TRAFFIC VOLUME FORECAST MODELS WITH HIGH SENSOR DATA UNCERTAINTY

K. Jinakub[1], J Prabnasak[2], P. Ngamdee[3] and S. Boonsiripant[4*]

[1,2] *Department of Civil Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, THAILAND*

[3] *Traffic Engineer, Mobinary Co., Ltd., THAILAND*

[4] *Department of Civil Engineering, Faculty of Engineering, Kasetsart University, Bangkok, THAILAND*

**Corresponding author; E-mail address: saroch.b@ku.th*

Abstract

Traffic volume forecasting is an important task for the motorway planning and management. The performance of these forecasts is often degraded by the high uncertainty of sensor data, particularly when the data are subject to delay. This study aims to develop methods for imputing and forecasting traffic volume under high uncertainty and delayed data conditions. The objective is to enhance the precision of predictions for traffic volume. This study introduced a new data imputation method as well as a sequence-based machine learning model, namely, Long Short-term Memory (LSTM) model, to handle highly uncertain sensor data. The model's performance is evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) with a result of 26.67 vehicles, 17.31 vehicles and 9.26% respectively. Specifically, the model demonstrated a high level of sensitivity to delayed data, 17.45% of delayed data, meaning that it was able to accurately adjust its predictions based on changes in data availability and processing times. This suggests that the proposed approach has the potential to significantly improve the accuracy and reliability of traffic volume forecasting in real-world settings, where delays and disruptions are common occurrences. Overall, our study provides strong evidence for the efficacy of the proposed approach in the face of delayed data and highlights its potential as a valuable tool for traffic management and planning in Thailand and beyond.

Keywords: Traffic volume forecasting model, Long Short-term Memory, data imputation.

1. Introduction

Traffic volume forecasting is an essential task for transportation planning and management. Accurate predictions of traffic volume help authorities to make informed decisions on road network planning, traffic control, and infrastructure investments. However, forecasting traffic volume is a challenging task due to the high degree of uncertainty associated with sensor data. Inaccurate sensor data can lead to incorrect predictions and have significant consequences for transportation planning.

The main challenge in traffic volume forecasting is dealing with high sensor data uncertainty. Traffic sensors are often subject to various factors such as environmental conditions, technical faults, and human errors that can affect their accuracy. The aim of this study is to improve the accuracy of traffic volume predictions in the context of a dataset that exhibits a high proportion of missing data, with roughly 80% of the values being absent.

To address the issue of high uncertainty and missing values in traffic data, the kNN (k-Nearest Neighbors) method is a popular traditional method for imputation. Its relative simplicity and potential for producing accurate results when implemented correctly make it a preferred choice. An example of this method can be seen in a comparison study of k-Nearest Neighbors (kNN) with Nearest History (NH) and Bootstrap-based Expectation, which demonstrated that kNN outperformed the other methods when dealing with missing data ranging from 0.1% to 50%[1]. Another example of kNN, this research paper evaluates and compares the effectiveness of three missing data imputation methods for traffic flow prediction. The k-nearest neighbor (kNN) method, singular value decomposition (SVD), and autoregressive integrated moving average (ARIMA) models are evaluated using a real-world traffic flow dataset with missing values ranging from 10% to 50%. The study analyzes the performance of each method in terms of imputing missing values and predicting traffic flow. The results show that the kNN method outperforms the SVD and ARIMA methods in terms of accuracy. Moreover, the study finds that the kNN method's performance is influenced by the number of nearest neighbors used, and selecting the optimal number of neighbors can further improve the accuracy of the imputation results [2]. However, kNN method may not be optimal for imputing data with missing values exceeding 80%. The computational complexity of kNN imputation is a major challenge, as it requires calculating distances between the missing data point and all other data points in the dataset. This process can be computationally expensive, particularly for large datasets and real-time applications where speed is critical. Furthermore, the availability of nearest neighbors in real-time is another challenge, as kNN imputation relies on the values of the k nearest neighbors to impute missing data. In a real-time scenario, the nearest neighbors may not be available immediately, and there may be delays in calculating the imputed value.

