Short-Term and Long-Term Traffic Flow Prediction in Dublin Using Deep Learning

MSc Research Project
MSc in Data Analytics

Gayathri Gangadharan
Student ID: X22203427

School of Computing
National College of Ireland

Supervisor: Aaloka Anant

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Gayathri Gangadharan
Student ID:	X22203427
Programme:	MSc in Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Aaloka Anant
Submission Due Date:	12/12/2024
Project Title:	Short-Term and Long-Term Traffic Flow Prediction in Dublin Using Deep Learning
Word Count:	7074
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Gayathri Gangadharan
Date:	29th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Short-Term and Long-Term Traffic Flow Prediction in Dublin Using Deep Learning

Gayathri Gangadharan
X22203427

Abstract

Efficient traffic control relies on accurate predictions of traffic distribution, particularly in highly interconnected cities like Dublin. This work assesses the effectiveness of Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs) for predicting traffic flow across short-term (hourly) and long-term (daily and monthly) intervals. The models were assessed using Dublin's traffic flow dataset with metrics such as R^2, RMSE, MSE, and MAE.The results revealed that ANN outperformed CNN in short-term (hourly) predictions due to its suitability for structured data. In contrast, CNN demonstrated superior performance in long-term (daily and monthly) predictions by effectively capturing temporal dependencies. However, both the models exhibited limitations in daily predictions.Additionally, regional analysis highlighted the sensitivity of the models to localized traffic dynamics, emphasizing the challenges in accurately simulating specific regional traffic behaviors.This paper explores the strengths and weaknesses of deep learning models for traffic forecasting in Dublin, providing valuable insights into their application for developing intelligent traffic systems.These findings contribute to a deeper understanding of the potential roles of ANN and CNN in enhancing smart traffic solutions for urban environments.

Keywords— **SCATS, Deep learning methods, Artificial Neural Networks, Convolutional Neural networks,Short-Term and Long-Term traffic forecasting, Traffic flow prediction.**

1 Introduction

Urban traffic management has become increasingly critical as cities like Dublin face rising congestion, inefficiencies, and environmental challenges. Dublin's road network exemplifies the complexities of managing growing traffic demand, necessitating innovative and adaptive solutions. This study investigates the integration of the Sydney Coordinated Adaptive Traffic System (SCATS) through deep learning techniques. SCATS, a widely used traffic control system, adjusts signal timings based on real-time traffic flow (McCann (2014)). By leveraging SCATS data and employing advanced neural network models, this research seeks to enhance predictive accuracy and develop smarter traffic management strategies.

Intelligent traffic control offers a robust solution to challenges such as traffic jams, pollution, and the exacerbated traffic issues caused by rapid urbanization. These difficulties are further compounded by increasing vehicle populations and road density, highlighting the need for efficient traffic control solutions. Among the various adaptive traffic

management systems,SCATS is a widely adopted example that adjusts signal phases in real-time. While SCATS has proven effective, its ability to respond flexibly to the dynamics of traffic congestion requires improvement to address the growing complexity and variability of traffic patterns.Machine learning techniques hold significant promise, leveraging traffic big data to solve complex modeling problems.Integrating neural network models with SCATS to enhance real-time traffic predictions contributes significantly to the development of intelligent transportation systems.
Dublin's growing traffic congestion significantly impacts daily commutes, urban productivity, and environmental sustainability. Addressing these challenges requires accurate traffic forecasting and adaptive traffic control systems.Combining SCATS with deep learning offers the potential to enhance traffic management by providing precise short-term (hourly) and long-term (daily and monthly) predictions.Previous studies, such as Medina-Salgado et al. (2022), have demonstrated the effectiveness of various machine learning and deep learning models in traffic forecasting but underscore the need for evaluating performance across varying temporal intervals. By bridging this gap, this study aims to advance Dublin's vision for smarter and more sustainable city infrastructure.

Research Question: How do short-term (hourly) and long-term (daily and monthly) prediction intervals impact the accuracy and reliability of deep learning models in predicting traffic flow in Dublin, both at a city-wide scale and within specific regions within the city?
Research Objectives: The primary objectives of this study are:

- To analyze the impact of short-term and long-term prediction intervals (hourly, daily and monthly) on the accuracy of deep learning models.
- To evaluate the effectiveness of Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs) in forecasting traffic flow using Dublin's traffic dataset.
- To explore the applicability of this research in urban planning scenarios, such as predicting traffic flow during events at specific locations in the city and at different times, by leveraging actual traffic flow data and applying models like ANN and CNN across various time intervals.

