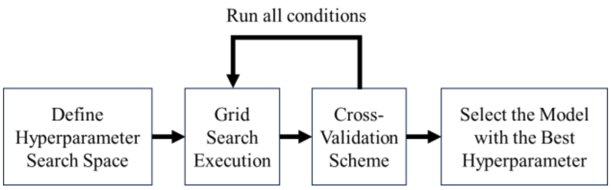

Fig. 2. Overview of the Grid Search Algorithm.

It systematically works through multiple combinations of parameter values, cross-validating to determine which set of parameters produces the best results.

2.4 Particle Swarm Optimization Algorithm

Particle Swarm Optimization is an adaptive intelligent method inspired by the social behaviour of birds flocking or fish schooling. It optimises a problem by iteratively trying to improve a candidate solution concerning a given quality measure. The PSO algorithm update equations are as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 \big(p_i - d_i(t)\big) + c_2 r_2 \big(g - d_i(t)\big) \tag{11}$$

$$d_i(t+1) = d_i(t) + v_i(t+1) \tag{12}$$

Where v_i is the velocity, d_i is the position, p_i is the personal best position, g is the global best position, ω is the inertia weight, c_1 and c_2 are the cognitive and social coefficients, and r_1 and r_2 are the random values between 0 and 1. Fig. 3 shows the overview of the PSO algorithm.

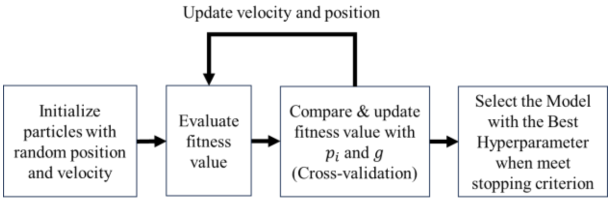

Fig. 3. Overview of the PSO Algorithm.

The algorithm repeatedly updates the velocity and position steps until a stopping criterion is met. The stopping criterion used in this paper is the maximum number of iterations run. The best position the particles encounter during the search represents the optimal solution.

2.5 Adaptive-XGBoost

The proposed Adaptive-XGBoost model in this paper is built upon integrating three advanced vital components: the direct model, Grid Search, and PSO algorithm. The novel hybrid optimisation algorithm, named the Particle Swarm Optimization Grid Search (PSOGS) algorithm, was first introduced by Açıkkar and Altunkol [19]. PSOGS is a redesign of PSO that can operate on a discrete search space. PSO is executed on the promising region identified by Grid Search. Thus, integrating PSO will fine-tune the hyperparameters by exploring the search space more adaptively and efficiently. Each proposed Adaptive-XGBoost using the direct method will be optimised using the PSOGS algorithm. Fig. 4 below shows the architecture of the proposed singular Adaptive-XGBoost model for optimisation before applying the direct method.

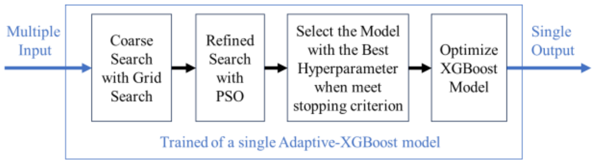

Fig. 4. Single Adaptive-XGBoost model during the training process for the direct multi-step prediction.

2.6 Transformer

The Transformer model, introduced by Vaswani et al. in 2017, revolutionised the field of natural language processing by enabling parallel processing of sequence data [8]. The Transformer architecture consists of an encoder and a decoder, each composed of multiple layers. The encoder processes the input sequence and generates a set of attention-based representations. The decoder uses these representations to generate the output sequence. Each layer in the encoder and decoder has two main components: a multi-head self-attention mechanism and a position-wise feed-forward network. The multi-head self-attention mechanism allows the model to attend to different parts of the sequence simultaneously while the feed-forward network applies transformations independently to each position.

2.7 Long-Short Term Memory

The LSTM is a deep learning model capable of learning long-term dependencies, introduced by Hochreiter and Schmidhuber in 1997 [20]. The model is explicitly designed to avoid the long-term dependency problem, making them well-suited for time series prediction tasks, where predicting future values is highly dependent on past information [21]. The key to LSTM is their ability to maintain a cell state over time. This cell state acts like a conveyor belt, running through the sequence chain with minor linear interactions. LSTM can selectively remember or forget information through their gates: the forget gate, input gate, and output gate. as shown in Fig. 5. Where X_t denotes the input sequence, while z_t represent the hidden state and C_t is the updated cell state.