To address this problem, firstly, a waiting time of 30-minute approach is proposed to reduce the limitation of high missing data from 80% to 10%. Then, the proposed approach aims to leverage historical traffic data and employ Long Short-Term Memory (LSTM) algorithms for missing data imputation and traffic volume prediction [3]-[10]. LSTM (Long Short-Term Memory) is a type of recurrent neural network that is capable of modeling sequential data, making it a powerful tool for time-series prediction and forecasting tasks. Unlike kNN (k-Nearest Neighbors) imputation, which is a simple imputation method that relies on the values of the nearest neighbors to fill in missing data, LSTM can learn the patterns and relationships within the data and use them to impute missing values more accurately. One of the main advantages of LSTM over kNN imputation is its ability to handle missing data patterns that are not random. In contrast, kNN imputation assumes that the missing data is missing at random (MAR), which may not always be the case in real-world datasets. LSTM can learn the underlying patterns in the data and use them to impute missing values, even when the missing data patterns are not MAR. Another advantage of LSTM is its ability to handle time-series data and model temporal dependencies. This makes it particularly useful for imputing missing values in time-series datasets, where the missing data points are often correlated with nearby data points in time. kNN imputation does not take into account the temporal relationships between the data points, and as a result, may not be able to accurately impute missing values in time-series datasets.

LSTM (Long Short-Term Memory) networks can be used to deal with missing input data. One way to do this is by using a technique called "masked input," where missing values are marked with a special value, and the LSTM learns to ignore those values during training and prediction. Another approach is to use an LSTM-based imputation model to fill in the missing values before feeding the data into the main LSTM model. LSTM algorithms are well-suited for time-series data, making them a powerful tool for forecasting traffic volume. This approach has the potential to be cost-effective and efficient, while still producing reliable and accurate traffic volume forecasts.

2. Related Works

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is widely used for time series analysis and sequential data processing. The LSTM algorithm can effectively capture and comprehend the patterns and relationships present in data, allowing for accurate imputation of missing values. This approach has been shown to outperform other methods such as mean imputation, linear interpolation and k-Nearest Neighbors (kNN).

Smoothed LSTM-AE was used as a method for imputing missing values in multiple time-series data using a combination of long short-term memory (LSTM) and autoencoder (AE) models. The approach was designed to capture temporal dependencies and spatial correlations within the data and to handle missing data that occurs in long consecutive intervals. The paper provided a comprehensive experimental evaluation of the proposed method, comparing it to several state-of-the-art methods on several benchmark datasets. The results demonstrated that the proposed approach outperformed the compared methods, including k-nearest neighbors (kNN) and other deep learning-based methods, in terms of imputation accuracy and ability to handle long consecutive missing data [3].

Long Short-Term Memory (LSTM) neural networks were utilized for traffic flow prediction with missing data. The methodology involved three main steps, namely data preprocessing, LSTM-based prediction model construction, and model evaluation. In the first step, the missing values in the data were imputed using a k-nearest neighbor (kNN) algorithm, followed by normalization using a min-max scaler. In the second step, the LSTM-based prediction model was constructed, consisting of an LSTM layer and a dense layer. The model was trained using the mean squared error (MSE) loss function and the Adam optimization algorithm. Finally, the model's performance was evaluated using various metrics such as root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The paper compared the proposed LSTM-based approach's performance with a baseline autoregressive integrated moving average (ARIMA) model and a feedforward neural network (FNN) model. The results showed that the LSTM-based model outperformed the other two models in terms of prediction accuracy, especially in dealing with missing data. Based on the findings, the authors concluded that the proposed LSTM-based approach was effective in traffic flow prediction with missing data and could be used for traffic management and control [4].

The combination of bidirectional and unidirectional Long Short-Term Memory (LSTM) networks in a stacked configuration was used to model temporal dependencies in the traffic flow data. The model was trained on a large dataset of traffic flow data and tested on various scenarios with missing data. The performance of the proposed approach was compared with various benchmark methods, such as Support Vector Regression (SVR) and k-Nearest Neighbors (kNN), and it was shown that the proposed approach outperformed the benchmark methods, especially in scenarios with high percentages of missing data.The results suggest that LSTM is better than kNN for predicting network-wide traffic state with missing values, using the combination of bidirectional and unidirectional LSTM networks in a stacked configuration to capture temporal dependencies in the data [5].

