

Traffic Modelling Using Convolutional Long-Short Term Memory Neural Network

Nicolae FLORIAN^{1*}, Dumitru POPESCU^{1,2}, Severus-Constantin OLTEANU¹

¹ Faculty of Automatic Control and Computers, Department of Systems Engineering, National University of Science and Technology Politehnica Bucharest, Splaiul Independenței 313, 060042 Bucharest, Romania
florian_nick@windoslive.com (*Corresponding author), dumitru.popescu@upb.ro, severus.olteanu@upb.ro

² Romanian Academy of Scientists, 125 Calea Victoriei, 010071 Bucharest, Romania

Abstract: This paper examines hybrid neural networks for traffic management and prediction. These networks combine recurrent and convolutional neural network architectures. The current studies on FNN, RNN, and CNN architectures in this field are reviewed and the conventional traffic modelling techniques and their limitations are also evaluated. The proposed hybrid model integrates the strengths of two architectures: it uses RNNs for identifying the temporal dependencies in traffic data and it employs CNNs for recognizing spatial patterns. The paper details the employed methodology, which includes data collection, network architecture design, training, hyperparameter tuning, and model performance evaluation. The implications of the findings of this paper for urban planning and intelligent transportation systems (ITS) are discussed and the obtained results are compared with those achieved by the conventional approaches.

Keywords: Traffic, Modelling, Neural network, Convolution, Prediction, Intelligent transportation system.

1. Introduction

This paper investigates the potential of neural networks in traffic modelling applications. The current studies on the application of FNN, RNN, and CNN architectures in this field are examined and conventional traffic modelling techniques and their drawbacks are evaluated. The proposed hybrid model combines the advantages of RNNs for identifying temporal dependencies in traffic data and CNNs for recognizing spatial patterns. This methodology covers data collection, network architecture design, training, hyperparameter tuning, and performance evaluation, and it is explained in detail in this paper. The implications of the findings of this paper for urban planning and intelligent transportation systems (ITS) are discussed and compared with those obtained by using conventional approaches.

Neural networks, a branch of machine learning, have shown promise in handling large datasets and capturing non-linear relationships (Murphy, 2012; LeCun, Bengio & Hinton, 2015). The goal is to build traffic models that are more precise and flexible, and that improve traffic management and prediction in urban transportation systems (McCrea & Moutari, 2010; Helbing et al., 2002).

When trying to understand and supervise urban transportation systems, traffic modelling is essential. It offers insights on travel times, traffic patterns, areas prone to congestion and the effects of different interventions like building new roads or improving public transportation. In order to maximize the infrastructure investments,

implement the intelligent traffic control strategies in effect, and improve the overall mobility within cities, accurate models are important.

Conventional traffic modelling techniques use simulation methods, statistical methods, or a mix of the two.

Because of their simplicity, statistical models such as Autoregressive Integrated Moving Average (ARIMA) (Nelson, 1998) have found widespread use, but they often fall short in reflecting the nonlinearities found in real-world traffic data. Simulation models like MACROSCOPE require a high level of processing power even though they are physically based and may not capture the intricate interactions between road users adequately (Karpathy et al., 2015).

Statistical models often use time series analysis techniques to identify trends and relationships in historical traffic data to forecast future traffic patterns.

Restrictions include the inability to capture complex non-linear dynamics, the static nature of the model, and the limited spatial representation. Simulation models consider variables such as road capacity, driver behaviour, and incident occurrences to model the movement of vehicles on a network (Karpathy et al., 2015).

Traffic is treated as a continuous fluid flow in a continuum simulation, controlled by macroscopic conservation laws such as LWR. It lacks

individual vehicle representation but is suitable for studying the behaviour of large networks (Karpathy et al., 2015).

In agent-based traffic simulation each vehicle acts as an agent that makes its own decisions based on a set of rules. This kind of simulation shows what happens when vehicles interact with each other at the level of individual vehicles, but it can be computationally expensive. Hybrid traffic models try to combine different characteristics of different types of models so one can better understand how traffic really works and calculations can still be done quickly (Cantarella & De Luca, 2005; Epelbaum et al., 2017).

