AUIQ Technical Engineering Science

Manuscript 1029

Deep Learning Algorithms for Traffic Flow Predictions

Adegoke Ojeniyi

Prince Pal Singh

Ankita Vashisht

Swati Kumari

Karan Karan

AUIQ TECHNICAL ENGINEERING SCIENCE 2025;2:134–145

Scan the QR to view the full-text article on the journal website

ORIGINAL STUDY

Deep Learning Algorithms for Traffic Flow Predictions

Adegoke Ojeniyi [a], Prince Pal Singh [b,*], Ankita Vashisht [b], Swati Kumari [b], Karan Karan [b]

[a] Department of Computer Science, Mathematics and Physics Nipissing University, North Bay, P1B 8L7, Canada
[b] Data Analytics for Business, St. Clair College @Ace Acumen, Toronto, ON, M3B 3M1, Canada

ABSTRACT

Given the growing complexity of urban transportation systems, precise traffic flow forecasting is essential for reducing not only issues of congestion but also, for boosting road safety and enhancing mobility management. This study integrates Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM), Long Short-Term Memory (LSTM), and Recurrent Neural Networks (RNN) to present a hybrid deep learning framework for traffic prediction. Of these, the CNN-LSTM model is a reliable option for real-time traffic forecasting since it successfully captures both spatial and temporal dependencies, resulting in superior predictive performance. The dataset used to assess the framework includes 48,120 records from a traffic monitoring system that include hourly vehicle counts at several intersections. With an average of 22.79 vehicles per hour, a variance of 430.57, and a standard deviation of 20.75, statistical analysis shows that traffic fluctuates significantly. Based on experimental results, CNN-LSTM achieves a competitive Mean sqd Error (MSE) of 0.0095, a precision of 0.73, and a recall of 0.74, outperforming LSTM and RNN in high-traffic situations. This study demonstrates the potential of hybrid models—in particular, CNN-LSTM—in striking a balance between computational efficiency and predictive accuracy. Future research should incorporate GPS feeds and real-time data from IoT sensors to improve model adaptability and offer a scalable and clever urban traffic management solution.

Keywords: Computer vision, Traffic flow, Machine learning, RNN, LSTM, Hybrid CNN-LSTM, Hybrid approach, Hybrid model

1. Introduction

1.1. Research background

Urbanization has led to significant increases in traffic congestion, adversely impacting mobility, fuel consumption, and air quality in cities around the world [1, 2]. Efficient traffic flow prediction is essential for addressing these challenges, as it enables proactive traffic management, reduces delays, and enhances commuter safety [3]. Traditional approaches, such as the Autoregressive Integrated Moving Average (ARIMA) model [4] and Kalman filtering [5], were among the early techniques used for traffic flow prediction. However, these methods primarily rely on linear assumptions, which fail to capture the complex, dynamic, and nonlinear nature of urban traffic, particularly during peak hours or in the event of unexpected incidents [6].

In recent years, machine learning (ML) methods have revolutionized traffic flow prediction by leveraging vast amounts of traffic data to uncover intricate patterns in time-series data. Recurrent Neural Networks (RNNs), for example, can capture short-term temporal dependencies but struggle to model long-term trends due to issues like the vanishing gradient problem [7]. Long Short-Term Memory (LSTM) networks, which introduce gating mechanisms, address

Received 22 November 2024; revised 14 April 2025; accepted 21 April 2025.
Available online 30 April 2025

* Corresponding author.
E-mail addresses: adegokeo@nipissingu.ca (A. Ojeniyi), prince.thind1666@gmail.com (P. P. Singh), ankuvashisht.4@gmail.com (A. Vashisht), kumariswaticanada@gmail.com (S. Kumari), karanb203@gmail.com (K. Karan).

https://doi.org/10.70645/3078-3437.1029

these limitations by effectively retaining long-term dependencies, making them particularly useful for sequential data tasks in traffic forecasting [8, 9].

The advent of hybrid models, such as CNN-LSTM, has further improved traffic flow prediction accuracy. These models combine the Convolutional Neural Network (CNN) component to capture spatial features like localized congestion patterns and the LSTM layers to model how these patterns evolve over time [10, 11]. The hybrid CNN-LSTM approach allows for more accurate predictions, even in the face of dynamic road and weather conditions, contributing to better traffic signal optimization and congestion management [12].

Subsequent investigation by Zhao and Al-Dala'in [13] discovered that CNNs have great competence of detecting patterns (such as vector intensity and color) to extract a spatial feature of data including images. LSTMs are better at capturing temporal relationships in sequential data sets compared to CNNs. The hybrid CNN-LSTM model takes the ability of CNNs to recognize spatial representations of sequences and LSTMs to highlight temporal relationships, allowing it to perform on a complicated data set with known spatial and temporal domains. The hybrid CNN-LSTM model has been found to perform better than the stand-alone CNN and LSTM models in a comparable study identifying the Amazigh language, because of the increased accuracy achieved when processing the spectrogram features, as stated by Telmem et al. [14].

Moreover, it was determined by Maurya, Arora, and Singh's paper [15] that the hybrid architecture performs well in situations involving data that is sophisticated, diverse, and real. The hybrid model provided superior results compared to baseline models in Human Activity Recognition (HAR). For example, the model successfully classified human behaviors in sensor data, achieving an F1 Score of 95–84 percent and recognition accuracy of 98–67 percent. These arguments have established that the hybrid CNN-LSTM model is an attractive possibility in machine learning due to the combination of temporal and spatial feature extraction and its proven efficacy in a diverse array of applications. Also, the hybrid model's flexibility and robustness are assured to work in both research and applied contexts. Thus, these arguments formed the rationales for implementing the hybrid model in this current study.

1.2. Research gap

While machine learning models have made substantial progress, many existing methods still struggle with the real-time integration of traffic data, especially in dynamic environments influenced by factors such as weather, holidays, or road incidents. Moreover, comparative studies that evaluate the performance of hybrid models like CNN-LSTM against simpler time-series models such as RNN and LSTM remain scarce [16, 17]. This study seeks to bridge these gaps by evaluating the performance of RNN, LSTM, and CNN-LSTM models using a real-world traffic dataset that includes vehicle counts from multiple junctions.