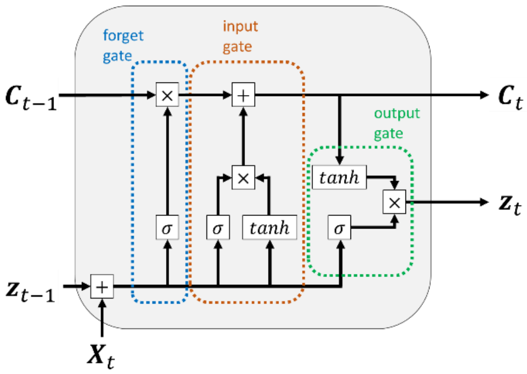

Fig. 5. The forget, input and output gate of the LSTM.

3. EXPERIMENTAL SETUP

3.1 Dataset description

A real-time traffic flow dataset with hourly data was used in this paper. The source of the dataset is stated in [22]. The dataset represents the traffic flow data from the westbound lanes of Interstate 94 at ATR station 301 in Minnesota. The dataset was filtered to consider only the weekday traffic flow from January to March 2018, yielding 1561 observable data points. This selection was explicitly made to focus on standard workweek traffic patterns, which are critical for accurate traffic flow prediction. Any missing data was imputed by the linear interpolation method, ensuring a complete and consistent dataset for analysis. The whole dataset except for the last five days of March 2018 (Monday to Friday) was used as the train set. In the train set, 35 days (840 hours) from 1st January to 16th February 2018 was used as the training set, while the next 25 days (600 hours) from 19th February to 23rd March 2018 was used as the validation set. The last five days (120 hours) for the final weekdays of March 2018 were used as the test set.

3.2 Dataset processing

The min-max normalisation method had been used to scale the input data, adjusting the feature values to fall within a specific range [0, 1]. This method ensures that each feature contributes proportionately to the model's prediction results. The final predicted results were applied inverse normalisation to ensure they returned to the original scale, allowing for a direct and meaningful comparison with measured traffic flow values. The normalisation and inverse normalisation formulas are shown below:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{11}$$

$$x_{scale} = x_{norm} \times (x_{max} - x_{min}) + x_{min} \tag{12}$$

where x is the original value of the feature, x_{min} is the minimum value of the training dataset, x_{max} is the maximum value of the training dataset, x_{norm} is the value after normalisation, and x_{scale} represents the value of the original scale.

3.3 Model evaluation metrics

The metrics that evaluate all model's performances are the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The MAE provides a precise measure of average prediction error in the same units as the data and the RMSE focuses on the magnitude of prediction errors.

3.4 Model parameter setting

The combination of different parameter values significantly impacts the proposed model's prediction accuracy, making parameter tuning crucial for accurate predictions.

Table I lists the essential parameters of the search space of the Adaptive-XGBoost model that are being focused on, which include max_depth, learning_rate, n_estimators, reg_lambda, and gamma. The max_depth controls the maximum depth of the trees; deeper trees can capture more complex patterns but may lead to overfitting. The learning_rate determines the step size during the model update; smaller values require more trees but can improve model accuracy. The n_estimators refer to the number of trees to be built, where increasing the number of trees can improve the model but also increase computational complexity. The reg_lambda, which is the ridge regularisation term on weights, helps in controlling the model's complexity and prevents overfitting. Finally, gamma represents the minimum loss reduction required to partition a leaf node further, with higher values leading to fewer splits and a more conservative model. Other hyperparameter settings follow the default values of the Python sklearn XGBoost hyperparameter settings in [23]. For the PSO algorithm, the ω, c_1 and c_2 values are set to 0.5, the default value of Python's pyswarm library. The number of iterations and particles are fixed at 100 to ensure an optimal search run.

Table I. Adaptive-XGBoost Parameters Search Space

Parameter	Values
max_depth	[3, 6, 9]
learning_rate	[0.01, 0.1, 0.2, 0.3]
n_estimators	[50, 100, 200]
reg_lambda	[0, 0.1, 1]
gamma	[0, 0.1, 1]

By specifying the proposed model's search space, the PSOGS algorithm can adaptively find the best hyperparameter combination in the above search space.

Inspired by reference [9], the Transformer model has redefined hyperparameter values for optimal performance. Specifically, it utilises 2 attention heads, an embedding dimension of 32, a feed-forward dimension of 128, and includes 1 encoder and 1 decoder block. The output features have also been adjusted to 24, which is suitable for the predicted horizon length. The LSTM model architecture and hyperparameter settings are based on reference [17].

4. RESULTS AND DISCUSSION

Table II shows the best performance results for the proposed Adaptive-XGBoost compared with the Transformer and LSTM for hourly predictions over a 24-hour horizon across five days.