Neural networks excel in modelling traffic systems because of their ability to learn patterns from large datasets. They find relationships and causations that are more complex than linear ones, and they are frequently used for regression tasks. They can be used to predict traffic flow based on historical data. However, feedforward neural networks (FNNs) have limitations. One of their biggest shortcomings is their inability to capture temporal dynamics. They have difficulty with dependency relationships. While FNNs are a useful type of neural networks, modelling traffic flow is a project for more powerful neural networks (Epelbaum et al., 2017).

Recurrent Neural Networks (RNNs) are designed to work with sequences of inputs. This gives them the ability to capture what happens over time. Therefore, they are able to learn traffic patterns and recognize a recurring behaviour. Some RNNs can carry memory over from previous time periods, which is useful when analysing traffic. RNNs include Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Unit (GRU) neural networks. These networks are especially good at seeing how things are connected over periods of time.

Convolutional Neural Networks are usually used for identifying patterns in pictures (Yamashita et al., 2018). They can also be used to understand traffic by representing the road network as a map. Convolutional Neural Networks are good at finding patterns in various areas of the map. The problem with using Convolutional Neural Networks for traffic is that they might not be able to see how things are connected if they are apart, on the map.

Feedforward Neural Networks (FNNs) are basic neural network architectures used for regression

tasks, predicting the future traffic flow based on historical patterns. This type of networks features the following limitation: the inability to capture temporal dependencies effectively (Epelbaum et al., 2017).

Recurrent Neural Networks (RNNs) were created to work with sequential data such as time series, allowing them to learn long-term dependencies in traffic patterns (Sewall, Wilkie & Lin, 2011). The most relevant RNNs for this paper are LSTMs (Long Short-Term Memory) and GRUs (Gated Recurrent Units), which are particularly effective for capturing long-range temporal correlations (Epelbaum et al., 2017).

Convolutional Neural Networks (CNNs) are well known for their cost-effectiveness and precision in image processing. CNNs can be applied to traffic modelling by representing road networks as spatial grids or maps. They excel at identifying patterns and relationships within localized areas of the network. This type of approach features the following limitation: it may struggle to capture long-range dependencies across large distances on a map (Epelbaum et al., 2017).

The contribution of this paper in this field lies in the development and validation of a highly optimized hybrid CNN-LSTM framework. This framework explicitly balances spatial feature extraction with temporal dynamic modelling to address the limitations of systems based on a single architecture. The framework is validated on multi-city datasets, including cities like London, Paris and San Francisco, which reduces the risk of overfitting to a specific urban topology. Hyperparameters were selected by Grid Search Cross-Validation, and the model is more accurate than traditional baselines such as ARIMA while remaining efficient enough for real-time use in Intelligent Transportation Systems (ITS).

The remainder of this paper is organized as follows. Section 2 details the methodology, encompassing data collection and the proposed architecture. Section 3 presents the experimental results obtained. Further on, Section 4 provides a discussion of the findings. Finally, Section 5 concludes this paper.

2. Methodology

Traffic flow refers to the volume of vehicles on the road, their speed, and spatial distribution. This data is obtained from sensors (loops, cameras) and

GPS devices. Additionally, road network topology significantly influences traffic dynamics. This includes lane count, intersection configurations, and traffic lights. Weather conditions, such as temperature, rainfall, and visibility also play a role. External factors, like special events, accidents, and construction works, further affect traffic.

For this study, traffic data included flow, speed, occupancy, and turning movement counts (TMCs). This data was collected from connected vehicle testbeds. Specifically, the Urban Challenge datasets from 2015 and 2018 were used (Sewall, Wilkie & Lin, 2011; Lieberthal et al., 2024). These open-source datasets, made public by diverse city administrations that are open to suggestions regarding the traffic in their city, provide diverse urban traffic scenarios. Data from three major cities, namely London, Paris, and San Francisco was integrated. Using several cities supports the robustness and broad applicability of the results. The raw data from these municipal databases varied with regard to its formatting and structure. Therefore, extensive preprocessing was required to harmonize the inputs for the neural network.