A Type-2 fuzzy LSTM (T2F-LSTM) neural network model was used for long-term traffic volume prediction. T2F Sets (T2FSs) were utilized to provide greater flexibility in describing membership information and processing data with higher uncertainty, compared to traditional fuzzy systems. The model introduced interval T2FSs to extract probability distributions and spatial-temporal characteristics of traffic volume. The closure of support parameters obtained from the interval T2FSs was then used to update and converge the weights of the input gate in the LSTM neural network to regions with a larger slope of the sigmoid function. This resulted in faster convergence and increased network interpretability, achieved through better control of the information flow using motivational factors constructed from the parameters. Overall, the proposed T2F-LSTM model offered improved accuracy and interpretability for long-term traffic volume prediction [6].

The KNN-LSTM model, which combines k-nearest neighbor (KNN) and long short-term memory network (LSTM) techniques is used as a spatiotemporal traffic flow prediction method that effectively improves prediction accuracy. KNN is used to identify neighboring stations that are closely related to the test station, capturing spatial features of traffic flow. LSTM is then used to mine the temporal variability of traffic flow, with a two-layer LSTM network applied to predict traffic flow in the selected stations. By combining these two techniques, the KNN-LSTM model is shown to achieve high prediction accuracy, enabling more effective traffic guidance and management [7].

The long short-term memory (LSTM) recurrent neural network was used to analyze the effects of various input settings on the LSTM prediction performances. Predicting traffic based solely on flow data may not yield optimal results. Therefore, in this study, a combination of flow, speed, and occupancy data from the same detector station was used as inputs to improve prediction performance. The results showed that the inclusion of occupancy/speed information helped to enhance the performance of the model overall [8].

A long short-term memory neural network (LSTM) was used as a method for predicting network traffic volume. The method used observed traffic volume changes, time window indices, and a seasonality factor as input features to predict future traffic volume. Results from experiments with real datasets showed that this method outperformed other time series forecasting methods for predicting upcoming network traffic [9].

Recurrent neural networks (RNNs) were used as a model to develop robust, multi-step-ahead forecasting models. The models utilized simple RNN, gated recurrent unit (GRU), and long short-term memory (LSTM) units, and two approaches were used to address the missing value issue: masking and imputation, in conjunction with the RNN models. The datasets used in this study were characterized by long-term temporal dependencies and missing values, which posed challenges for traditional time-series forecasting models. Seasonal variations occurred in all roadways, which could be weekly, monthly, or yearly. Based on the analysis, it was concluded that the LSTM model outperformed the simple RNN and GRU models in predicting future traffic volume. Additionally, the study found that imputation was a more effective approach than masking for handling missing values in the dataset [10].

3. Data

3.1 Study corridor

The data was collected from microwave radar and image processing data installed along 25 km. of the motorway number 7 which sees approximately 80,000 vehicles passing through each day. The data were collected between KM. 0+000 and KM. 25+000 (Ladkrabang to Srinagarindra direction).

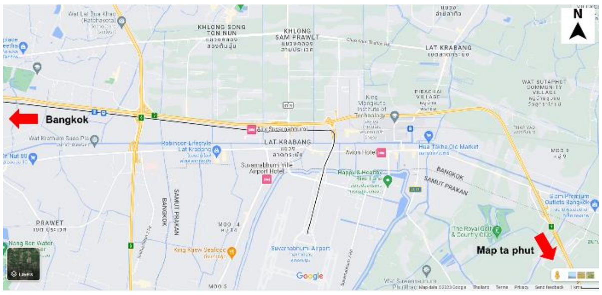

Fig. 1 The study corridor from Ladkrabang to Srinagarindra, Bangkok.

The 25 km motorway corridor was divided into 8 sections based on the interchanges, resulting in the midblock segments, 2-5 km in length. However, in this study, Segment W03, KM 9+700 to KM 9+900, westbound, was selected.

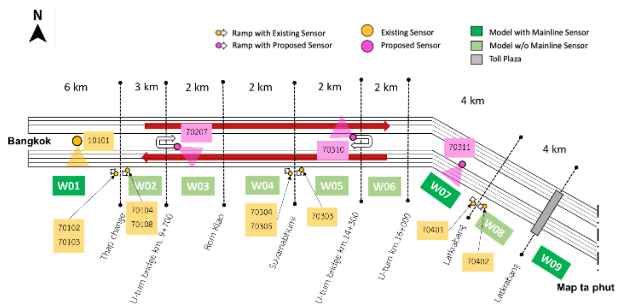

Fig. 2 Sensor locations along the study corridor

3.2 Data collection

Traffic volume was collected from 14 sensors, which were composed of 2 mainline sensors and 12 ramp sensors. The traffic volume data was collected on 112 days between November 2022 to February 2023. Long holidays were intentionally included in the study period to train the model under various traffic conditions.