Table 2. Performance comparison of different models

Model	MAE	RMSE
Adaptive-XGBoost	**247.35**	**329.13**
Transformer [9]	255.26	329.98
LSTM [17]	260.31	348.99

The results in Table 2 show that the Adaptive-XGBoost model outperforms both the Transformer and LSTM models in terms of MAE and RMSE. Specifically, the Adaptive-XGBoost model achieved a 3.10% improvement in MAE over the Transformer and a 4.98% improvement over the LSTM. Additionally, the model achieved a 0.26% improvement in RMSE over the Transformer and a 5.69% improvement over the LSTM. These findings indicate that the proposed Adaptive-XGBoost model is more accurate and reliable for multi-step traffic flow prediction over the specified period. These results demonstrate the model's suitability for applications requiring high-precision traffic flow predictions. All the models predicted values and the actual unseen values are plotted in Fig. 6.

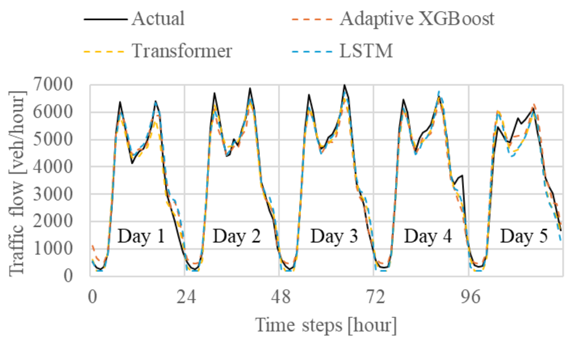

Fig. 6. Plot of prediction results for all models.

By observing the plotted line graph of all the models compared to the actual unseen data, the proposed Adaptive-XGBoost can capture the traffic patterns over all five days relatively reliably, similar to popular multi-step prediction models like LSTM and Transformer within the family of deep learning.

5. CONCLUSION AND FUTURE WORK

In this paper, an Adaptive-XGBoost model was proposed for long-term multi-step traffic flow prediction. The model leverages a combination of Grid Search and PSO algorithms for hyperparameter tuning, aiming to improve the accuracy and reliability of predictions. Experimental results demonstrated that the Adaptive-XGBoost model outperforms traditional deep learning models like LSTM and Transformer regarding MAE and RMSE. Specifically, the Adaptive-XGBoost model achieved a 3.10% and 4.98% improvement in MAE over the Transformer and LSTM models, respectively, and a 0.26% and 5.69% improvement in RMSE. The proposed model's ability to capture complex traffic patterns and exemplary performance in multi-step prediction tasks highlight its potential for practical applications in the ITS. The Adaptive-XGBoost model can help in effective traffic management and urban planning, reducing congestion and improving overall traffic flow by providing accurate long-term multi-step traffic flow prediction.

While the results are promising, there are several avenues for future research to enhance the model and its applications further. One possible direction is to investigate other advanced optimization techniques, such as Genetic Algorithms or Bayesian Optimization, to improve the model's performance further. These techniques could provide a more efficient and practical approach to hyperparameter tuning, potentially leading to more accurate and reliable traffic flow predictions. Additionally, exploring these optimization methods could reveal insights into the strengths and weaknesses of different approaches, contributing to the broader field of multi-step traffic flow prediction research. Another potential area of research is to extend the model to include additional external factors, such as weather conditions, special events, or road incidents, which can significantly impact traffic flow and improve multi-step prediction accuracy. Incorporating these factors into the model could enhance its ability to capture the complex dynamics of real-world traffic systems. For instance, weather conditions like heavy rain or snow can drastically reduce traffic speeds, while special events such as concerts or sports games can lead to sudden spikes in traffic volume. Road incidents, including accidents or construction work, can cause unexpected delays and congestion. By integrating these variables, the model could provide more robust and contextually aware predictions, ultimately aiding in more effective traffic management and urban planning.

6. REFERENCES

[1] Z. Si, M. A. Hossain, J. Tanimoto, Linear stability analysis of an improved car-following model considering vehicle's inertia effect, in 8th International Exchange and Innovation Conference on Engineering and Sciences, IEICES 2022, Kyushu, Japan. (2022) 196-200.

[2] B. M. Metran, J. P. Elumba, M. C. Alvardo, M. M. Bermoy, Mapping and Estimation of Road Traffic Emissions in Caraga Region using Sentinel 5P Satellite Images, in 9th International Exchange and Innovation Conference on Engineering and Sciences, IEICES 2023, Kyushu, Japan. (2023) 6-12.

