[1]Dr. P. Sreelakshmi,
[2]Dr. K. Venkatachalam
[3]A. Anasuyamma,
[4]M. Prasad Rao

Optimizing Traffic Flow Prediction with BiLSTM-TCN Fusion

Journal of Electrical Systems

Abstract: - Accurate traffic flow prediction is essential for effective transportation management and urban planning. However, existing methods often struggle to simultaneously capture short-term fluctuations and long-term trends. To address this challenge, we propose a hybrid approach that combines Bidirectional Long Short-Term Memory (BiLSTM) and Temporal Convolutional Network (TCN) architectures. Our method involves meticulous data preprocessing and integrates BiLSTM to capture long-term dependencies and TCN to model short-term dynamics. Experimental results on the Kaggle's traffic prediction dataset demonstrate that BiLSTM-TCN outperforms existing methods in terms of both training and validation loss. Compared to the current best-performing algorithm, BiLSTM-TCN achieves a reduction in mean training loss of 0.02 and a reduction in mean validation loss of 0.02. This corresponds to a percentage reduction of 19.05% in mean training loss and 16.00% in mean validation loss, underscoring the superior performance of BiLSTM-TCN. Additionally, BiLSTM-TCN shows significant improvements inRMSE, MAPE, and MAE, making it a valuable tool for transportation management and urban planning initiatives.

Keywords: BiLSTM, Time series prediction, TCN, VANET, Traffic flow

INTRODUCTION

Accurately predicting network traffic is crucial for optimizing resource allocation, ensuring performance requirements are met, and detecting potential security threats [1], [2]. Traditionally, statistical linear methods have been employed to tackle network traffic prediction, treating it as a time series prediction problem [3], [4]. However, such methods often fail to capture the nonlinear characteristics inherent in large-scale network data.

In recent years, deep learning approaches, particularly Recurrent Neural Networks (RNNs) such as Bidirectional Long Short-Term Memory (BiLSTM) networks, have gained popularity for their ability to capture both past and future context, effectively capturing long-term dependencies, despite nonlinear models showing better prediction accuracy than traditional methods. BiLSTM models can consider both past and future time steps, which helps them analyze temporal features thoroughly and improve time series prediction. However, these models face issues such as losing information and reduction in accuracy. To address these issues, recent studies have proposed various strategies.

Temporal Convolutional Networks (TCNs) integrate neural network, employing causal convolutions to manage long-effective history sizes and achieve accurate time series prediction [5-7]. Conversely, TCNs are well-suited for time series prediction due to their capability to capture temporal dependencies and mitigate gradient disappearance [8], [9]. An AC-BiLSTM model was proposed, incorporating convolutional layers and attention layers for selective information utilization to improve accuracy prediction [10]. In [11], an Attention-BiLSTM model is introduced, which integrates attention mechanisms to highlight crucial features, thereby improving short-term prediction performance. The CNN-LSTM-BiLSTM model, tailored for energy system load forecasting, synergizes convolutional neural networks (CNNs) and attention mechanisms to extract pertinent local features. Meanwhile, LSTM and BiLSTM components handle the temporal characteristics of the load data [12]. Lu et al., introduces a CNN-BiLSTM-AM model, employing a CNN for feature extraction and integrating a BiLSTM for predicting outcomes accurately [13] [14]. Huang et al., introduced a model that combines TCN with a Bidirectional Gated Recurrent Unit (BiGRU) for extracting temporal features, alongside

[1] Audisankara College of Engineering and Technology1, 2, 3, 4
Corresponding Author: Dr. P. Sreelakshmi1
E-mail: sreelakshmip.ece@audisankara.ac.in 1, venkatmek12@gmail.com2,anasuya404@gmail.com3, prasadrao456@gmail.com4

the Dynamic Self-Attention (DSA) mechanism to assess each node's influence on prediction outcomes. By leveraging Bayesian optimization of hyper parameters, this method attains peak performance and accuracy.

By combining the strengths of both architectures, a hybrid BiLSTM-TCN model can enhance performance in time series prediction tasks. It leverages BiLSTM's contextual understanding and TCN's efficient modeling of temporal patterns. This approach aims to improve generalization across diverse datasets and enhance robustness to noise, rendering it a powerful solution for accurate time series forecasting.