Data preprocessing involved handling missing values and scaling features. This stabilises neural network training. Normalization used Min-Max scaling. It mapped all continuous variables, such as speed and flow volume, into a $[0, 1]$ range. The harmonized dataset was then partitioned. 80% of it was used for training, 15% for validation, and 5% for testing. This partitioning is meant for evaluating the model's generalization capabilities.

The model combines a CNN, which extracts features such as congestion patterns at intersections, with an LSTM, which models how traffic changes over time. The approach relies on the ability of CNNs to analyse traffic patterns and on the ability of LSTMs to retain those patterns over an extended period. It is suitable for predicting traffic flow at given time intervals. The prediction uses flow, speed, occupancy and turning movement count data. The CNN analyses traffic flow maps to identify areas with high congestion or dense traffic. The recurrent (LSTM) layers then process the extracted features step by step and learn how traffic patterns change from one moment to the next. The output layer predicts traffic conditions at specific places and times based on what the model has learned.

The proposed model was trained using traffic data, and its predictions were compared with the actual

outcomes to assess its performance. The model applies machine learning by being presented with past traffic data alongside the correct outcomes for training. The optimizer adjusts the learning rate to the problem, which helps in training networks. Data preprocessing involved normalization using Min-Max scaling and splitting the data into training (80%), validation (15%), and testing (5%) sets, as it was previously mentioned. The model was trained using the Adam optimizer with a learning rate of 0.001 (see Table 1). This optimizer was selected because it performs well for problems of this type (Lieberthal et al., 2024).

The performance of the proposed hybrid model was evaluated by using the standard measures commonly applied to assess the accuracy of time series forecasts. Mean Absolute Percentage Error (MAPE) was used for assessing the accuracy of the model. Also, the Mean Absolute Error (MAE) was employed, which measures the difference between the predictions made by the hybrid model and the actual outcomes, indicating how accurate the hybrid model is overall. Further on, the Root Mean Squared Error, which represents the square root of the average squared differences between the predictions of the hybrid model and the actual outcomes, helps one understand how far the model's estimates are from the true values. To that, R-squared measures how much of the variation in the data one is trying to predict is explained by the hybrid model. It indicates how well this model fits the data.

The approach presented in this study involves using a combination of CNNs for spatial feature extraction. LSTM is used for temporal dynamics modelling, while the congestion patterns are analysed across the intersections. This hybrid architecture uses CNNs' ability to capture local spatial dependencies and LSTMs' capacity to retain long-term memory, which makes it suitable for traffic flow prediction with variable time intervals. This model follows the minimal architecture represented in Figure 1.

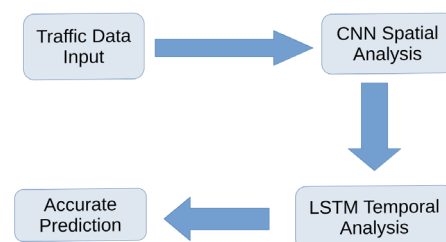


Figure 1. Basic methodological steps and data flow diagram

The proposed CNN-LSTM architecture consists of two convolutional layers, a LSTM layer, and a fully connected output layer.