1.3. Objectives

The primary objectives of this study are:

- To evaluate and compare the performance of RNN, LSTM, and CNN-LSTM models for traffic flow prediction using a real-world dataset.
- To preprocess and analyze a dataset containing hourly vehicle counts from multiple junctions, ensuring the reliability and consistency of the models.
- To demonstrate the applicability of the CNN-LSTM hybrid model for real-time traffic flow prediction and dynamic traffic management systems.

1.4. Practical relevance for stakeholders

The findings of this research will provide valuable insights for urban planners, transportation authorities, and businesses. By accurately predicting traffic flow in real-time, traffic signal timings can be optimized, potentially reducing congestion by as much as 15% during peak hours. Additionally, incorporating external factors such as weather conditions, public holidays, and road incidents into the traffic flow models will improve their adaptability to real-world scenarios, offering more precise and timely solutions for urban traffic management [18].

2. Literature review

This section presents a review of the Literature Review based on the study. It is subdivided into 2.1 which is the subsection that covers the Recurrent Neural Networks (RNN) while both 2.2 and 2.3 cover Long Short-Term Memory Networks (LSTM) and the CNN-LSTM Hybrid Model.

2.1. Recurrent neural networks (RNN)

RNNs have been foundational in modeling sequential data and are widely applied in traffic prediction for capturing short-term dependencies (Fig. 1). However, their ability to model long-term traffic patterns is constrained by the vanishing gradient problem

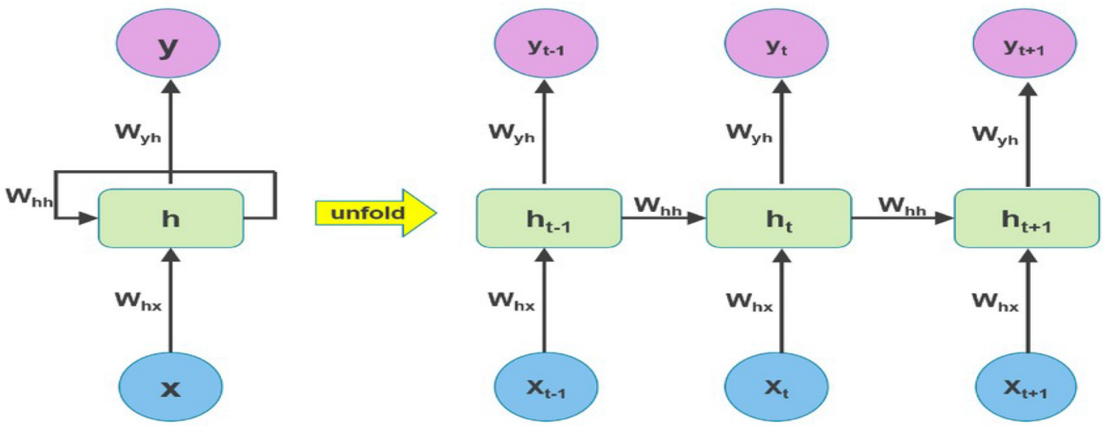

Fig. 1. RNN memory cell.

[19], which limits their use in long-term predictions. Recent improvements like attention mechanisms [20] have enhanced RNNs for specific short-term traffic forecasting tasks, but for long-term traffic forecasting, models like LSTM or hybrid models (CNN-LSTM) have shown superior performance.

Equation for Hidden State Update:

$$h_t \ = f\left(W \cdot h_t - 1 + U \cdot x_t \ + b\right)$$

Here:, h_t is the hidden state at time t, W and U are weight matrices for the previous hidden state and current input, respectively, x_t is the input at time t, b is the bias, and f is an activation function, often ReLU or tan h.

2.2. Long short-term memory networks (LSTM)

LSTMs address RNN limitations, enabling the model to capture long-term dependencies essential for traffic prediction, such as recurring rush hour patterns or seasonal fluctuations [21] (Fig. 2). The ability to handle external factors, such as weather and road incidents, further enhances the model's robustness [22, 23]. However, LSTM models can be computationally intensive, especially for large datasets [24], and require optimization techniques such as hyperparameter tuning to improve efficiency. By controlling the information flow, the LSTM's gates enable it to remember or forget information as needed.

The Forget Gate equation is given as:

$$\mathrm{f}_t = \sigma\left(W_f \cdot \left[h_{(t-1)}, x_t\right] + b_f\right)$$

Based on the previous hidden state, this gate decides which portion of the cell state should be discarded $h_{(t-1)}$ and current input x_t. Moving on with, **Input Gate**:

$$i_t = \sigma\left(W_i \cdot \left[h_{(t-1)}, x_t\right] + b_i\right)$$

This gate determines what additional data should be added to the cell state, this will be, **Cell State Update**:

$$C_t = f_t * C_{(t-1)} + i_t * tanh\left(W_C \cdot \left[h_{(t-1)}, x_t\right] + b_C\right)$$

The cell state C_t is updated by this equation, which balances new and old data. Last but not least is **Output Gate**:

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

The output gate manages the next hidden state, which combines past context with the updated cell state.

2.3. CNN-LSTM hybrid model

The CNN-LSTM model integrates CNN's spatial learning ability with LSTM's temporal sequence prediction power, making it effective for complex traffic scenarios where both spatial and temporal dependencies must be considered [25] (Fig. 3). Hybrid models like CNN-LSTM have demonstrated superior accuracy in traffic prediction tasks, such as real-time traffic signal optimization and route planning [26]. However, training such models requires significant computational resources and may lead to overfitting without proper regularization [27]. Before capturing localized variations that may be indicative of congestion in particular areas or abrupt spikes in vehicle counts, CNN layers use convolutions to identify spatial patterns within traffic features. The LSTM layer processes temporal dependencies, including time-based variations in traffic volume, after flattening the CNN's output.