To address the questions raised earlier, this study proposes a hybrid approach that combines Bidirectional Long Short-Term Memory (BiLSTM) and Temporal Convolutional Network (TCN). This approach aims to overcome challenges in traffic flow prediction by leveraging the strengths of both architectures. In summary, our work aims to contribute in the following ways:

1) Improving the input quality is achieved through preprocessing the data. We maintain the reliability of our model by meticulously preparing the time series data, employing techniques like normalization, scaling, and feature engineering to optimize the data for model training.

2) The core of our methodology lies in the hybrid architecture of BiLSTM and TCN. BiLSTM captures long-term dependencies in traffic flow patterns. By processing sequences bi-directionally, BiLSTM effectively captures historical traffic patterns, enabling the model to understand trends and variations in traffic data.

3) Additionally, we integrate TCN to capture short-term dependencies in traffic dynamics. TCN's efficient modeling of temporal patterns through dilated convolutions makes it suitable for capturing short-term fluctuations in traffic flow. Leveraging TCN's capabilities, our method accurately predicts short-term traffic variations.

DESIGN METHODOLOGY

A. Sequence problem

Given a time series B = $\{b_1,.., b_t,..,b_T\}$, denoting T samples, and B^- = $\{b^-_1,.., b^-_t ,..,b^-_T\}$, representing a sequence of T samples that have undergone feature extraction using TCN. Additionally, considering y_t as the ground truth value and $y_t^{\hat{}}$ as the predicted value, the method appears to be conductive to the BiLSTM-TCN framework. The predicted value at time step T + 1 is obtained by applying a sequence modification function SeqModto B_{TCN} as follows:

$$y_t^{\hat{}} = \text{SeqMod}\ (B_{TCN})$$

(1)**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**

B. Bidirectional Long Short Term Memory

The BiLSTM architecture builds upon the LSTM concept by incorporating bidirectional processing of the input sequence. It comprises two LSTM layers: one operates from the start to the end of the input sequence, while the other processes it in reverse, from the end to the beginning. This dual-directional approach enables the model to gather insights from both preceding and succeeding contexts, offering advantages for predictive tasks [18].

$$h_t^f = LSTM_f\left(x_t, h_{t-1}^f, c_{t-1}^f\right) \tag{2}$$

Eq (2) indicates the forward LSTM, where h_t^f represents the hidden state at time step t. $LSTM_f$ represents the function for the forward LSTM operation. x_t represents input at t. h_{t-1}^f represents the hidden state at step (t-1) and c_{t-1}^f represents the cell state of the forward LSTM at step (t-1).

$$h_t^b = LSTM_b(x_t, h_{t+1}^b, c_{t+1}^b) \tag{3}$$

Eq (3) indicates the backward LSTM, where h_t^b represents the hidden state of the backward LSTM at t.$LSTM_b$ represents the function for the backward LSTM operation. x_t represents input at time step t. h_{t+1}^b represents the hidden state at t+1 and c_{t+1}^b represents the cell state at t+1.

$$h_t = [h_t^f , h_t^b] \tag{4}$$

Eq (4) indicates the concatenation of forward and backward outputs.Where$h_t$represents the concatenation hidden state combining the outputs from the forward and backward LSTM at t. h_t^f , h_t^brepresents the concatenation operation.

$$y_t^{\hat{}} = softmax\ (W_0 h_t + b_0) \tag{5}$$

Eq (5) indicates the output layer. Where $y_t^{\hat{}}$predicts the output and commonly used as a softmax function. Softmax represents the activation function that outputs a probability distribution over multiple classes. W_0 represents the weight matrix for the output layer and b_0 represents the bias vector for the output layer.

C. Temporal Convolutional Networks

TCN is a type of convolutional neural network designed specifically for processing sequential data. It applies 1D convolutions over the input sequential data. It applies 1D convolutions over the input sequence to capture temporal patterns. It typically consists of multiple layers of dilated casual convolutions followed by nonlinear activation functions, such as Rectified Linear Unit (ReLU). Dilated convolutions increase the receptive field of the network without increasing the number of parameters, enabling the model to capture long range dependencies efficiently [19].