The first convolutional layer applied a filter f_1 of size $k_1 \times k_1$ (8×8 , see Table 1) with stride s_1 producing feature maps F_1 as described in (Epelbaum et al., 2017):

$$F_1 = \text{ReLU}(\text{Conv}(X, f_1, s_1 + b_1)) \quad (1)$$

The input X denotes the traffic data entering the layer, the bias term b_1 is an additional parameter that shifts the activation function ReLU (Nair & Hinton, 2010), used for introducing non-linearity. The second convolutional layer applied a filter f_2 of size $k_2 \times k_2$ (8×8 , see Table 1) with stride s_2 , producing feature maps F_2 (Epelbaum et al., 2017):

$$F_2 = \text{ReLU}(\text{Conv}(X, f_2, s_2 + b_2)) \quad (2)$$

The output from the convolutional layer was then input into the Long Short-Term Memory layer, described by the following equations:

$$i_t = \sigma(W_i \cdot h_{(t-1)} + U_i \cdot F_2 + b_i) \quad (3)$$

$$f_t = \sigma(W_f \cdot h_{(t-1)} + U_f \cdot F_2 + b_f) \quad (4)$$

$$o_t = \sigma(W_o \cdot h_{(t-1)} + U_o \cdot F_2 + b_o) \quad (5)$$

$$C_t = \tau(W_c \cdot h_{(t-1)} + U_c \cdot F_2 + b_c) \quad (6)$$

$$C_t = f_t \odot C_{(t-1)} + i_t \odot \tau_t \quad (7)$$

$$h_t = o_t \odot \tanh(C_{(t-1)}) \quad (8)$$

Here, the parameters i, f, o denote the input gate, the forget gate and the output gate. The cell state C_t at the given time stamp t represents the memory of the LSTM unit at that moment. The σ function is the sigmoid function. The Hadamard product $\odot$ is a method of multiplying elements in matrices by multiplying the corresponding entries directly. The hidden states denoted by h_t pass through a layer that applies the weights denoted by W_{out} and the bias b_{out} to produce the output. This layer is fully connected, meaning that every input is linked to every output, which helps to produce the intended result.

$$y = \text{Softmax}(W_{out} \cdot h_t + b_{out}) \quad (9)$$

In the proposed CNN-LSTM hybrid model for predicting traffic flow, the Softmax function is used in the output layer to classify traffic into

categories like free flow, moderate congestion, or heavy congestion. This function takes the model's output and converts it into a probability distribution in which context a value of 1 means certainty. This is a key function for the prediction model as it allows the model to not only predict a category the flow of traffic would fall into, but it also shows how confident the model is in predicting the respective category (see also Abdar et al., 2021, on uncertainty quantification in deep learning). This matters for Intelligent Transportation Systems (ITS): the prediction model's outputs can be integrated into advanced traffic management systems, which use the predicted flows and probability estimates as inputs to evaluate risks more accurately and to issue alerts (Bishop, 2006; Bridle, 1990).

The model was constructed using TensorFlow 2.9. This allowed the computations to be run efficiently on the GPU and TPU. These processors, known as parallel processors, can handle a large task load, which also means a large amount of data sent to be analysed. This shortens training and testing considerably.

To obtain certain results based on this model, many different settings were chosen. A method called Grid Search Cross-Validation (GSCV) was used for finding a combination of settings. When this model was tested on certain sample data, it managed to predict traffic flow with a high accuracy. The error was around 8 percent, which is acceptable given that the traffic data came from different periods, so the proposed model predicts traffic flow well. Similar model configurations are also described in (Zhang, 2018; Ma et al., 2015).

The hyperparameters were tuned using GSCV. The experiments carried out involved different numbers of filters (k), of kernel sizes (s) in the CNN layers, and of LSTM units (u). Table 1 below includes the selected parameters, the known values of these parameters in the search space and the optimal values:

On the validation set, this model achieved a MAE of about 8%. That shows it predicts traffic flow data with variable time intervals accurately. The computer used for this project has an eight

Table 1. Hyperparameters: search space values and optimal values

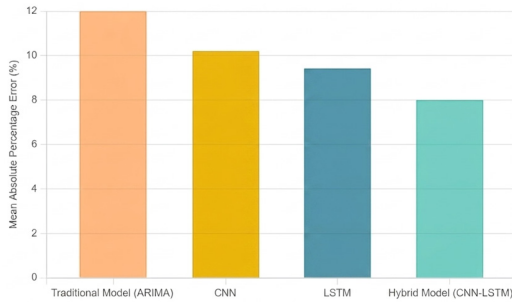
Parameter	Search Space Values	Optimal Value
Number of CNN Filters	16, 32, 64	32
Kernel Size	4, 8, 16	8
Number of LSTM Units	32, 64, 128, 256	64
Batch Size	16, 32, 64	32
Learning Rate	0.01, 0.001, 0.0001	0.001
Dropout Rate	0.1, 0.2, 0.3	0.2

core/sixteen threads AMD Ryzen 7 CPU, with 64 GB RAM and a Nvidia RTX 4060 GPU with 8 GB VRAM.