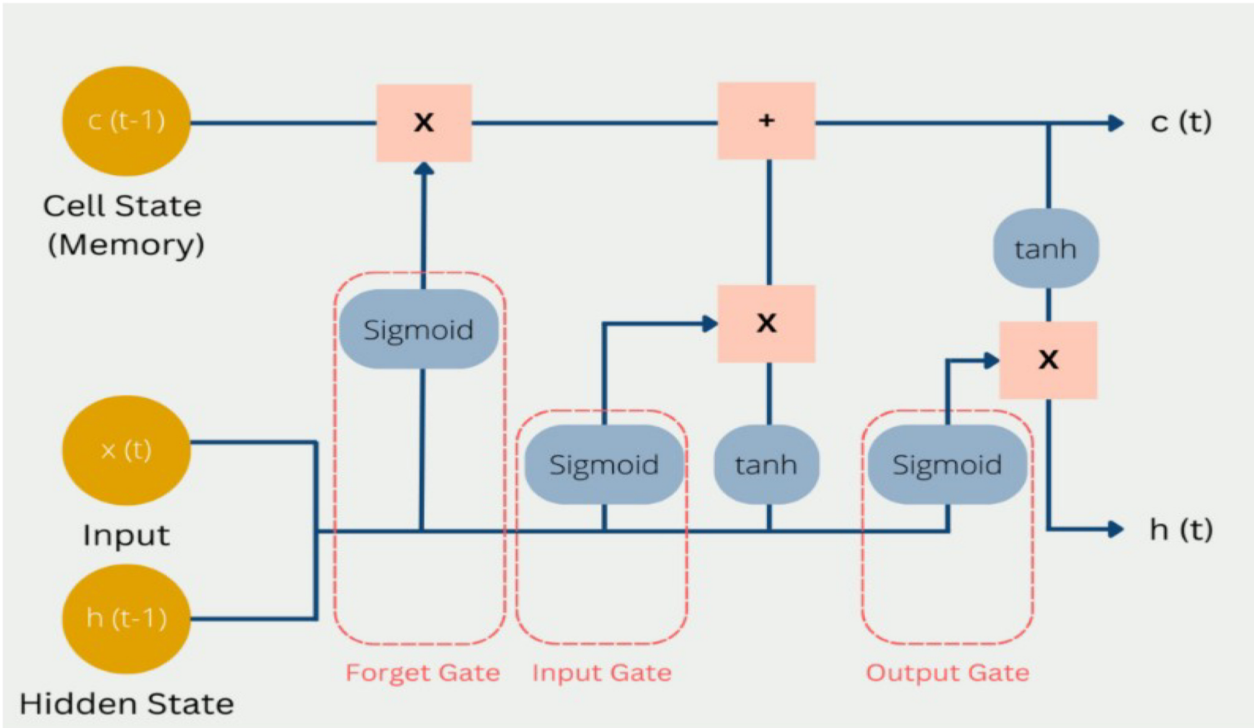

Fig. 2. Long and short-term memory cell.

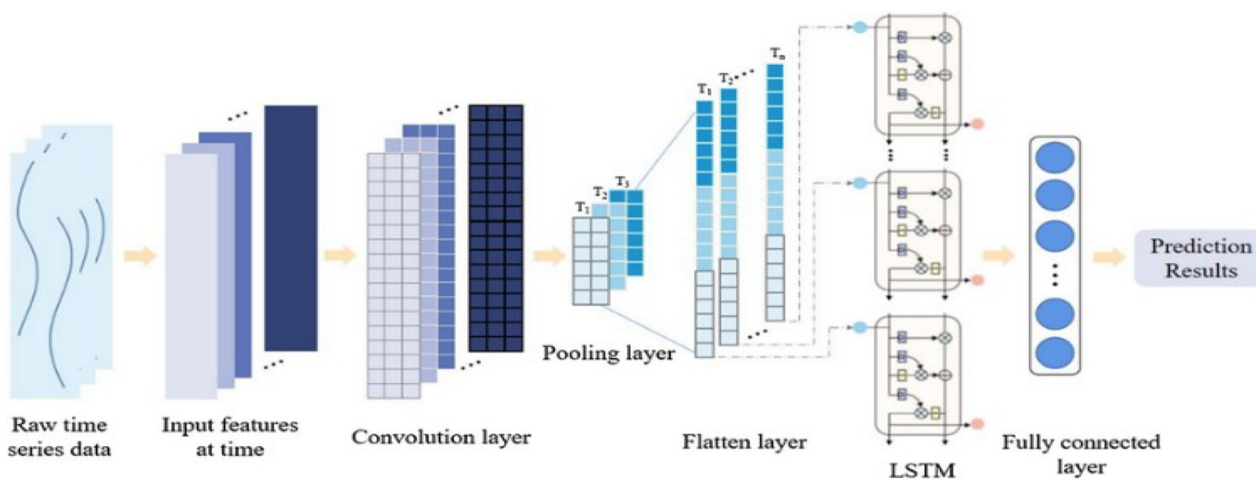

Fig. 3. CNN-LSTM memory cell.

3. Methodology

3.1. Dataset description

This study utilized a dataset consisting of 48,120 records, each representing the number of vehicles observed at various traffic junctions every hour. The dataset provides detailed information essential for predicting traffic flow across different times and locations (Table 1).

- DateTime: This attribute records the exact date and time of each observation, capturing the time-dependent nature of traffic flow, such as rush hours or specific periods of the day.
- Junction: The junction identifier helps us pinpoint the location of each traffic observation, allowing for the analysis of traffic flow at specific junctions.
- Vehicles: This attribute reflects the number of vehicles passing through a junction during a given

Table 1. Dataset attributes.

Attribute	Description	Data Type
DateTime	Timestamp of the recorded observation	Object
Junction	Identifier for the junction location	Integer
Vehicles	Count of vehicles observed per hour	Integer
ID	Unique identifier for each observation	Integer

Table 2. Statistical overview of vehicle counts.

Metric	Value
Mean	22.79
Variance	430.57
Standard Deviation	20.75
Minimum	1
Maximum	180

hour. It is the primary feature we aim to predict, as it directly impacts traffic flow and congestion.
- ID: A unique identifier for each record, ensuring the uniqueness of each observation.

To better (Table 2) understand the distribution of vehicle counts, we calculated basic statistical measures:

- Mean: 22.79 vehicles per hour (average vehicle count)
- Variance: 430.57, indicating significant fluctuations in vehicle counts
- Standard Deviation: 20.75, showing variability in the data
- Minimum: 1 vehicle (lowest observed count)
- Maximum: 180 vehicles (highest observed count)

Additional features such as CarCount, BikeCount, BusCount, and TruckCount were extracted from the dataset. These features represent the number of vehicles of each type passing through the junction, providing valuable insight into traffic volume. The Traffic Situation attribute categorizes traffic conditions as low, medium, or high, based on vehicle counts, helping the model understand traffic density.