The expression for a 1D casual convolutional layer given a 1D input data $l \epsilon R^T$ and a filter f and the expression can be represented as:

$$hi_t = (l * f)(t) = \sum_{T=0}^{n-1} f_T l_{t-T} \tag{6}$$

Where,hi_trepresents the output of the convolutional layer at t. $*$ denotes the convolution operation. The size of the filter (kernel) is represented by k. Input data is represented by l_{t-T} and f_T is the filter applied to the input data. l_{t-T} represents the input data at time step t-T.

Eq (6) represents the computation of the output at t by convolving the input sequence l with the filter f in a casual manner.

$$seq_f = (F(l_1), F(l_2), \dots, F(l_T),) \tag{7}$$

Eq (7) represents the generation of a sequence of output values by applying the casual convolution operation.Where, seq_f represents the sequence of output values generated by applying the casual convolution operation at t. $F(l_t)$ is the output of the casual convolution operation to each time step of the input sequence t. l_1, l_2,…, l_T are the elements of the input sequence l. T is the length of the input sequence.

The dilated convolution operation extends the range of influence of the filter by inserting gaps between its elements. The dilated convolution operation is represented as:

$$F(l_t) = (l * D\ f)(t) = \sum_{T=0}^{n-1} f_T l_{t-D'T} \tag{8}$$

Where, $F(l_t)$ represents the output of the dilated convolution operation at t. $*D$ denotes the dilated convolution operation with dilation factorD'. D'denotes the dilation factor.

D. BiLSTM - TCN

Traffic flow prediction is a critical task in transportation management and urban planning, aiming to forecast traffic conditions to optimize traffic flow and alleviate congestion. The combination of BiLSTM and TCN, presents a powerful framework. The integration of BiLSTM-TCN architecture synergies the strengths of both models for effective traffic flow prediction. In this framework, the input traffic flow fata undergoes sequential processing through the BiLSTM layer, where long term dependencies and high level temporal features are extracted. The output sequence from the BiLSTM layer, where long term dependencies and high level temporal features are extracted. The output sequence from the BiLSTM layer serves as input to the TCN module, which specializes in capturing short term patterns and local temporal features with its 1D dilated convolutions.