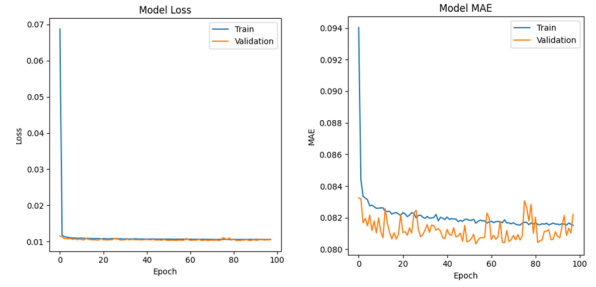
The Adam optimizer was selected for its efficiency in handling high-dimensional parameter spaces and its ability to adjust the learning rate dynamically based on the first and second moments of the gradients. Training was performed with a weight decay regulator to mitigate overfitting (see also Zaremba, Sutskever & Vinyals, 2014, on the regularization of recurrent networks), which helps the model generalize across the different urban centers.

3. Results

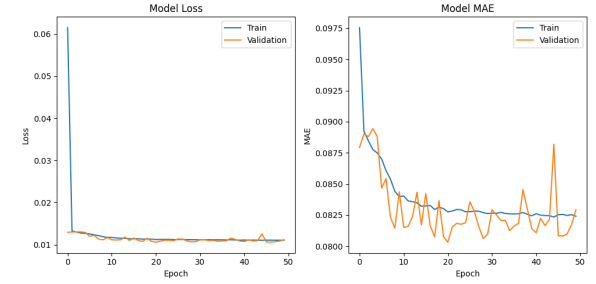
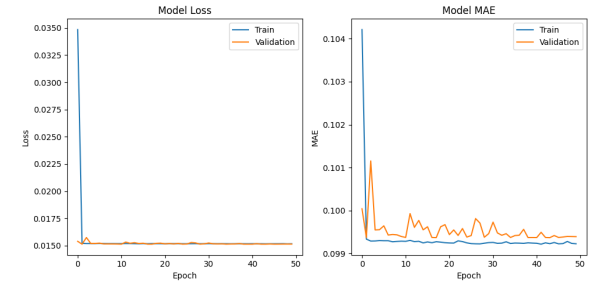
This CNN-LSTM architecture outperformed the traditional statistical models like ARIMA (MAPE: ~12%) and demonstrated a comparable performance in relation to other neural network-based approaches, such as those of Epelbaum et al. (2017), Sewall, Wilkie & Lin (2011) and Halbouni et al. (2022), as shown in Figure 2.

**Figure 2.** Direct comparative performance visualization

The traditional methods struggle to adapt in real time due to either their oversimplified assumptions or the high computational demands they require. The proposed model provides accurate predictions while maintaining an efficiency suitable for real-time use. Its loss (RMSE) was 0.0104 and its MAE 0.0805, as shown in Figure 3, while its MAPE was 0.0001%.

**Figure 3.** CNN-LSTM model loss and error

Other machine learning models, namely the CNN or the LSTM, have obtained the following results, as shown in Figure 4 and Figure 5.