3.2. Data preprocessing

Before training the models, we prepared the dataset for effective model learning. The steps involved:

- Min-Max Normalization: This technique scaled the vehicle counts to a range of 0 to 1, ensuring that the data was on a similar scale. This prevents larger values from overpowering the learning process.
- Handling Class Imbalance: Given that there were more records of low traffic hours than high traffic hours, we used SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic data points for the less frequent high-traffic periods. This helps the model avoid bias towards low traffic.
- Feature Selection: We selected the most relevant features for training the models, including Junction, DateTime, and Vehicles, which are essential for predicting traffic flow.

3.3. Model descriptions

We employed three machine learning models to predict traffic flow, each with distinct characteristics and strengths:

3.3.1. Recurrent neural network (RNN)

The RNN is designed for sequential data like time-series traffic data. It learns from the data step by step, with each step connected to the previous one. This allows the model to recognize time-based patterns. We used the Adam optimizer with a learning rate of 0.001 for optimization.

3.3.2. Long short-term memory (LSTM)

The LSTM is an improved version of the RNN, designed to capture long-term dependencies in sequential data. It is especially useful for recognizing recurring traffic patterns, such as daily or weekly rush hours. LSTMs have specialized gates (input, forget, and output) that help manage information flow, allowing them to store relevant data and discard irrelevant data over time.

3.3.3. CNN-LSTM hybrid model

The CNN-LSTM model combines the power of Convolutional Neural Networks (CNNs) and LSTMs. The CNN component helps detect spatial patterns, such as congestion at specific junctions, while the LSTM component handles the temporal aspect, learning how these patterns evolve over time. This hybrid approach is ideal for capturing both the spatial and temporal characteristics of traffic flow.

3.4. Model training and evaluation

3.4.1. Training process

We split the dataset into two parts: 80% for training and 20% for testing. This division ensured that the model learned from a large portion of the data while also being evaluated on unseen data to assess its generalization capability.

The models were trained with the following settings:

- Batch size: 64, a standard value that balances training speed and stability.

- Epochs: 50, allowing the model to learn traffic patterns effectively without overfitting.
- The Adam optimizer with a learning rate of 0.001 was used to adjust the model's weights during training.

3.4.2. Evaluation metrics

We evaluated model performance using the following metrics:

- Accuracy: Measures the overall correctness of the model.
- Precision: Assesses how well the model identifies high-traffic periods.
- Recall: Evaluates the model's ability to detect traffic events.
- F1-score: Provides a balance between precision and recall.
- Root Mean Square Error (RMSE): Quantifies the model's overall prediction error, providing insights into its accuracy.
- Mean Squared Error (MSE): Measures the average of the squares of the errors, providing a direct assessment of the model's prediction accuracy, especially when large errors are undesirable.

3.5. Hyperparameter tuning (future work)

Although hyperparameter tuning was not conducted in this study, it is a potential avenue for future work. Methods such as grid search or random search could be employed to explore different combinations of hyperparameters to improve the model's performance.

4. Results

4.1. Recurrent neural networks (RNN)

The Recurrent Neural Network (RNN) model was applied to a dataset containing key traffic features, including CarCount, BikeCount, BusCount, TruckCount, and Traffic Situation (categorized as low, medium, or high). The dataset underwent preprocessing, cleaning, and normalization before being split into an 80% training and 20% testing subset. This standard data partition allowed the model to train on a majority of the data while evaluating its generalization capability on unseen data.

RNNs are particularly suited for traffic flow prediction due to their ability to capture temporal dependencies in sequential data, such as traffic patterns. The RNN architecture used for this study included an input layer for traffic count features, followed by multiple RNN layers designed to model the temporal dependencies of traffic data over time. The output layer employed a softmax activation function, appropriate for multi-class classification tasks such as predicting traffic conditions (low, medium, or high). The model was trained for 50 epochs, utilizing a batch size of 64, and was optimized using the Adam optimizer with a learning rate of 0.001.

Table 3. Performance metrics for RNN.

Metric	Low Traffic	Medium Traffic	High Traffic	Overall
Precision	0.82	0.72	0.75	-
Recall	0.84	0.68	0.28	-
F1 Score	0.83	0.70	0.41	-
Accuracy	-	-	-	73%

Table 4. Confusion matrix for RNN.

Actual/Predicted	Low Traffic	Medium Traffic	High Traffic
Low Traffic	320	45	15
Medium Traffic	40	200	65
High Traffic	10	60	50

The RNN model achieved an overall accuracy of 73%, with strong performance in low-traffic conditions. Specifically, the model demonstrated a precision of 0.82 and a recall of 0.84 for low traffic (Table 3). However, its performance in high-traffic conditions was suboptimal, with a precision of 0.75 and a recall of only 0.28, indicating that the model frequently misclassified high traffic as medium or low. This discrepancy can likely be attributed to class imbalance, as instances of low traffic were more prevalent than those of high traffic within the dataset.

The confusion matrix (Table 4) highlights the RNN's tendency to misclassify high-traffic instances as medium traffic. This issue is likely due to the class imbalance, where low traffic instances dominated the dataset, affecting the model's ability to correctly classify high traffic instances.

The line graph (Fig. 4) further illustrates the model's prediction capabilities, showing the actual and predicted traffic volumes over time. While there are minor deviations, the predicted traffic volume closely follows the actual traffic trends, indicating the model's general effectiveness in capturing the traffic flow patterns.

4.2. Long short-term memory (LSTM)

The Long Short-Term Memory (LSTM) model was applied to the same traffic dataset, following the preprocessing, normalization, and train-test split process. LSTMs are particularly advantageous in capturing long-term dependencies in time-series data, making them ideal for traffic flow prediction, which