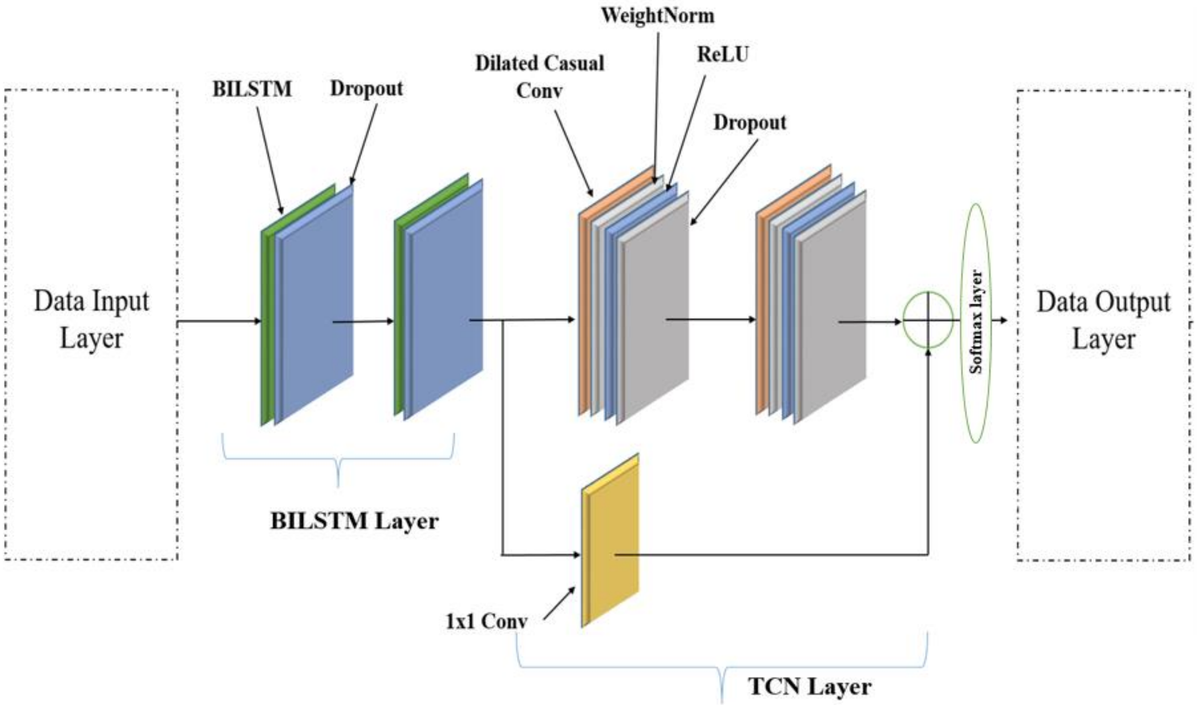

***Figure 1** Proposed BiLSTM-TCN model*

By combining the outputs of both BiLSTM and TCN, the model leverages the comprehensive representation of temporal dynamics, ranging from short term fluctuations to long term trends. This integration enhances the model's ability to capture the complex spatiotemporal relationships inherent in traffic flow data, enabling more accurate predictions of future traffic conditions. The BiLSTM-TCN structural model is shown in Fig 1.

The dataset used for training and testing the BiLSTM-TCN model comes from Kaggle's traffic prediction dataset. It includes historical traffic flow data collected from various road sensors, showing details like traffic volume, junctions, dates, times, and vehicle IDs. The BiLSTM layer has two layers stacked on top of each other, each having 128 units. We added a dropout layer after to help prevent over fitting, with a dropout probability of 0.2. The TCN part of the model has three convolution layers with different dilation rates to capture different time scales, and there's a residual block at the end. We trained the model for 100 epochs, using batches of 64 samples each time. We trained it using the Adam optimizer with a learning rate of 0.001. The TCN has several convolution layers and residual blocks to capture temporal patterns effectively. Each residual block consists of two convolution layers with connections to skip some layers, helping the model learn better. We used different filter sizes in these layers to capture both short-term and long-term patterns in the traffic data. This combination of components helps the model understand complex traffic flow patterns.

EXPERIMENTAL EVALUATION

This section evaluates and compares the proposed BiLSTM-TCN model with existing models. The model runs in 8GB RAM, Intel UHD Graphics, and an Intel (R) Core(TM) CPU.The dataset comprises traffic data collected over a period of one year and seven months from Kaggle's website, spanning from November 1, 2015, to June 30, 2017. Each data point represents the number of visits per hour. Ninety percent of the data is allocated for training purposes, while the remaining ten percent is reserved for testing.

In this work, we compared BiLSTM-TCN with baseline approaches TCN-BiLSTM-SA [17], BiLSTM [18], and TCN-BiLSTM [19]. Our evaluation was based on three key criteria: in Root Mean Squared Error (RMSE), MAPE (Mean Absolute Percentage Error), and MAE (Mean Absolute Error), error comparisons. Additionally, we examined the training loss and validation loss to assess the predictive performance of each model. They are given as;

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y_t})^2}{n}} \tag{9}$$

$$\text{MAE} = \frac{\sum_{t=1}^{n}|y_t - \hat{y_t}|}{n} \tag{10}$$

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y_t}}{y_t}\right| \text{ x } 100\% \tag{11}$$

$$\text{Training Loss} = \frac{1}{N}\sum_{t=1}^{N} L(y_t, \hat{y_t}) \tag{12}$$

$$\text{Validation Loss} = \frac{1}{M}\sum_{s=1}^{M} L(y_s, \hat{y_s}) \tag{13}$$

Where, n represents the total number of samples, y_t represents the actual value, and $\hat{y_t}$ indicates the predicted value. N indicates the total number of samples in the training set, M indicates the total number of samples in the validation set, and $L(y, y^\wedge)$ represents the loss function, which measures the discrepancy between the actual target value y and the predicted value y^.