**Figure 4.** CNN spatial feature extraction performance**Figure 5.** LSTM temporal memory metrics

4. Discussion

This study presents a hybrid CNN-LSTM architecture for short-term traffic flow prediction. The proposed model improves the accuracy of predictions compared with traditional prediction methods (e.g., statistical models or simulation-based models). By leveraging the strengths

of CNNs for recognizing spatial patterns and LSTMs for capturing temporal dependencies (i.e. time-related characteristics), the limitations of traditional traffic modelling methods have been addressed. The most notable findings of this study are as follows:

- The hybrid CNN with LSTM model obtained a Mean Absolute Percentage Error (MAPE) of approximately 8%, compared with traditional methods such as ARIMA that featured a MAPE value of around 12%. This indicates that the proposed neural network-based modelling approach has a greater accuracy and reliability than traditional methods when predicting traffic flow.
- This architecture can maintain real-time adaptability, which benefits urban traffic management: accurate traffic forecasts with low latency can improve intelligent transportation systems (ITS) and support effective urban planning strategies.
- This model improves the prediction of traffic flow and can give urban planners and traffic management authorities useful information. With more accurate forecasts they can optimize route planning, reduce congestion and improve roadway safety, and they can identify critical intersections that need traffic management interventions so that proactive measures can be taken.

5. Conclusion

This paper presented the CNN-LSTM architecture for short-term traffic flow prediction with promising results on the available data sets. The model is more accurate than the traditional methods and remains suitable for real-time use in urban traffic management. The convolutional layers handle spatial inputs such as locations

(intersections, road segments), weather conditions and the type of day (workday, weekend, holiday), while the LSTM nodes handle time series of locations and timestamps.

However, while this CNN-LSTM model shows promising results, there are still several ways for improving traffic prediction:

1. Future work will investigate architectures like transformer-based models and other advanced neural network architectures, such as graph neural networks (Wu et al., 2020), including cluster graph wavelet networks for city-scale traffic prediction (Wang et al., 2022) and multi-graph convolution networks for demand forecasting (Geng et al., 2019). These models might work better by understanding complex traffic patterns more effectively.
2. Adding external factors, such as weather conditions or special events, should make the model more accurate and robust. A larger dataset would let the model capture how weather and special events relate to traffic congestion and incidents.
3. “Game Theory” can be used in tandem with traffic prediction to find routes that minimize time or fuel consumption. The efficiency of such approaches is shown by Bilbâie, Dimon & Popescu (2024) and Bilbâie, Dimon & Popescu (2025).
4. This model should be further tested in other cities. Implementing the model in real-world traffic systems and checking its performance in traffic will represent important steps.
5. The way how popular navigation applications, such as Google Maps, affect traffic should be studied. Then, a way to adjust the model could be found, so that it won’t be too reliable on these applications.

REFERENCES

- Abdar, M., Pourpanah, F., Hussain, S. et al. (2021) A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*. 76, 243–297. <https://doi.org/10.1016/j.inffus.2021.05.008>.
- Bishop, C. M. (2006) *Pattern Recognition and Machine Learning*. New York, NY, USA, Springer.
- Bilbâie, R., Dimon, C. & Popescu, D. (2024) Efficiency for Routing Networks Management in Supplier-customer Distribution Systems. *Studies in Informatics and Control*. 33(2), 51–58. <http://doi.org/10.24846/v33i202405>.
- Bilbâie, R., Dimon, C. & Popescu, D. (2025) Efficient decision in ambulance systems management. *Studies in Informatics and Control*. 34(3), 91–100. <https://doi.org/10.24846/v34i3y202509>.
- Bridle, J. S. (1990) Probabilistic interpretation of feedforward classification network outputs, with