3.2.1 Image processing camera

An image processing camera is located on the mainline at KM 3+000 and is used to capture images and videos of traffic flow on the motorway. The camera is equipped with the software that can detect and measure traffic volume, as well as monitor traffic conditions such as congestion and accidents.

Fig. 3 The installation of an image processing camera

3.2.2 Microwave sensor

There is one microwave sensor located on the mainline at KM 18+000 and other 12 sensors located at the on-ramps and off-ramps.

Fig. 4 The example of microwave sensor installation

3.3 Data cleaning and processing

Based on the traffic volume dataset obtained from the motorway number 7, there was supposed to be a total of 581,040 records. However, there are nearly 467,848 records missing from the dataset (80%). Therefore, data cleaning and pre-processing are important steps before using the data for the model development.

3.3.1 Traffic volume data cleaning

Historical data were utilized to establish a threshold for detecting and removing outliers in the dataset where a threshold was established for each hour, and the data were cleaned accordingly within that hour. Any traffic volume falling below the lowerbound (5th percentile) or exceeding the upperbound (95th percentile) was identified as an outlier and replaced with a NaN value before being used in the data imputation process.

To clean the outliers of the current week, historical data from the preceding 12 weeks were employed. For instance, data from week 44 (the first week of the dataset) utilized a threshold derived from the data between weeks 32 and 43.

3.3.2 Traffic volume data imputation

Due to issues with data delay and missing data, some NaN values were present in the dataset. To prepare the data for use in the model, these values were corrected by averaging the values of six timeslots. If the number of NaN values in the 6 timeslots exceeded 4, the result was set to NaN; otherwise, the average of the available values was computed, ignoring the NaN values.

3.3.3 Relevant data inputs

The model inputs were not limited to traffic volume from sensors alone. Other important variables, such as holidays, time of day, day of the week, and the indicator for unfixable data, were also considered.

3.3.4 Dependent variables

According to the location of our model (W03), the traffic volume (veh/5 min) was calculated using the image processing and the nearby on-off ramp sensors, using the following calculation.

$$VOL_{W03} = MAIN_{10101} - EXT_{70102} - EXT_{70103} + ENT_{70104} + ENT_{70207}$$

Where $MAIN_{10101}$ is number of the image processing and EXT_{70102}, EXT_{70103}, ENT_{70104} and ENT_{70207} are the number of sensors nearby as shown in Fig. 2

4. Model development

4.1 LSTM architecture for traffic volume forecasting

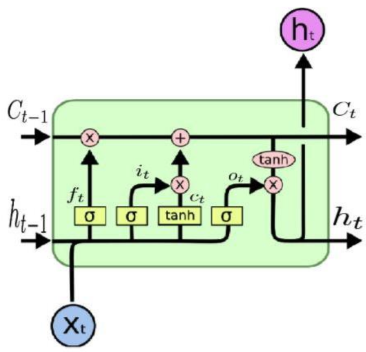

Fig. 5 The LSTM architecture for traffic volume forecasting [11]

From Fig. 5, X denotes time-series data, h_t denotes hidden vector, C_t denotes cell state, σ and tanh denotes activation functions, f_t denotes forget gate, i_t denotes input gate and o_t denotes output gate.

Let the input time-series data in the sliding window be $X=\{x_1,x_2,...,x_T\}$. The hidden vector sequence $H=\{h_1,h_2,...,h_T\}$ is then calculated. Next, an output sequence $Y=\{y_1,y_2,...,y_T\}$ is given by the LSTM network sequence. Eq. (1) and Eq. (2) are then iterated:

$$h_t = H(W_h \cdot [h_{t-1}, x_t] + b_h) \tag{1}$$

$$y_t = W_y h_t + b_y \tag{2}$$

Here W is a weight matrix (e.g., W_h is the input-hidden weight matrix), and b is a bias vector. $H(\cdot)$ is a hidden layer function and is computed by iterating Eq. (3)–(8).