The study provides data-driven insights to aid urban planners in optimizing traffic management and developing sustainable city infrastructures.
The thesis structure is as follows: Section 2 discusses the Related Work, followed by the Methodology in Section 3. Section 4 outlines the design specification, while Section 5 focuses on Implementation. The evaluation is discussed in Section 6, and the report concludes with Section 7, which includes the Conclusion and Future Work.

2 Related Work

As urban centers expand and traffic volume increases, effective traffic management becomes vital for maintaining mobility, reducing congestion, and ensuring environmental sustainability.Traditional methods struggle with the dynamic, non-linear nature of modern traffic systems.Deep learning has emerged as a powerful solution, addressing both short-term and long-term forecasting challenges. This review categorizes the research into four key areas as follows:

2.1 Deep Learning Approaches in Traffic Forecasting

Recent progress in deep learning has significantly enhanced the accuracy of traffic flow predictions by effectively modeling intricate temporal patterns in traffic data.For instance, Shao and Soong (2016) demonstrated the effectiveness of Long Short-Term Memory Networks(LSTMs) for short-term traffic forecasting, achieving a mean absolute percentage error(MAPE) of just 5.4% on PeMS datasets, outperforming models like SVR, WNN, and SAE.The research also emphasized the drawbacks of conventional methods like ARIMA and Kalman filtering, which struggle to address the non-linear and stochastic nature of traffic flows.While earlier non-parametric models like SOMs and K-NNs showed moderate success, deep learning(DL) models, including LSTMs and DBNs proved superior by capturing abstract data representations.The author also proposed integrating data from adjacent roadways to enhance LSTM performance, emphasizing the potential of deep learning in advancing intelligent transportation systems for urban environments.

Building on prior work in traffic forecasting with deep learning models,Fouladgar et al. (2017) developed a decentralized deep learning architecture for real-time traffic congestion prediction.Their approach uses local measurements and congestion levels from nearby stations, eliminating the need for historical data and enhancing scalability.A regularized loss function was employed to prioritize high-congestion samples, addressing dataset imbalances. This decentralized method provides a real-time feedback, making it suitable for new traffic station installations.Validated using traffic flow datasets from Northern California, the model demonstrated its effectiveness in optimizing real-time traffic simulation and congestion estimation in urban contexts.These findings align with Shao and Soong (2016) work, highlighting the potential of deep learning in advancing intelligent transportation systems.

To improve traffic forecasting accuracy, Ta et al. (2022) proposed the Ada-STNet(Adaptive spatio-temporal graph neural network) model, which enhances Spatio-Temporal Graph Neural Networks(ST-GNNs) by addressing their limitations. Unlike ST-GNNs, which rely on fixed graph structures, Ada-STNet dynamically learns graph topology using node attributes, capturing both macro-level and micro-level spatial dependencies. This approach better represents complex traffic networks and enhances forecasting accuracy.The proposed model, with a spatio-temporal convolutional layer and two-stage training, outperformed existing models in real-world tests, excelling in normal and peak traffic conditions, making it promising solution for intelligent transportation systems.

2.2 A Review on Short-Term and Long-Term Traffic Prediction

Short-term traffic prediction(STTP) has garnered significant interest due to the growth of traffic data and advancements in Deep Neural Networks(DNNs).While traditional methods like ARIMA struggle to identify complex temporal and spatial interrelations, DNNs excel in capturing these patterns to predict future traffic conditions.Recent research by **?** surveyed DNN-based methods for STTP and emphasized the importance of structuring inputs to model spatial and temporal relationships effectively. Graph based models are ideal for representing traffic networks, while grid-based models are often used in human movement prediction. The study reviewed techniques such as Restricted Boltzmann Machines and graph-based networks, highlighting their reliance on inductive biases like locality and temporal continuity.However, a major challenge in the field is the lack of standardized benchmark datasets for method evaluation, which limits comparisons between approaches.While STTP target short-term predictions, long-term forecasting of seasonal

trends also benefits from DNNs.
Accurate long-term traffic density forecasting is crucial for addressing fluctuations caused by increasing vehicle populations and urbanization.**?** addressed this by developing the W-CNN-LSTM model, which combines wavelet wavelet decomposition with CNN and LSTM networks for improved day-ahead predictions.Wavelet decomposition separates traffic data into high and low frequency components, allowing the model to better capture long-term trends.Tested on an England traffic dataset, the W-CNN-LSTM outperformed ARIMA, LSTM, CNN, and MLP models, demonstrating superior accuracy in forecasting both fluctuations and trends.This integration of wavelet decomposition and deep learning represents a significant advancement, offering a reliable approach for long-term traffic forecasting essential for urban transportation planning and congestion management.
? proposed a DBN(Deep Belief Network)-based ensemble model for short-term traffic flow prediction, combining ensemble Empirical Mode Decomposition (EEMD) for data decomposition and Minimum Redundancy Maximum Relevance (MRMR) for feature selection.Each component is trained with DBN, and the forecasts are integrated to produce the final prediction. This approach outperformed traditional methods, including single DBNs, by effectively addressing the non-linear nature of traffic data.Validation on real-world datasets confirmed its superior accuracy for short-term forecasting.Future research could explore LSTM-based ensembles or optimization strategies to adapt the model to more complex road networks.