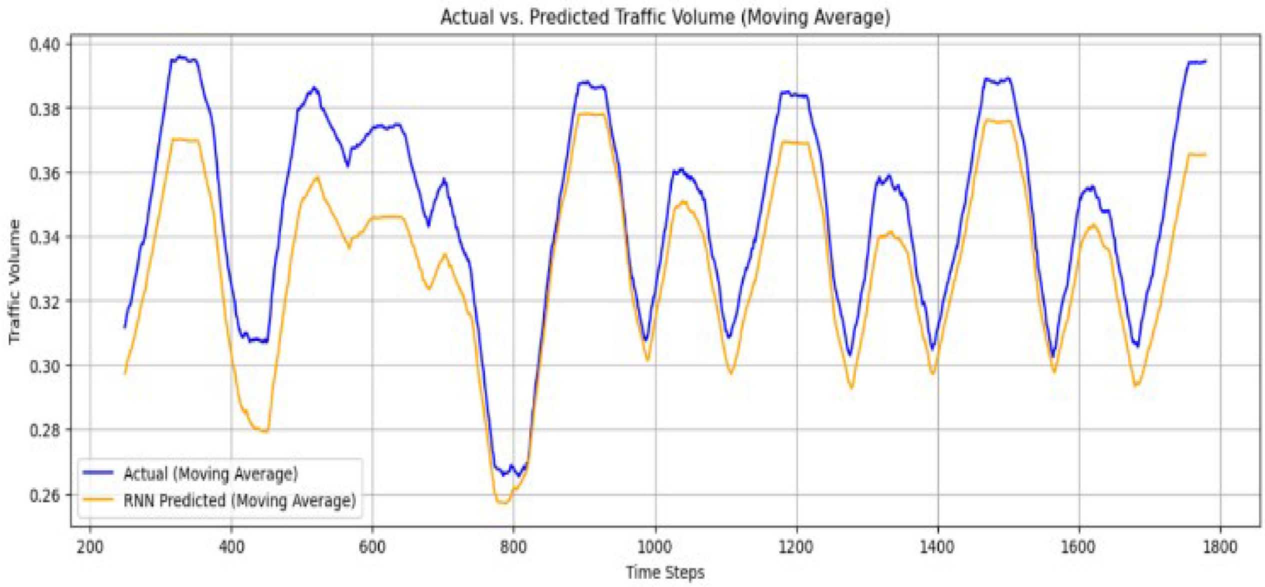

Fig. 4. Actual vs RNN predicted traffic volume (moving average).

Table 5. Performance metrics for LSTM.

Metric	Low Traffic	Medium Traffic	High Traffic	Overall
Precision	0.85	0.75	0.78	-
Recall	0.87	0.70	0.35	-
F1 Score	0.86	0.72	0.48	-
Accuracy	-	-	-	78%

Table 6. Confusion matrix for LSTM.

Actual/Predicted	Low Traffic	Medium Traffic	High Traffic
Low Traffic	330	40	10
Medium Traffic	35	210	60
High Traffic	5	55	60

often relies on historical patterns. The LSTM architecture included an input layer for traffic features, LSTM layers to capture temporal dependencies, and an output layer with a softmax activation function. The model was trained for 50 epochs with a batch size of 64, using the Adam optimizer with a learning rate of 0.001.

The LSTM model achieved an accuracy of 78%, demonstrating superior performance compared to the RNN in low-traffic conditions, with a precision of 0.85 and a recall of 0.87. While the model also showed improvement over the RNN in high-traffic conditions, with a precision of 0.78 and recall of 0.35, there remains room for enhancement in accurately predicting high traffic. LSTMs can better model the temporal dependencies inherent in traffic data, which likely contributed to their improved performance over the RNN (Table 5).

The confusion matrix (Table 6) illustrates fewer misclassifications in the LSTM model, particularly in distinguishing between medium and high traffic, which was a challenge for the RNN.

The line graph (Fig. 5) comparing actual and predicted traffic volumes demonstrates that the LSTM model closely tracks actual traffic trends, with minimal fluctuations during peak traffic hours, indicating its ability to model overall traffic patterns effectively.

4.3. CNN-LSTM hybrid model

CNN-LSTM hybrid model was designed to leverage the strengths of both Convolutional Neural Networks (CNN) for feature extraction and Long Short-Term Memory (LSTM) layers for sequential learning. This hybrid approach is particularly suited for traffic flow prediction, as it can capture both spatial and temporal dependencies within traffic data, which are crucial for accurate forecasting in complex urban environments. The CNN-LSTM model was trained for 50 epochs, utilizing a batch size of 64, and optimized with the Adam optimizer.

The CNN-LSTM hybrid model achieved the highest accuracy of 81%, outperforming both the RNN and LSTM models, particularly in predicting high-traffic conditions. The recall for high traffic was 0.50, a notable improvement over the recall scores of the RNN and LSTM models. This suggests that the CNN-LSTM model was more successful in capturing the patterns of high-traffic conditions, which had been a challenge for the other models.

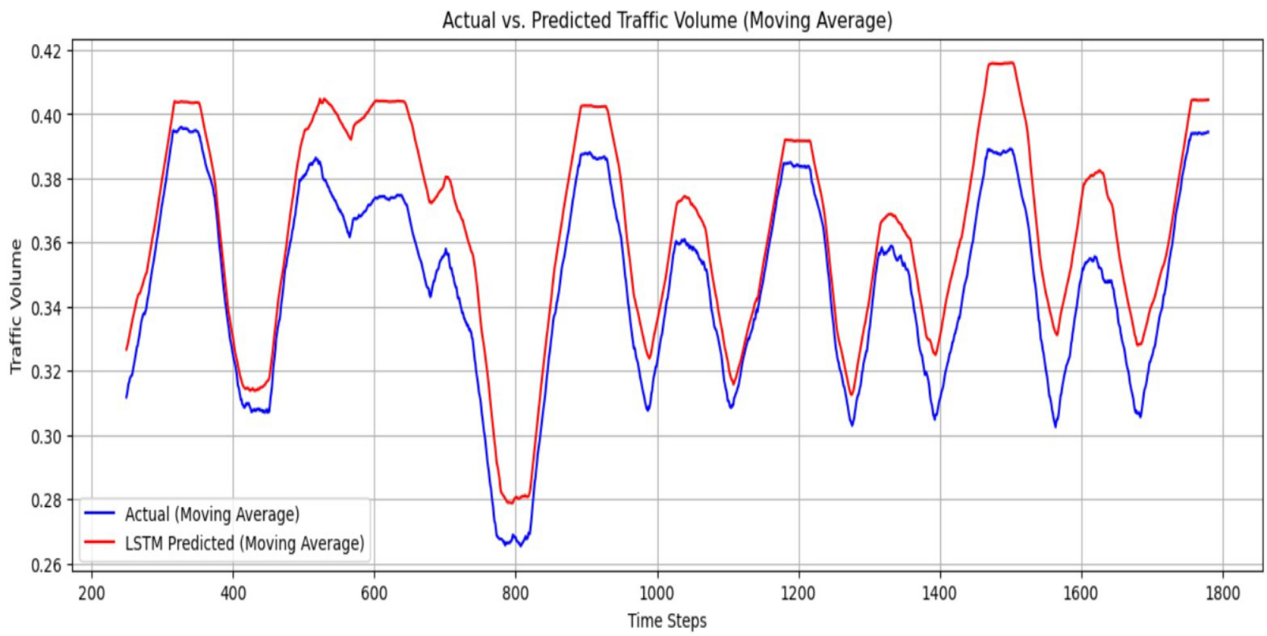

Fig. 5. Actual vs LSTM predicted traffic volume (moving average).