Gates:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_c) \tag{5}$$

Input transform:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{6}$$

Memory update:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{8}$$

Here $\sigma(\cdot)$ is a sigmoid function; $\sigma(\cdot)$ and $\tanh(\cdot)$ are defined in Eq. (1) and Eq. (2), respectively. f is a forget gate, i is an input gate, o is an output gate, and c is a cell update gate.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{9}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

Eq. (1) and Eq. (2) together define the activate function.

4.2 Training and test datasets

4.2.1 Model data inputs

During the training process, the input layer used 75% of the overall data as the training dataset, while the remaining 25% was utilized for testing.

- Training dataset (2022/10/31 - 2023/01/22)
- Test dataset (2023/01/23 - 2023/02/19)

To address the issue of non-imputed data causing NaN values in dependent variables, Y, records with NaN values in Y were eliminated. This was necessary because the dataset contained NaN variables that were used in the calculation of the Y variable.

4.2.2 Cross validation

Cross-validation for time series data is a technique used to evaluate the performance of a model when dealing with sequential data. In contrast to traditional cross-validation methods, where data is randomly divided into training and testing sets, time series cross-validation involves creating multiple training and testing sets by using a sliding window approach.

4.3 Performance evaluation metrics

Three metrics widely used to evaluate the prediction performance of the models were deployed [12]:

4.3.1 Root Mean Squared Error (RMSE):

Root Mean Squared Error (RMSE) is a statistical metric that measures the average magnitude of the differences between predicted and actual values in a dataset. It is often used in regression analysis and machine learning to evaluate the accuracy of models.

RMSE calculates the square root of the average of the squared differences between the predicted and actual values. A lower RMSE value indicates a better fit of the model to the data. The calculation of RMSE is as shown below:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - f(x_i)\right)^2}$$

4.3.2 Mean Absolute Error (MAE):

MAE (Mean Absolute Error) is a statistical metric that measures the average absolute difference between predicted and actual values in a dataset. It is often used in machine learning and statistical modeling to evaluate the accuracy of regression models, and it is also a useful metric for assessing the performance of forecasting models.

MAE calculates the absolute difference between the predicted and actual values, and then takes the average of these absolute differences to provide an overall measure of error. A lower MAE value indicates a better fit of the model to the data. The calculation of MAE is as shown below:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - f(x_i)\right|$$

4.3.3 Mean absolute percentage error (MAPE):

Mean absolute percentage error (MAPE) is a statistical metric that measures the average absolute percentage difference between predicted and actual values in a dataset. It is often used in forecasting models and is particularly useful when dealing with data of varying magnitudes or scales.

MAPE calculates the absolute percentage difference between the predicted and actual values, and then takes the average of these absolute percentage differences to provide an overall measure of error. A lower MAPE value indicates a better fit of the model to the data. The calculation of MAPE is as shown below:

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\frac{\left|y_i - f(x_i)\right|}{y_i}$$

where y_i and $f(x_i)$ represent the real traffic volume information and predicted traffic volume. N is the number of the total real traffic volume information.

4.4 Model Hyperparameter Selection

The LSTM model's hyperparameters primarily consist of the following: the learning rate, batch size, training epochs, and number of units. During the experiment, manual adjustments were made to set the following hyperparameters: learning rate to 0.001, batch size to 16, number of units to double the shape of the training data, and training epochs to the number of epochs that resulted in the lowest MAE during cross-validation.

5. Results

Based on the data, approximately 80% of the data are missing, rendering it impractical to execute the model. To address this issue, a waiting time of 30 minutes before running the model is implemented, in addition to utilizing traffic volume imputation to correct the data. The data preparation process is shown in Fig. 6.