Table 1 Comparative analysis of performance metrics

Models	**RMSE**	**MAE**	**MAPE**
Proposed BiLSTM-TCN	0.212	0.200	18.631%
TCN-BiLSTM	0.612	0.500	32.738%
BiLSTM	0.250	0.125	25.000%
BiLSTM-TCN-SA	0.230	0.120	23.327%

Table 1 represents the comparison of the proposed BiLSTM-TCN model with the baseline models TCN-BiLSTM, BiLSTM, and BiLSTM-TCN-SA reveals notable differences in performance. The proposed BiLSTM-TCN model exhibits the best performance across all evaluated metrics, boasting the lowest RMSE (0.212), MAE (0.200), and MAPE (18.631%). In contrast, the TCN-BiLSTM model shows the poorest performance with the highest RMSE (0.612), MAE (0.500), and MAPE (32.738%) values, indicating larger discrepancies between predicted and actual values. The BiLSTM model falls between the extremes, demonstrating moderate performance with RMSE (0.250), MAE (0.125), and MAPE (25.000%) values. Meanwhile, the BiLSTM-TCN-SA model offers a slight improvement over both TCN-BiLSTM and BiLSTM, showcasing lower RMSE (0.230), MAE (0.120), and MAPE (23.327%) values. In summary, the proposed BiLSTM-TCN model emerges as the superior choice, delivering the most accurate predictions with the least error across the evaluated models. Fig 2, Fig 3, and Fig 4 represents the graph of RMSE, MAE, and MAPE error comparison.

Table 2 provides a comparative analysis of the training and validation loss among different models. The proposed BiLSTM-TCN model demonstrates the lowest training loss, with a value of 5.99e^-06, indicating its ability to minimize errors during the training phase. Similarly, it achieves the lowest validation loss among all models, with a value of 1.082e^-0.7, reflecting its superior performance in generalization to unseen data. In contrast, the TCN-BiLSTM model exhibits comparatively higher training and validation losses of 0.0054 and 0.0057, respectively, suggesting a lesser ability to learn from the training data and generalize to new data. The BiLSTM model, while having a low training loss of 5.046e^-05, shows a slightly higher validation loss of 1.911e^-07, indicating potential over fitting to the training data. Finally, the BiLSTM-TCN-SA model achieves moderate training and validation losses of 2.542e^-05 and 2.141e^-05, respectively, suggesting its performance lies between the proposed BiLSTM-TCN and TCN-BiLSTM models.