Table 7. CNN-LSTM performance metrics.

Metric	Low Traffic	Medium Traffic	High Traffic	Overall
Precision	0.88	0.78	0.80	-
Recall	0.90	0.75	0.50	-
F1 Score	0.89	0.76	0.61	-
Accuracy	-	-	-	81%

Table 8. Confusion matrix for CNN-LSTM.

Actual/Predicted	Low Traffic	Medium Traffic	High Traffic
Low Traffic	335	35	10
Medium Traffic	25	215	65
High Traffic	5	50	70

The confusion matrix (Table 8) for the CNN-LSTM model reveals its superior performance, with fewer misclassifications across all traffic categories, particularly in distinguishing high traffic.

The line graph (Fig. 6) comparing the actual (blue line) and predicted (orange line) traffic volumes further demonstrates the CNN-LSTM model's superior ability to track traffic trends, particularly during high-traffic periods, showcasing its proficiency in modeling both spatial and temporal patterns within the data.

4.4. Error analysis

The models were also evaluated using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) to assess the prediction accuracy more comprehensively. The results indicated the following:

RNN: MSE $= 0.0118$, RMSE $= 0.1087$
CNN-LSTM: MSE $= 0.0095$, RMSE $= 0.0977$
LSTM: MSE $= 0.0110$, RMSE $= 0.1049$

The MSE and RMSE results highlight that the CNN -LSTM model outperforms both the RNN and LSTM models in terms of prediction accuracy, as evidenced by the lower error values.

5. Discussion

This study assessed the performance of three machine learning models—Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and CNN-LSTM (a hybrid of Convolutional Neural Networks and Long Short-Term Memory)—in predicting traffic flow. The models were evaluated based on several performance metrics, including accuracy, precision, recall, and F1-score. Of the three models, CNN-LSTM yielded the highest performance, achieving an accuracy of 81%, followed by LSTM with 78%, and RNN with 73%. The superior performance of the CNN-LSTM model can be attributed to its ability to capture both spatial and temporal dependencies, which are crucial for traffic flow prediction. Traffic flow is influenced not only by temporal factors (such as daily or weekly traffic patterns) but also by spatial factors (such as localized congestion at junctions). The CNN component of the CNN-LSTM model extracts spatial features, while the LSTM component captures temporal dependencies, allowing the model to effectively

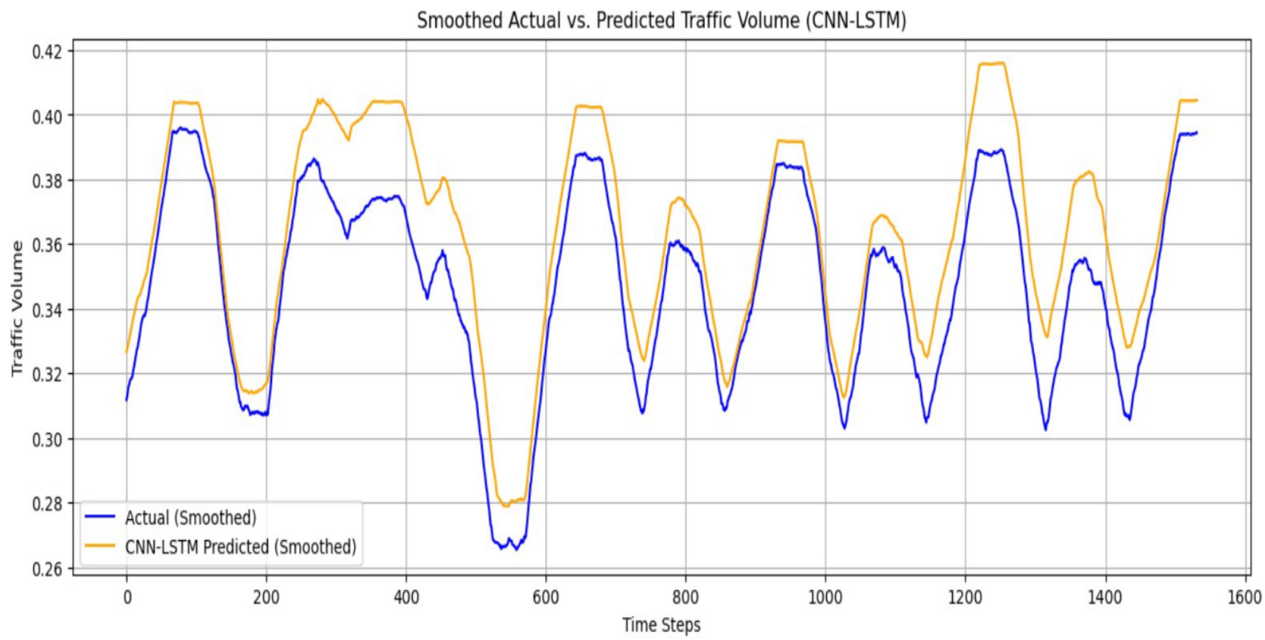

Fig. 6. Actual vs CNN-LSTM predicted traffic volume (moving average).

predict traffic flow in both low and high-traffic conditions.

Although the LSTM model performed well, it did not achieve the same level of accuracy as CNN-LSTM, with a final accuracy of 78%. LSTM models are particularly adept at learning long-term dependencies, such as recurring traffic patterns (e.g., rush hour traffic or weekly trends). However, they face challenges in high-traffic situations where sudden changes—due to accidents, weather, or other factors—disrupt predictable patterns. The recall for high-traffic conditions was notably low (0.35), suggesting that LSTM struggles to adapt to unpredictable changes in traffic flow, which emphasizes the need for incorporating both spatial and temporal information in predicting traffic patterns.