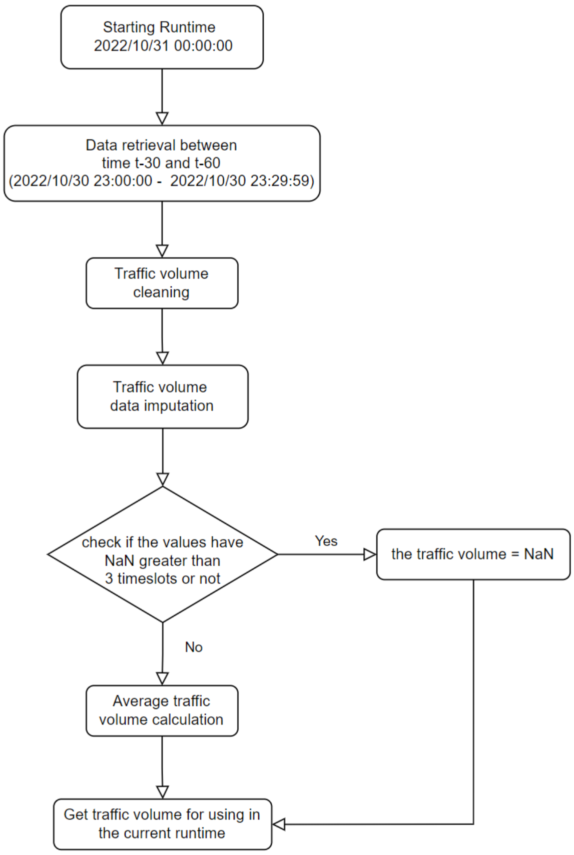

Fig. 6 The overview of traffic volume preparation

As a result, the missing data is reduced to around 10%, and the model's performance improves, achieving the following results.

Table 1 The traffic prediction results.

Model	MAPE (%)	MAE (Veh)	RMSE (Veh)
W03	9.26	17.31	26.67

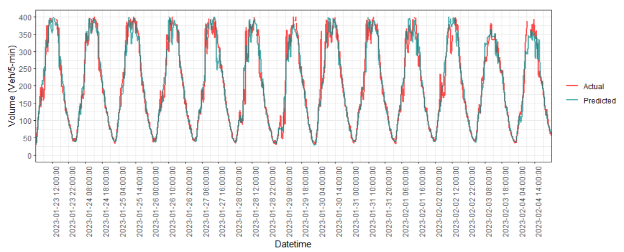

Fig. 7 The visualization results for traffic prediction of 5 minutes

6. Sensitivity Analysis

In this section, the sensitivity of the delayed data on the model performance was analyzed. Four delayed data scenarios were simulated including 15%, 25%, 35%, and 45% of delayed data. The baseline is 10% delayed data and random sampling method was used to produce various datasets.

The data were processed by introducing random missing data, with a distribution of 80% of the data being correctable via traffic imputation, and the remaining 20% of the data being unfixable. The randomization process was kept consistent by utilizing a seed number of 1.

Fig. 8 displays the results of the four scenarios with delayed data ranging from 15% to 45%. The findings indicate that when the percentage of delayed data is 17.45%, the MAPE is 20%. As the percentage of delayed data increases to 25%, the MAPE also increases to around 30%. When the delay reaches 50% in the third scenario, the MAPE deteriorates further. Finally, in the fourth scenario with 45% delayed data, the model's error rate is almost 100%.

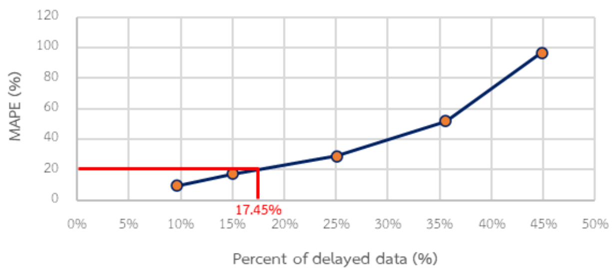

Fig. 9 Impact of delayed data sensitivity on model performance

7. Discussion and Conclusion

7.1 Model performances

The model's performance was assessed using three metrics: MAPE (Mean Absolute Percentage Error) with a result of 9.26%; MAE (Mean Absolute Error) with a result of 17.31 vehicles, and RMSE (Root Mean Square Error) with a result of 26.67 vehicles. The results indicate that the application of traffic imputation and a waiting period of 30 minutes were successful in reducing the missing data from 80% to 10%. These findings suggest that this approach is highly effective in enhancing the model's performance. However, the implementation of a waiting time of 30 minutes may potentially compromise the model's responsiveness to real-time traffic. For example, if data are gathered on a day when traffic conditions rapidly change, the model will solely rely on information from the preceding 30 minutes, leading to inaccurate predictions.