2.3 Advancement and Challenges in SCATS Integration With Deep Learning for Traffic Management

The integration of SCATS with deep learning has the potential to improve urban traffic management through real-time signal control and predictive algorithms, as noted by Panda and Nguyen (2016).However, challenges include the spatio-temporal complexity of SCATS data, irregular traffic patterns, noise, incomplete data, and significant computational demands.Scalability issues also arise, with risks of losing spatial relationships in large networks.Additionally, random events like accidents, and public gatherings remain difficult to predict, and real-time synchronization with SCATS is technically challenging.Improvements could include advanced neural architectures like LSTMs or GRUs, hybrid models, automated traffic data clustering, and IOT-based data acquisition for higher accuracy.Incorporating contextual factors like weather or event data can further enhance model robustness, paving the way for more intelligent and adaptive traffic systems.
Xu et al. (2019) highlighted the importance of traffic forecasting in intelligent transport systems, particularly under SCATS.Traditional methods like ARIMA, Kalman Filters, KNN, and SVM struggle with the dynamic, non-stationary, and spatio-temporal nature of traffic data. While LSTMs and GRus better capture temporal features, they fall short in identifying spatial relationships critical to urban traffic patterns.To address this Xu et al. (2019) developed the Graph Embedding Recurrent Neural Network(GERNN), combining Deep-walk for spatial features and LSTM for temporal dependencies. Tested on Hangzhou SCATS data, GERNN improved RMSE and MAE by 19-25%.Future work includes enhancing architectures, tuning parameters, and scaling GERNN for broader SCATS datasets.

2.4 A comparison of Model Performances in Predicting Traffic Flow Using Deep Learning Methods

Related works	Methods	Findings	Limitations
"A Deep Learning Approach for Long-Term Traffic Flow Prediction With Multi-factor Fusion Using Spatiotemporal Graph Convolutional Network." Qi et al. (2022)	Spatiotemporal Graph Convolutional Networks(STGCN), GRU, Temporal Graph Convolutional network(T-GCN), LSTM, Attention-based Spatiotemporal Graph Convolutional Networks(ASTGCN), Adaptive Graph Convolutional Recurrent Networks(AGCRN)	The proposed STGCN model excels in prediction, achieving up to 43.27% RMSE and 50.47% MAE reductions, with R^2 value of 0.96,further enhanced by integrating meteorological factors, which reduce RMSE by 5.82% and MAE by 8.28%.	The proposed deep learning method is limited by its reliance on monitoring point data with clear location, computationally expensive fully connected graphs for larger datasets, and a narrow focus on meteorological factors while neglecting influences like holidays or traffic accidents.
"Deep learning for short-term traffic flow prediction" Polson and Sokolov (2017)	Sparse Vector Auto-Regressive(VARM8L), Deep learning model(DLM8L, using sparse predictors and median filtering preprocessing), One-Layer Neural Network(NNM8L)	DL model DLM8L outperformed both VARM8L and NNM8L, achieving higher accuracy(R^2:0.87in-sample, 0.85 out-of-sample) and lower MSE (6.32 in-sample, 6.54 out-of-sample).	The paper highlights limitations in the DL model's low interpretability, dependence on effective data preprocessing, and reduced accuracy without external predictors like weather during non-current events.
"Short-term traffic flow prediction with Conv-LSTM" Liu et al. (2017)	ARIMS, SVM, K-NN, SVR, Stacked Denoising Autoencoder (SAE), Deep Belief Network (DBN), LSTM,CNN-LSTM, And proposed models: Conv-LSTM and Bi-directional LSTM (Bi-LSTM)	The Conv-LSTM model, enhanced by Bi-LSTM, outperforms CNN-LSTM and traditional methods, achieving the highest accuracy for urban and freeway short-term predictions with RMSE 6.419	The limitations include reliance on high-quality input data, which may not always be available, and the computational cost, particularly in real-time applications.Also, its performance may suffer from insufficient or imbalanced data during unusual traffic events.
"Attention based spatiotemporal graph attention networks for traffic flow forecasting" Wang et al. (2022)	Baseline models (HA, ARIMA, VAR) and DL models (LSTM, GRU, STGCN, GeoMan, ASTGCN). And the proposed model Attention-Based Spatiotemporal Graph Attention Network (ASTGAT)	The proposed model outperformed other models, reducing RMSE by 7.4% and MAE by 5.8%, while addressing over-smoothing and network degradation for improved medium and long-term traffic prediction accuracy	The ASTGAT model lacks validation for multi-scale information, broader utility beyond traffic flow predictions, and generalization to datasets with differing characteristics.