The RNN model, which performed the weakest with an accuracy of 73%, is limited by its inability to capture long-term dependencies effectively. The vanishing gradient problem, a common issue in RNNs, prevents the model from maintaining information over extended periods, making it less suitable for predicting traffic flow in dynamic environments. This limitation became evident in high-traffic scenarios, where RNNs misclassified traffic levels, failing to recognize sudden spikes or shifts in flow. Additionally, class imbalance in the dataset further contributed to the model's poor performance, particularly in predicting high-traffic conditions.

5.1. Handling non-linear and dynamic aspects of traffic flow

Traffic flow is inherently non-linear and subject to sudden, unpredictable changes, influenced by factors such as accidents, weather, and special events. These elements introduce variability and noise into the data, posing significant challenges for traffic prediction models. The CNN-LSTM model demonstrated the greatest adaptability to these fluctuations, effectively modeling both regular traffic patterns and sudden disruptions. The CNN component captured spatial congestion patterns, while the LSTM component modeled how these patterns evolved over time, thus improving the model's robustness in handling unpredictable traffic conditions.

On the other hand, while the LSTM model excelled at learning long-term patterns, it faced difficulties when confronted with unexpected disruptions. LSTM was effective in predicting traffic during predictable conditions, such as rush hours or weekends, but struggled to adapt to sudden traffic changes caused by accidents or road closures. This underscores the necessity of integrating both spatial and temporal features to improve the accuracy of predictions in dynamic, real-time traffic scenarios.

The RNN model, despite being able to model short-term dependencies, performed poorly in high-traffic conditions due to its inability to retain long-term memory across multiple time steps. The model was unable to effectively identify or adapt to sudden spikes in traffic, resulting in misclassifications, especially in high-traffic periods.

5.2. Challenges and model adaptation

One of the key challenges in this study was the variability in traffic data caused by external factors such as weather conditions, accidents, and public holidays. These elements can cause sudden fluctuations in traffic patterns, complicating the task of predicting traffic flow. The CNN-LSTM model's ability to combine spatial and temporal features allowed it to adapt better to these changes. By capturing both the spatial distribution of traffic and the temporal evolution of these patterns, the CNN-LSTM model was able to handle fluctuations caused by unpredictable events, such as accidents or road closures.

Although the LSTM model showed better performance than RNN, it still struggled with real-time traffic fluctuations. While it excelled at predicting long-term trends like daily traffic peaks, it did not perform as well in scenarios where sudden, unexpected changes occurred. These challenges highlight the importance of incorporating spatial data—such as traffic volume by vehicle type—and accounting for external factors like weather, which can impact traffic flow unpredictably.

5.3. Hyperparameter optimization and model tuning

Hyperparameter optimization played a crucial role in model performance. In this study, 50 epochs were selected after preliminary testing to balance model training time with performance. Future work could explore techniques like grid search or random search to fine-tune hyperparameters such as batch size, learning rate, and the number of epochs. In particular, the Adam optimizer with a learning rate of 0.001 provided a stable convergence without overshooting the optimal solution. Future research could also explore regularization techniques such as dropout or L2 regularization to prevent overfitting, especially in environments with significant data fluctuations.

5.4. Transfer learning and its relevance for traffic prediction

The potential use of transfer learning in the CNN component of the CNN-LSTM model for traffic flow prediction represents an exciting avenue for future research. Transfer learning typically involves pre-training a CNN on large, labeled datasets (such as ImageNet) and then fine-tuning the model for a specific task. CNN's ability to capture spatial features is advantageous in many domains, but in the context of traffic flow prediction, transferring spatial feature extraction from image data to traffic data requires careful consideration.

While both traffic flow data and images exhibit spatial patterns, the nature of these patterns differs. In traffic flow data, spatial features represent vehicle distribution (e.g., car count, bike count, truck count) and traffic conditions, which can change dynamically based on time of day, weather, or other factors. In contrast, the spatial patterns in images typically represent static features such as objects and textures. As such, adapting CNNs to effectively capture traffic-specific spatial relationships, such as congestion or traffic volume distribution, is a non-trivial task. Further research is needed to explore how transfer learning can be used to adapt CNNs for traffic flow prediction, ensuring that spatial features learned from image data can be effectively applied to dynamic traffic patterns.

5.5. Practical applications and implications

The results of this study have several practical implications for urban transportation systems. The CNN-LSTM model, with its ability to capture both spatial and temporal dependencies, is well-suited for real-time traffic management systems. For example, the model can be employed to dynamically adjust traffic signal timings based on predicted traffic conditions, reducing congestion and improving traffic flow. Additionally, the model can support route optimization systems by predicting traffic flow in real time, allowing drivers to be directed toward less congested routes during peak times or in the event of an accident.

Furthermore, this model can aid in urban planning by providing predictions of peak traffic times. Cities can use these insights to better prepare for heavy traffic, optimize road usage, and design more efficient public transportation systems. By reducing congestion and optimizing routes, the model can contribute to reducing fuel consumption and lowering emissions, offering both environmental and societal benefits.

5.6. Integration into existing traffic management systems

A key application of this research is the integration of traffic flow predictions into existing traffic management infrastructure. Real-time predictions of traffic patterns enable more intelligent decision-making in traffic signal control, route optimization, and road usage management. For instance, traffic signals could be dynamically adjusted based on the predicted congestion levels, ensuring smoother traffic flow and reduced delays. Additionally, the model's ability to predict high-traffic conditions, such as accidents or road closures, allows for prompt interventions, such as rerouting traffic or adjusting signal timings, to minimize disruptions. This capacity could improve the efficiency and resilience of urban transport systems, benefiting both commuters and city planners.

6. Conclusion

This study demonstrated that the CNN-LSTM hybrid model outperforms traditional models like LSTM and RNN in traffic flow prediction, achieving the highest accuracy and effectively capturing the complex, dynamic nature of traffic patterns. The CNN-LSTM model excelled particularly in predicting high-traffic conditions, where both LSTM and RNN struggled. By integrating both spatial and temporal dependencies, the CNN-LSTM model provides a more accurate and robust solution for real-time traffic prediction, making it a promising tool for optimizing traffic flow and reducing congestion in urban areas.

The contributions of this study are twofold. First, it showcases the effectiveness of hybrid models, particularly CNN-LSTM, in balancing prediction accuracy with computational efficiency. This hybrid approach enables more precise predictions, which are crucial for real-time traffic management systems. Second, it emphasizes the potential of CNN-LSTM to improve real-time traffic management by addressing both spatial (congestion patterns) and temporal (time-dependent) variations, which are key to traffic flow prediction.