7.2 Sensitivity to data uncertainty

Based on the results illustrated in Fig. 9. It can be concluded that the model's performance is not significantly affected by delays, as it maintains a high level of accuracy even when faced with a delay impact of 17.45%. This suggests that the model can reliably and accurately process data even in situations where there may be delays in information transmission or processing. These findings have important implications for real-world applications, where delays can be a common occurrence, and highlight the potential of this model to be used in a range of contexts where reliability and accuracy are crucial.

7.3 Future research

1) According to the application of a waiting time of 30 minutes, it represents a provisional measure. As a more sustainable solution, the issue could be addressed at the sensor level, or an examination could be made into whether forecasting every half hour is the most feasible option. Nevertheless, it is imperative to consider that forecasting too frequently may not be optimal due to potential limitations in the frequency of data updates.

2) In terms of model advancement, the current mean absolute percentage error (MAPE) value of 9.26% results from training the data during high holiday seasons. Nonetheless, this approach may have an impact on the accuracy of traffic volume predictions, as traffic patterns vary between weekdays and holidays. Presently, the model performs well for weekday predictions, but tends to underperform during high holiday seasons. To address this issue, a potential solution would be to train the model with data spanning a broader time range, which would enable the model to acquire more comprehensive knowledge. Furthermore, the significance of each sensor may have varying effects on the model's performance, with mainline sensors having more weight than other on-off ramp sensors.

3) During the sensitivity test where delayed data were introduced randomly, it was observed that certain sensors, such as mainline and on-off ramp sensors, had different levels of significance. To enhance accuracy, multiple runs should be conducted to obtain an average result. Moreover, in real-world settings where delays and disruptions are common occurrences, it may be necessary to consider alternative strategies for mitigating the impact of delayed data on the model's performance. One such strategy could be to develop more robust data imputation techniques that can handle a larger percentage of unfixable missing data. Another strategy could be to combine the proposed approach with other forecasting models that are less sensitive to delayed data.

Acknowledgement

Data used in this project were provided by the Department of Highways, Thailand.

References

[1] Tak, S., Woo, S., & Yeo, H. (2016). Data-driven imputation method for traffic data in sectional units of road links. *IEEE Transactions on Intelligent Transportation Systems*, *17*(6), 1762–1771.

[2] Chang, G., & Ge, T. (2011). Comparison of missing data imputation methods for traffic flow. *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, 639–642.

[3] Li, D., Li, L., Li, X., Ke, Z., & Hu, Q. (2020). Smoothed LSTM-AE: A spatio-temporal deep model for multiple time-series missing imputation. *Neurocomputing*, *411*, 351–363.

[4] Tian, Y., Zhang, K., Li, J., Lin, X., & Yang, B. (2018). LSTM-based traffic flow prediction with missing data. *Neurocomputing*, *318*, 297–305.

[5] Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2020). Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, *118*, 102674.

[6] Li, R., Hu, Y., & Liang, Q. (2020). T2F-LSTM method for long-term traffic volume prediction. *IEEE Transactions on Fuzzy Systems*, *28*(12), 3256–3264.

[7] Luo, X., Li, D., Yang, Y., & Zhang, S. (2019). Spatiotemporal traffic flow prediction with KNN and LSTM. *Journal of Advanced Transportation*, *2019*.

[8] Kang, D., Lv, Y., & Chen, Y. (2017). Short-term traffic flow prediction with LSTM recurrent neural network. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 1–6.

[9] Nguyen, G.-T., Nguyen, V.-Q., Nguyen, H.-D., & Kim, K. (2018). Lstm based network traffic volume prediction. *Proceedings of the Korea Information Processing Society Conference*, 362–364.

[10] Khan, Z., Khan, S. M., Dey, K., & Chowdhury, M. (2019). Development and evaluation of recurrent neural network-based models for hourly traffic volume and annual average daily traffic prediction. *Transportation*

Research Record, *2673*(7), 489–503.

[11] Liu, Y., Wang, Y., Yang, X., & Zhang, L. (2017). Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 1–8.

[12] Bai, J., Zhu, J., Song, Y., Zhao, L., Hou, Z., Du, R., & Li, H. (2021). A3t-gcn: Attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, *10*(7), 485.