- relationships to statistical pattern recognition. In: Fogelman Soulie, F. & Herault, J. (eds.) *Neurocomputing: Algorithms, Architectures and Applications*. Berlin, Heidelberg, Springer-Verlag, pp. 227–236.
- Cantarella, G. E. & De Luca, S. (2005) Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models. *Transportation Research Part C: Emerging Technologies*. 13(2), 121–155. <https://doi.org/10.1016/j.trc.2005.03.001>.
- Epelbaum, T., Gamboa, F., Loubes, J. M. et al. (2017) Deep learning applied to road traffic speed forecasting. [Preprint] <https://arxiv.org/abs/1710.08266> [Accessed: 19th September 2026].
- Geng, X., Li, Y., Wang, L. et al. (2019) Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In: *Proceedings of the AAAI Conference on Artificial Intelligence, 27 January–1 February 2019, Honolulu, USA*. 33(1), 3656–3663.
- Halbouni, A., Gunawan, T. S. & Habaebi, M. H. (2022) CNN-LSTM: Hybrid deep neural network for network intrusion detection system. *IEEE Access*. 10, 99837–99849. <https://doi.org/10.1109/ACCESS.2022.3206425>.
- Helbing, D., Hennecke, A., Shvetsov, V. et al. (2002) Micro- and macro-simulation of freeway traffic. *Mathematical and Computer Modelling*. 35(5–6), 517–547. [https://doi.org/10.1016/S0895-7177\(01\)00200-8](https://doi.org/10.1016/S0895-7177(01)00200-8).
- Hochreiter, S. & Schmidhuber, J. (1997) Long short-term memory. *Neural Computation*. 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Karpathy, A., Johnson, J. & Fei-Fei, L. (2015) Visualizing and understanding recurrent networks. [Preprint] <https://arxiv.org/abs/1506.02078> [Accessed: 19th September 2026].
- LeCun, Y., Bengio, Y. & Hinton, G. (2015) Deep learning. *Nature*. 521, 436–444. <https://doi.org/10.1038/nature14539>.
- Lieberthal, E. B., Serok, N., Duan, J. et al. (2024) Addressing the urban congestion challenge based on traffic bottlenecks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 382(2285), Art. No. 20240095. <https://doi.org/10.1098/rsta.2024.0095>.
- Ma, X., Tao, Z., Wang, Y. et al. (2015) Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*. 54, 187–197. <https://doi.org/10.1016/j.trc.2015.03.014>.
- McCrea, J. & Moutari, S. (2010) A hybrid macroscopic-based model for traffic flow in road networks. *European Journal of Operational Research*. 207, 676–684. <https://doi.org/10.1016/j.ejor.2010.04.011>.
- Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA, MIT Press.
- Nair, V. & Hinton, G. E. (2010) Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10), 21–24 June 2010, Haifa, Israel*. pp. 807–814.
- Nelson, B. K. (1998) Time series analysis using autoregressive integrated moving average (ARIMA) models. *Academic Emergency Medicine*. 5(7), 739–744. <https://doi.org/10.1111/j.1553-2712.1998.tb02605.x>.
- Sewall, J., Wilkie, D. & Lin, M. C. (2011) Interactive hybrid simulation of large-scale traffic. In: *Proceedings of the 2011 SIGGRAPH Asia Conference, 12–15 December 2011, Hong Kong, China*. New York, USA, Association for Computing Machinery, Art. No. 135. <https://doi.org/10.1145/2024156.2024169>.
- Wang, J., Zhang, Y., Lv, Y. et al. (2022) Towards city-scale traffic prediction: Cluster graph wavelet networks. In: *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), 8–12 October 2022, Macau, China*. New York, USA, IEEE. pp. 1388–1393. <https://doi.org/10.1109/ITSC55140.2022.9922245>.
- Wu, Z., Pan, S., Chen, F. et al. (2020) A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*. 32(1), 4–24. <https://doi.org/10.1109/TNNLS.2020.2978386>.
- Yamashita, R., Nishio, M., Do, R. K. G. et al. (2018) Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*. 9(4), 611–629. <https://doi.org/10.1007/s13244-018-0639-9>.
- Zaremba, W., Sutskever, I. & Vinyals, O. (2014) Recurrent neural network regularization. [Preprint] <https://arxiv.org/abs/1409.2329> [Accessed: 19th September 2026].
- Zhang, Z. (2018) Improved Adam optimizer for deep neural networks. In: *Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 4–6 June 2018, Banff, Canada*. New York, USA, IEEE. <https://doi.org/10.1109/IWQoS.2018.8624183>.



This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.