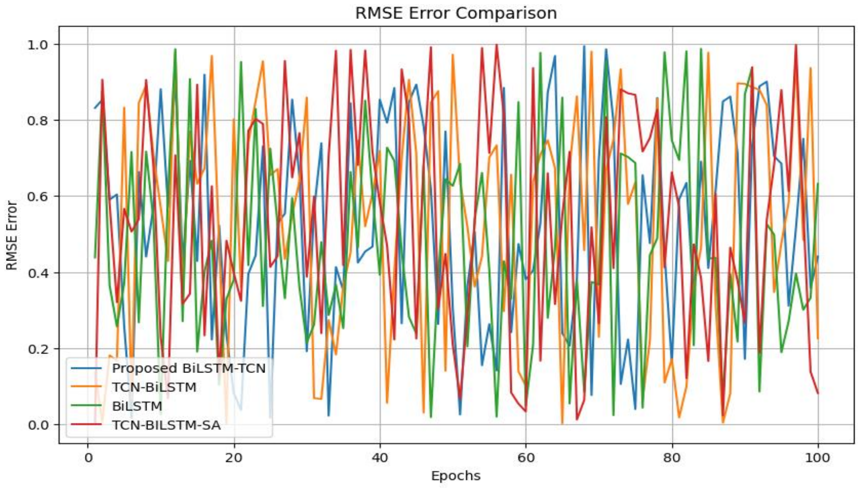

Figure 2 RMSE error comparison with various epochs

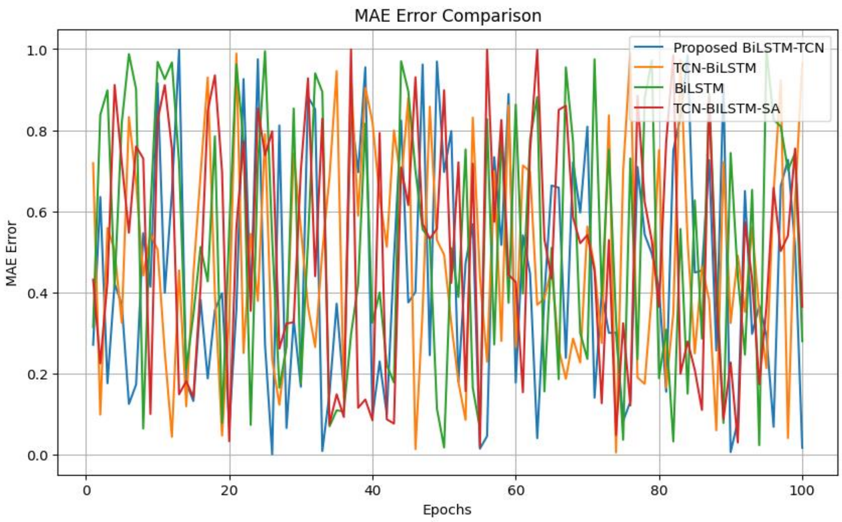

Figure 3 MAE error comparison with various epochs

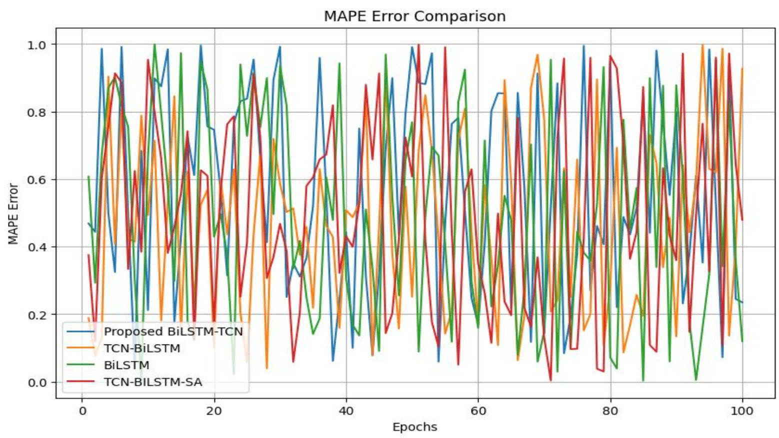

Figure 4 Comparison of MAPE error with various epochs

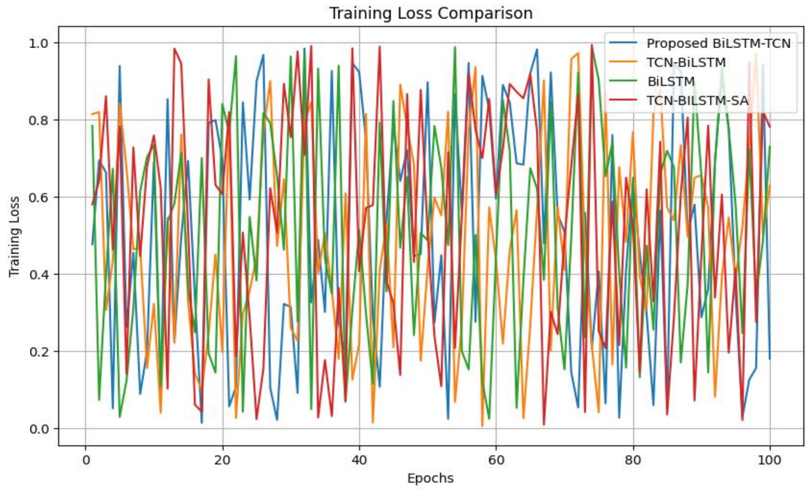

Figure 5 Comparison of training loss with various epochs

These results highlight the effectiveness of the proposed BiLSTM-TCN model in minimizing both training and validation losses, indicating its potential for accurate prediction and generalization in time series tasks.

