Traffic Flow Prediction for Road Networks Using Deep Learning

Stanley Ekene Uzum
Data Science
University of Roehampton
London, United Kingdom
uzums@roehampton.ac.uk

Abstract—Traffic Flow Prediction is one of the major tasks in urban transport intellegence. In this project, I explore a dataset comprising four junctions and construct a predictive model for traffic flow. The study investigates the efficacy of multivariate deep learning models, including the Custom model, GRU, LSTM, CNN, and MLP, for traffic forecasting. By analyzing the results, we identify the most suitable approach.

The potential impact of this research lies in addressing traffic congestion. A deeper understanding of traffic patterns can inform infrastructure development, ultimately mitigating the problem.
Keywords—**Traffic Flow Prediction, Custom model**, **GRU**, **LSTM**, **CNN**, and **MLP**

I. Introduction

In contemporary urban landscapes, efficient traffic management stands as a crucial facet for sustaining economic activities and ensuring quality of life. Traffic flow prediction serves as a fundamental component in this domain, offering insights into future traffic conditions and facilitating proactive decision-making for congestion alleviation and resource optimization. Deep Learning, a subset of machine learning methodologies, has emerged as a potent tool for tackling the complexities inherent in traffic flow prediction tasks.

This project aims to delve into the realm of traffic flow prediction utilizing diverse deep learning architectures including a custom model, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP). The integration of these methodologies endeavors to unravel intricate patterns inherent in road networks, thus enabling accurate prediction of traffic dynamics.

The conventional methods for traffic prediction often grapple with the intricate spatiotemporal dependencies prevalent in road networks, necessitating the adoption of more sophisticated techniques. Deep Learning models, characterized by their capacity to discern intricate patterns from vast datasets, present a promising avenue for addressing this challenge. The custom model, LSTM, GRU, CNN, and MLP architectures each offer unique capabilities in capturing and extrapolating the complex interplay of variables affecting traffic flow.

In this project, we embark on a comprehensive exploration of these deep learning methodologies applied to traffic flow prediction. Through empirical evaluations utilizing real-world traffic datasets, we assess the efficacy of each model in terms of prediction accuracy, computational efficiency, and scalability. Moreover, we delve into the interpretability of these models, aiming to gain deeper insights into the underlying factors influencing traffic dynamics.

II. Objective

1. **Developing Custom Deep Learning Model:** Construct a custom deep learning model tailored specifically for traffic flow prediction, integrating domain-specific knowledge and leveraging advanced neural network architectures.

2. **Comparative Analysis:** Conduct a comparative analysis of the custom model with established deep learning architectures including LSTM, GRU, CNN, and MLP, to discern their respective strengths and weaknesses in capturing spatiotemporal patterns in traffic data.

3. **Prediction Accuracy Assessment:** Evaluate the prediction accuracy of each model using real-world traffic datasets, employing metrics such as mean absolute error (MAE) and root mean square error (RMSE), to quantify the efficacy of the proposed methodologies.

Interpretability Evaluation: Investigate the interpretability of the developed models to gain insights into the underlying factors influencing traffic dynamics, facilitating a deeper understanding of the model predictions.

III. SIGNIFICANCE

1. **Enhanced Traffic Management:** By leveraging deep learning methodologies for traffic flow prediction, this project aims to contribute to the development of more effective and efficient traffic management strategies. Accurate predictions enable proactive decision-making, leading to reduced congestion, improved traffic flow, and enhanced overall transportation efficiency.

1. **Resource Optimization:** Reliable traffic flow predictions facilitate better resource allocation and utilization, such as optimizing traffic signal timings, managing public transportation schedules, and coordinating infrastructure maintenance activities. This can lead to cost savings and improved operational efficiency for transportation agencies and urban planners.

2. **Environmental Impact:** Efficient traffic management resulting from accurate flow predictions can have a positive environmental impact by reducing fuel consumption, emissions, and overall carbon footprint associated with transportation activities. By minimizing traffic congestion and idling, the project contributes to mitigating air pollution and promoting sustainability in urban environments.

Technological Advancement: The exploration and comparison of various deep learning architectures for traffic flow prediction contribute to the advancement of both transportation science and deep learning methodologies. Insights gained from this research can inform the development of novel algorithms and techniques applicable to a wide range of spatiotemporal forecasting tasks beyond traffic management.

IV. EXPLANATION OF ALGORITHMS AND TECHNIQUES USED

1. Custom Layer:

The CustomLayer class defines a custom layer for the neural network model. It implements a fully connected layer with custom weights and biases.
The layer parameters are initialized using random_normal distribution for weights and zeros for biases.
During the call method, the layer performs matrix multiplication of the input data with the weights, adds biases, and returns the result.