[3] U. Jilani, M. Asif, M. Y. I. Zia, M. Rashid, S. Shams, P. Otero, A Systematic Review on Urban Road Traffic Congestion, Wireless Personal Communications. (2023) 1-29.

[4] L. Zhang, M. Khalgui, Z. Li, Predictive intelligent transportation: Alleviating traffic congestion in the internet of vehicles, Sensors. 21 (2021) 1-21.

[5] E. Doğan, Performance analysis of LSTM model with multi-step ahead strategies for a short-term traffic flow prediction, Scientific Journal of Silesian University of Technology. Series Transport. 111 (2021) 15-31.

[6] Y. Wei, H. Liu, Convolutional Long-Short Term Memory Network with Multi-Head Attention Mechanism for Traffic Flow Prediction, Sensors. 22 (2022) 1-15.

[7] B. Fernandes, F. Silva, H. Alaiz-Moreton, P. Novais, J. Neves, C. Analide, Long Short-Term Memory Networks for Traffic Flow Forecasting: Exploring Input Variables, Time Frames and Multi-Step Approaches, Informatica. 31 (2020) 723-749.

[8] A.Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin,

Attention Is All You Need, in 31st Annual Conference on Neural Information Processing Systems, NIPS 2017, Long Beach, CA, USA. (2017) 5999-6009.

[9] S. Reza, M. C. Ferreira, J. J. M. Machado, J. M. R. S. Tavares, A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks, Expert Systems with Applications. 202 (2022) 1-11.

[10] Z. Xing, M. Huang, W. Li, D. Peng, Spatial linear Transformer and temporal convolution network for traffic flow prediction, Scientific Reports. 14 (2024) 1-12.

[11] Z. Wang, Y. Wang, F. Jia, F. Zhang, N. Klimenko, L. Wang, Z. He, Z. Huang, Y. Liu, Spatiotemporal Fusion Transformer for large-scale traffic forecasting, Information Fusion. 107 (2024) 1-11.

[12] B. Lim, S. Ö. Arık, N. Loeff, T. Pfister, Temporal Fusion Transformers for interpretable multi-horizon time series forecasting, International Journal of Forecasting. 37 (2021) 1748-1764.

[13] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, San Francisco, California, USA. (2016) 785-794.

[14] J. Cao, G. Cen, Y. Cen, W. Ma, Short-term highway traffic flow forecasting based on XGBoost, in 15th International Conference on Computer Science and Education, ICCSE 2020, Delft, Netherlands. (2020) 359-362.

[15] X. Zhang, Q. Zhang, Short-Term Traffic Flow Prediction Based on LSTM-XGBoost Combination Model, CMES - Computer Modeling in Engineering and Sciences. 125 (2020) 95-109.

[16] N. Tsalikidis, A. Mystakidis, P. Koukaras, M. Ivaškevičius, L. Morkūnaitė, D. Ioannidis, P. A. Fokaides, C. Tjortjis, D. Tzovaras, Urban Traffic Congestion Prediction: A Multi-Step Approach Utilising Sensor Data and Weather Information, Smart Cities. 7 (2024) 233-253.

[17] M. Omar, F. Yakub, S. S. Abdullah, M. S. A. Rahim, A. H. Zuhairi, N. Govindan, One-step vs horizon-step training strategies for multi-step traffic flow forecasting with direct particle swarm optimisation grid search support vector regression and long short-term memory, Expert Systems with Applications. 252 (2024) 1-15.

[18] J. H. Friedman, Greedy function approximation: A gradient boosting machine, Annals of Statistics. 29 (2001) 1189-1232.

[19] M. Açıkkar, Y. Altunkol, A novel hybrid PSO- and GS-based hyperparameter optimisation algorithm for support vector regression, Neural Computing and Applications. 35 (2023) 19961-19977.

[20] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation. 9 (1997) 1735-1780.

[21] M. S. A. Rahim, F. Yakub, M. Omar, R. A. Ghani, I. Dhamanti, S. Sivakumar, Prediction of Influenza A Cases in Tropical Climate Country using Deep Learning Model, in 2nd IEEE National Biomedical Engineering Conference, NBEC 2023, Melaka, Malaysia. (2023) 188-193.

[22] UC Irvine Machine Learning Repository, Metro Interstate Traffic Volume website: https://archive.ics.uci.edu/dataset/492/metro+interstate+traffic+volume (accessed 07.07.24).

[23] Xgboost Developers, XGBoost Documentation website: https://xgboost.readthedocs.io/en/stable/ (accessed 07.07.24).