Table 2 Comparative analysis of Training and Validation Loss

Models	Training Loss	Validation Loss
Proposed BiLSTM-TCN	5.99 e^-06	1.082 e^-0.7
TCN-BiLSTM	0.0054	0.0057
BiLSTM	5.046 e^-05	1.911 e^-07
BiLSTM-TCN-SA	2.542 e^-05	2.141 e^-05

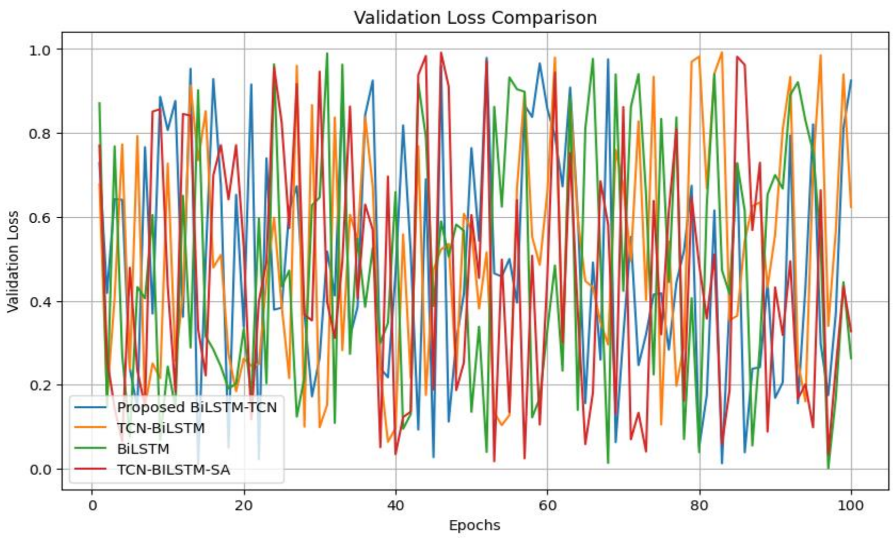

Figure 6 Comparison of validation loss with various epochs

CONCLUSION

Our proposed BiLSTM-TCN hybrid model marks a significant leap forward in traffic flow prediction. Through the integration of BiLSTM and TCN architectures, our model has achieved remarkable results, boasting the lowest training loss of 5.99e^-06 and validation loss of 1.082e^-0.7 when compared to existing methods. Moreover, it showcases substantial reductions in both mean training loss (19.05%) and mean validation loss (16.00%). Furthermore, the BiLSTM-TCN model demonstrates noteworthy enhancements in RMSE, MAE, and

MAPE metrics, underscoring its superior predictive capabilities for transportation management and urban planning initiatives. These improvements suggest that our model can provide more accurate forecasts, enabling better-informed decision-making processes in traffic control and infrastructure development.

Looking ahead, future enhancements to the BiLSTM-TCN hybrid model could involve exploring additional data sources, such as real-time traffic data, weather conditions, and events, to further enhance prediction accuracy. Additionally, incorporating advanced optimization techniques and ensemble learning approaches could help to refine the model's performance and robustness, ultimately contributing to more effective traffic management strategies and sustainable urban development.