2. Exponential Decay Learning Rate:

The ExponentialDecay class implements an exponential decay learning rate schedule.
It gradually decreases the learning rate over training epochs, allowing the model to converge smoothly.
Parameters such as initial learning rate, decay steps, and decay rate are specified to control the decay behavior.

3. Custom Model:

The Custom_model function constructs a neural network model with a custom architecture.
It consists of a Flatten layer to flatten the input data, followed by Dense layers with ReLU activation functions.

Dropout layers are added to prevent overfitting by randomly dropping a fraction of input units during training.

The custom layer defined earlier is incorporated into the model architecture.
Stochastic Gradient Descent (SGD) optimizer with exponential decay learning rate is used for optimization.
Early stopping callback is employed to halt training when validation loss stops improving, thus preventing overfitting.

4. GRU Model:

The GRU_model function builds a Gated Recurrent Unit (GRU) model for sequence prediction.
It comprises multiple GRU layers stacked on top of each other, facilitating the learning of sequential patterns.
Dropout layers are utilized for regularization to prevent overfitting.
Similar to the custom model, SGD optimizer with exponential decay learning rate and early stopping callback are employed.

5. LSTM Model:

The LSTM_model function constructs a Long Short-Term Memory (LSTM) model for sequence prediction.
LSTM layers are stacked to capture long-range dependencies in sequential data.
Dropout layers are inserted for regularization purposes.
The model is compiled with SGD optimizer, exponential decay learning rate, and early stopping callback.

6. CNN Model:

The CNN_model function creates a Convolutional Neural Network (CNN) model for sequence prediction.
It consists of Conv1D layers followed by max-pooling and flattening layers.
Dense layers with ReLU activation functions are added for further processing.
SGD optimizer with exponential decay learning rate and early stopping callback are utilized for optimization.

7. MLP Model:

The MLP_model function constructs a Multilayer Perceptron (MLP) model for sequence prediction.
It comprises multiple Dense layers with ReLU activation functions.
Dropout layers are incorporated for regularization.
Similar to other models, SGD optimizer with exponential decay learning rate and early stopping callback are employed.

8. Root Mean Squared Error (RMSE) Calculation:

The RMSE_Value function calculates the root mean squared error between the true target values and the predicted values.
It employs the mean_squared_error function from the scikit-learn library to compute the mean squared error, and then takes the square root to obtain the RMSE.

9. Predictions Plotting:

The PredictionsPlot function visualizes the true target values and the predicted values for comparison.
It generates a line plot showing the trend of true values and predicted values over time.

10. Model Evaluation:

Each model is trained and evaluated on the training and testing datasets.
Early stopping is applied to prevent overfitting and ensure optimal model performance.
RMSE values are calculated to quantify the prediction accuracy of each model.
Predictions are plotted to visually assess the performance of the models in capturing traffic flow patterns.

11. Results Analysis:

The RMSE values of all models are compared to identify the best performing model.
The model with the lowest RMSE value is considered the best model for traffic flow prediction.
A bar graph is plotted to visualize the RMSE values of different models for easy comparison.
Additionally, a tabular representation of model names and corresponding RMSE values is provided for comprehensive analysis.

V. CODE DOCUMENTATION:

This documentation provides an overview of the Python code for predicting traffic flow using various machine learning models. The code involves data preprocessing, model training, and prediction tasks.
Libraries Used
The following libraries are used in the code:
Pandas: For data manipulation and analysis.
NumPy: For numerical computations.
Matplotlib: For data visualization.
Seaborn: For statistical data visualization.
TensorFlow: For building and training deep learning models.
Data Preprocessing

Loading Data: The code loads a dataset containing traffic data, where the "DateTime" column is converted to datetime objects, and the "ID" column is dropped.

Feature Engineering: New features such as "Year," "Month," "Date_no," "Hour," and "Day" are added based on the "DateTime" column to provide additional context for the analysis.

Data Splitting: The dataset is split into training and testing sets using a specified ratio.

Model Definition:
The code defines multiple machine learning models for traffic flow prediction:
Custom Model: A custom neural network model with customizable layers.
GRU Model: A Gated Recurrent Unit (GRU) model for sequence prediction.
LSTM Model: A Long Short-Term Memory (LSTM) model for sequence prediction.
CNN Model: A Convolutional Neural Network (CNN) model for sequence prediction.
MLP Model: A Multi-Layer Perceptron (MLP) model for sequence prediction.

Model Training and Evaluation:
Each model is trained using the training data and evaluated using the testing data. The training process includes defining the model architecture, compiling it with a specified optimizer and loss function, fitting it to the training data, and early stopping to prevent overfitting. The model's performance is evaluated based on the mean squared error (MSE) loss metric.