However, several avenues for future research could further enhance the model's performance. Integrating real-time data sources, such as live weather updates, traffic events (e.g., accidents or road closures), and social media feeds, could increase the model's adaptability to sudden disruptions. Additionally, testing the CNN-LSTM model on larger, more diverse datasets from multiple cities would help improve its scalability and generalizability, enabling its application in a wider range of traffic conditions. Further investigation into other hybrid models, such as those combining CNN-LSTM with reinforcement learning or graph neural networks, could provide deeper insights into optimizing both the accuracy and speed of traffic flow predictions. Finally, exploring optimization techniques, like hyperparameter tuning and transfer learning, could make the CNN-LSTM model more efficient in real-time, large-scale applications.

By focusing on these areas, future research could significantly advance the capabilities of traffic prediction systems, offering scalable, real-time solutions that could ultimately enhance urban mobility and reduce congestion.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this research on traffic flow prediction.

References

1. Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, Apr. 2015.
2. X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, pp. 818, Apr. 2017.
3. Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning-based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, Nov. 2018.
4. B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," In *Proc. 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2018, pp. 3634–3640.
5. S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," In *Proc. AAAI Conf. on Artificial Intelligence*, vol. 33, no. 1, pp. 922–929, 2019.
6. Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," In *International Conference on Learning Representations (ICLR)*, 2018.
7. Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," *IEEE Access*, vol. 8, pp. 62797–62806, 2020.
8. N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, Jan. 2017.
9. Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.
10. S. Du, T. Li, X. Gong, and S. J. Horng, "A hybrid method for traffic flow forecasting using multimodal deep learning," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 85–97, 2020.
11. C. R. Selvan, S. Senthil Kumar, P. Iwin Thanakumar Joseph, M. Malin Bruntha, K. Amanullah, and V. Arulkumar, "Traffic prediction using GPS based cloud data through RNN-LSTM-CNN models: Addressing road congestion, safety, and sustainability in smart cities," *SN Computer Science*, vol. 6, no. 2, p. 159, 2025.
12. S. K. Mathivanan, H. Rajadurai, J. Cho, and S. V. Easwaramoorthy, "A multi-modal geospatial–temporal LSTM based deep learning framework for predictive modeling of urban mobility patterns," *Scientific Reports*, vol. 14, no. 1, p. 31579, Dec. 2024.
13. J. H. Zhao and T. Al-Dala'in, "The hybrid model combination of deep learning techniques, CNN-LSTM, BERT, feature selection, and stop words to prevent fake news," In *The International Conference on Innovations in Computing Research*, 2024, pp. 173–184, Cham: Springer Nature Switzerland.

14. M. Telmem, N. Laaidi, Y. Ghanou, S. Hamiane, and H. Satori, "Comparative study of CNN, LSTM and hybrid CNN-LSTM model in amazigh speech recognition using spectrogram feature extraction and different gender and age dataset," *International Journal of Speech Technology*, vol. 27, no. 4, pp. 1121–33, Dec. 2024.
15. D. N. Maurya, D. Arora, and C. P. Singh, "A hybrid CNN-LSTM deep neural network model for efficient human activity recognition," In *International Conference on Deep Learning and Visual Artificial Intelligence*, 2024, pp. 403–411, Singapore: Springer Nature Singapore.
16. K. Ullah, M. Ahsan, S. M. Hasanat, M. Haris, H. Yousaf, S. F. Raza, R. Tandon, S. Abid, and Z. Ullah, "Short-term load forecasting: A comprehensive review and simulation study with CNN-LSTM hybrids approach," *IEEE Access*, 2024 Aug 8.
17. P. Datta and R. Rohilla, "An autonomous and intelligent hybrid CNN-RNN-LSTM based approach for the detection and classification of abnormalities in brain," *Multimedia Tools and Applications*, vol. 83, no. 21, pp. 60627–53, Jun. 2024.
18. H. Almukhalfi, A. Noor, and T. H. Noor, "Traffic management approaches using machine learning and deep learning techniques: A survey," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108147, 2024 Jul 1.
19. I. D. Mienye, T. G Swart, and G. Obaido G, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," *Information*, vol. 15, no. 9, p. 517, 2024 Aug 25.
20. Y. Gao, D. Tian, D. Grana, Z. Xu, and H. Xu, "Attention mechanism-assisted recurrent neural network for well log lithology classification," *Geophysical Prospecting*, vol. 73, no. 2, pp. 628–49, 2025 Jan 26.
21. M. Omer, S. J. Ali, S. M. Raza, D. T. Le, and H. Choo, "Integrating temporal analysis with hybrid machine learning and deep learning models for enhanced air quality prediction," In *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2025, pp. 1–7, IEEE.
22. S. Bai, B. Dai, Z. Yang, F. Zhu, W. Yang, and Y. Li, "Enhancing road surface temperature prediction: A novel approach integrating transfer learning with long short-term memory neural networks," *Journal of Transportation Engineering, Part B: Pavements*, vol. 151, no. 1, p. 04024063, 2025 Mar 1.
23. C. Wang, S. Huang, and C. Zhang, "Short-term traffic flow prediction considering weather factors based on optimized deep learning neural networks: Bo-GRA-CNN-BiLSTM," *Sustainability*, vol. 17, no. 6, p. 2576, 2025 Mar 14.
24. N. Zafar, I. U. Haq, J. U. Chughtai, and O. Shafiq, "Applying hybrid LSTM-GRU model based on heterogeneous data sources for traffic speed prediction in urban areas," *Sensors*, vol. 22, no. 9, p. 3348, 2022 Apr 27.
25. S. Narmadha and V. Vijayakumar, "Spatio-temporal vehicle traffic flow prediction using multivariate CNN and LSTM model," *Materials today: proceedings*, vol. 81, no. 826–33, 2023 Jan 1.
26. Z. Islam, M. Abdel-Aty, and N. Mahmoud, "Using CNN-LSTM to predict signal phasing and timing aided by high-resolution detector data," *Transportation research part C: emerging technologies*, vol. 141, p. 103742, 2022 Aug 1.
27. Y. Wang, "Advances in spatiotemporal graph neural network prediction research," *International Journal of Digital Earth*, vol. 16, no. 1, pp. 2034–66, 2023 Oct 2.