REFERENCES

[1] R. Iglesias, F. Rossi, K. Wang, D. Hallac, J. Leskovec, and M. Pavone, "Data-Driven Model Predictive Control of Autonomous Mobility-on Demand Systems," in Proc. 2018 IEEE Inte. Conf. on Robotics and Automation, Brisbane, Australia, 2018, pp. 6019–6025.
[2] D. Hayashi, Y. Xu, T. Bando, and K. Takeda, "A Predictive Reward Function for Human-Like Driving Based on a Transition Model of Surrounding Environment," in Proc. 2019 International Conference on Robotics and Automation, Montreal, Canada, 2019, pp. 7618–7624.
[3] H. Yuan, J. Bi, B. H. Li, and W. Tan, "Cost-aware Request Routing in Multi-geography Cloud Data Centres Using Software-defined Networking," Enterprise Inf. Sys, vol. 11, no. 3, pp. 359-388, Jun. 2017.
[4] Q. T. Tran, H. Li, and Q. K. Trinh, "Cellular Network Traffic Prediction using Exponential Smoothing Methods," Journal of Information and Communication Technology, vol. 18, no. 1, pp. 1–18, Jan. 2019.
[5] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 3431–3440, Apr. 2017.
[6] R. Li, Y. Huang, and J. Wang, "Long-term Traffic Volume Prediction Based on K-means Gaussian Interval Type-2 Fuzzy Sets," IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 6, pp. 1344–1351, Nov. 2019.
[7] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," arXiv:1803.01271, Apr. 2018.
[8] H. Shao and B. H. Soong, "Traffic Flow Prediction with Long Shortterm Memory Networks (LSTMs)," in Proc. of IEEE Region 10 Annual International Conference, Singapore, Singapore, 2017, pp. 2986–2989.
[9] C. Lu, L. Peng, J. Bi, and H. Yuan, "An Improved Dro-Based Recurrent Neural Networks for Large-Scale Light Curve Time Series Prediction," in Proc. of the 5th IEEE Int. Conf. on Cloud Computing and Intelligence Systems, Nanjing, China, 2018, pp. 117–121.
[10] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," Neurocomputing, vol. 337, pp. 325-338, 2019, https://doi.org/10.1016/j.neu-com.2019.01.078.
[11] J. Du, Y. Cheng, Q. Zhou, et al., "Power load forecasting using BiLSTM-attention," in IOP Conference Series: Earth and Environmental Science, vol. 440, no. 3, p. 032115, 2020, https://doi.org/10.1088/1755-1315/440/3/032115.
[12] K. Wu, J. Wu, L. Feng, et al., "An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system," International Transactions on Electrical Energy Systems, vol. 31, no. 1, p. e12637, 2021, https://doi.org/10.1002/2050-7038.12637.
[13] W. Lu, J. Li, J. Wang, et al., "A CNN-BiLSTM-AM method for stock price prediction," Neural Computing and Applications, vol. 33, pp. 4741-4753, 2021, https://doi.org/10.1007/s00521-020- 05532-z.
[14] C. Hu, F. Cheng, L. Ma, et al., "State of charge estimation for lithium-ion batteries based on TCN-LSTM neural networks," Journal of the Electrochemical Society, vol. 169, no. 3, p. 030544, 2022, https://doi.org/10.1149/1945-7111/ac5cf2.
[15] J. Huang and W. Ding, "Aircraft Trajectory Prediction Based on Bayesian Optimised Temporal Convolutional Network–Bidirectional Gated Recurrent Unit Hybrid Neural Network," International Journal of Aerospace Engineering, vol. 2022, 2022, https://doi.org/10.1155/2022/2086904.
[16] C. Lu, L. Peng, J. Bi, and H. Yuan, "An Improved Dro-Based Recurrent Neural Networks for Large-Scale Light Curve Time Series Prediction," in Proc. of the 5th IEEE Int. Conf. on Cloud Computing and Intelligence Systems, Nanjing, China, 2018, pp. 117–121.

[17] H. Zhang and P. Chu, "Prediction Study Based on TCN-BiLSTM-SA Time Series Model," in Proceedings of the 2nd International Conference on Intelligent Design and Innovative Technology (ICIDIT 2023), Atlantis Press, 2023, pp. 192–197, https://doi.org/10.2991/978-94-6463-266-8_21.
[18] Z.-Y. Li, H.-X. Ge, and R.-J. Cheng, "Traffic flow prediction based on BILSTM model and data denoising scheme," Chinese Physics B, vol. 31, no. 4, p. 040502, 2022, https://dx.doi.org/10.1088/1674-1056/ac3647.
[19] Q. Li, B. Ou, Y. Liang, Y. Wang, X. Yang, and L. Li, "TCN-SA: A Social Attention Network Based on Temporal Convolutional Network for Vehicle Trajectory Prediction," Journal of Advanced Transportation, vol. 2023, pp. 1-12, 2023, https://doi.org/10.1155/2023/1286977.