Prediction:
After training, each model makes predictions on the testing data, and the predicted values are stored for further analysis.

VI. RESULT AND ANALYSIS

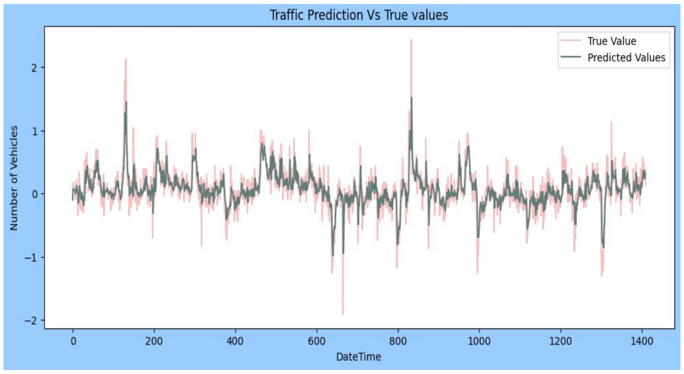
In this project, four different junctions was evaluated and the performance of various machine learning models for traffic flow prediction across Junctions were recorded. The root mean squared error (RMSE) values were used as the metric to assess the accuracy of each model.

Model Performance First Junction

model	RMSE
Custom Model	0.239748
GRU Model	0.249467
LSTM Model	0.273104
CNN Model	0.245254
MLP Model	0.246272

Best Model is the model with the minimum RMSE value is the Custom Model with an RMSE of 0.239748. Therefore, the Custom Model is considered the best model for traffic flow prediction at First Junction.
Fig 1 show the prediction for First Junction.

Fig1 Traffic prediction for First Junction.

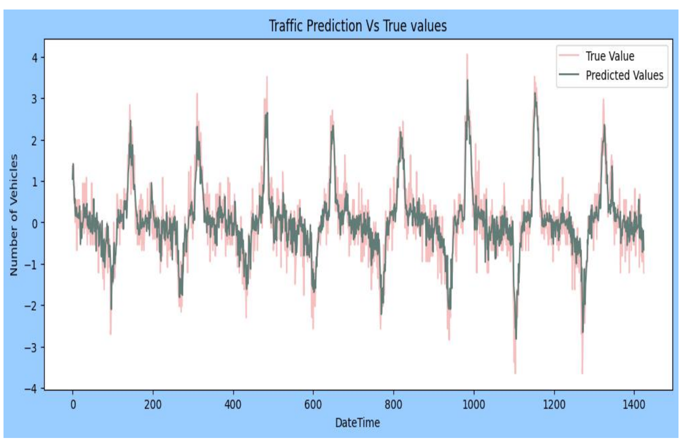


Model Performance Secon Junction

model	RMSE
Custom Model	0.473779
GRU Model	0.555820
LSTM Model	0.572657
CNN Model	0.547159
MLP Model	0.539472

Best Model is the model with the minimum RMSE value is the Custom Model with an RMSE of 0.473779. Therefore, the Custom Model is considered the best model for traffic flow prediction at Second Junction.

Fig2 Traffic prediction for Second Junction.

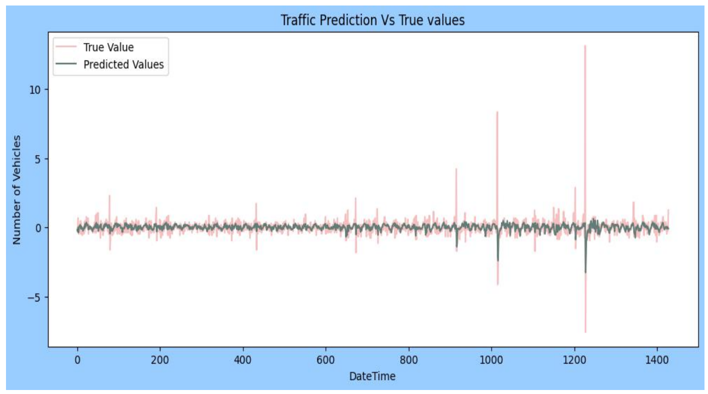


Model Performance Third Junction

model	RMSE
Custom Model	0.578800
GRU Model	0.606202
LSTM Model	0.620192
CNN Model	0.569653
MLP Model	0.558898

Best Model is the model with the minimum RMSE value is the MLP Model with an RMSE of 0.539472. Therefore, the MLP Model is considered the best model for traffic flow prediction at Third Junction.

Fig3 Traffic prediction for Third Junction.

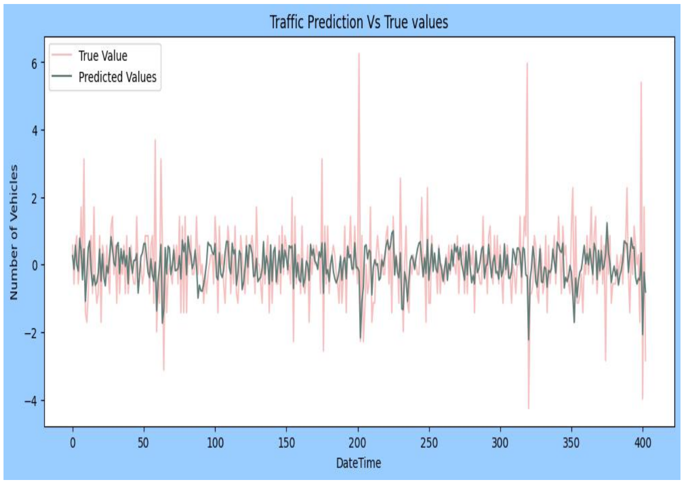


Model Performance Fourth Junction

model	RMSE
Custom Model	1.003844
GRU Model	0.993493
LSTM Model	1.096396
CNN Model	1.006663
MLP Model	0.989720

Best Model is the model with the minimum RMSE value is the MLP Model with an RMSE of 0.989720. Therefore, the MLP Model is considered the best model for traffic flow prediction at Fourth Junction.

Fig3 Traffic prediction for Fourth Junction.



VII. RESULT ANALYSIS

In this project, various machine learning models were trained and evaluated to predict traffic flow for four different junctions. The models included GRU, CNN, MLP, LSTM, and a custom model. Normalization and differencing transforms were applied to achieve stationary time series data, tailored to the unique characteristics of each junction.

A. Insights For First Junction.

First Junction exhibits a rapid increase in the number of vehicles compared to other junctions.
The traffic at Junction 1 displays strong weekly and hourly seasonality patterns.
The Custom Model achieved the best performance at First Junction, with an RMSE of 0.239748.

B. Insights For Second Junction.

Sparse data in Second Junction limits conclusive analysis.
The Custom Model outperformed other models, with an RMSE of 0.473779, making it the best model for Second Junction.

C. Insights For Third Junction.

Traffic at Third Junction shows a relatively linear trend.
The MLP Model exhibited the lowest RMSE value of 0.539472, making it the best model for Junction.

D. Insights For Fourth Junction.

Fourth Junction data presented challenges due to its sparse nature.
The MLP Model also performed the best at Junction 4, with an RMSE of 0.989720.

CONCLUSION AND FUTURE ENHANCEMENTS

Insights Gained:

Model Performance: Different machine learning models were applied to predict traffic flow at four junctions, revealing variations in performance across junctions and models.

Traffic Patterns: Analysis of traffic data uncovered insights into traffic trends, seasonality, and variability among junctions, providing valuable information for traffic management and planning.

Custom Model Effectiveness: The Custom Model demonstrated robust performance across junctions, indicating its adaptability and potential for accurate traffic prediction.

IMPACT OF THE PROJECT

Traffic Management: Accurate traffic prediction can aid in optimizing traffic flow, reducing congestion, and improving overall traffic management efficiency.

Resource Allocation: Insights gained from the project can inform resource allocation decisions, such as deploying traffic control measures and optimizing road infrastructure.

Urban Planning: Understanding traffic patterns helps in urban planning efforts, such as designing efficient transportation systems and mitigating traffic-related environmental impacts.

POTENTIAL FUTURE ENHANCEMENTS

Data Augmentation: Incorporating additional data sources such as weather conditions, events, and road conditions can enhance model accuracy and robustness.

Feature Engineering: Experimenting with different features and transformations can provide deeper insights into traffic patterns and improve prediction accuracy.

Ensemble Methods: Utilizing ensemble learning techniques to combine predictions from multiple models can potentially further enhance prediction performance.

Real-Time Prediction: Developing models capable of real-time traffic prediction can enable dynamic traffic management and adaptive control systems.

Integration with IoT: Integration with Internet of Things (IoT) devices and sensors can provide real-time data inputs for more accurate and responsive traffic prediction models.

In conclusion, this project sheds light on the effectiveness of machine learning models in predicting traffic flow and offers valuable insights for traffic management and urban planning. Continued efforts in model refinement, data integration, and real-time prediction can further enhance the impact and applicability of such projects in addressing traffic-related challenges in urban environments.

REFERENCES

[1]. Tan, M.-C., Wong, S. C., Xu, J.-M., Guan, Z. R., & Peng, Z. (2009). An aggregation approach to short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 10(1), 60–69.

[2]. Feng, J., & Sun, S. (2008). Neural network multitask learning for traffic flow forecasting. In Proceedings of the IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 1897, 1901).

[3]. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873.