Table 1: A comparison of Model performance in traffic flow prediction

2.5 Conclusion

This review explores the potential for further advancements in deep learning (DL) to enhance short-term and long-term traffic flow prediction. Deep learning models like LSTM and GRU's have demonstrated superior performance compared to simple baselines, while ST-GNNs have also shown better results than traditional methods. However, most previous research has focused on single-interval forecasts, such as short-term or long-term traffic predictions, leaving a gap in models capable of having multiple intervals like hourly, daily, and monthly traffic forecasting. The purpose of this project is to address this gap by proposing deep learning models that accurately estimate traffic flow across different intervals, providing valuable insights to urban traffic departments.
Despite these advancements, integrating deep learning with systems like SCATS presents challenges, including computational complexity, data sparsity, and real-time processing. Future works should prioritize hybrid models, enhancing scalability, and incorporating external parameters such as weather conditions and public events to refine prediction accuracy. Addressing these challenges will facilitate the development of robust traffic management architectures capable of adapting to the demands of modern cities.

3 Methodology

The process of steps adopted in this study to apply and incorporate the SCATS data into a deep learning model to forecast the traffic flow in Dublin is outlined in this section. The CRISP-DM is used to follow a model approach to deal with raw traffic data to gain valuable insights out of these data that will help to model accurately and effectively. There are six potentially six distinctive parts of the proposed methodological framework as depicted in Figure 1. These stages are designed specifically for incorporating SCATS data with the deep learning model to enhance an accurate and efficient traffic flow prediction system.

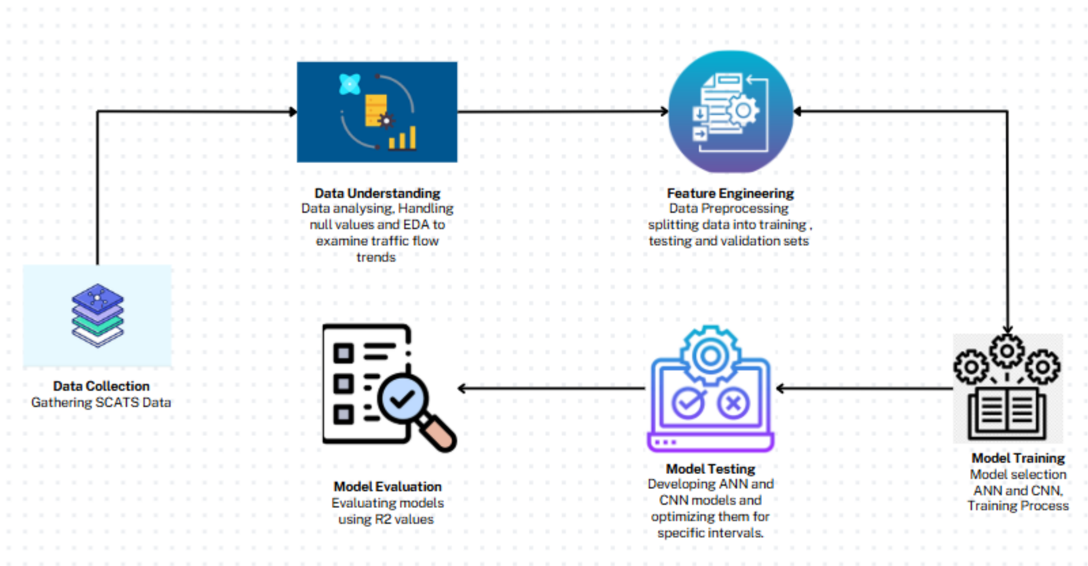

Figure 1: Methodology Flow Chart

3.1 Data Understanding

3.1.1 Data Overview

The dataset utilized in this study was sourced from the public platform Smart Dublin, maintained by the four Dublin Local Authorities. It provides historical traffic flow records for Dublin, accessed through the link: `https://data.smartdublin.ie/dataset/dcc-scats-detector-volume-jul-dec-2024`.

The SCATS traffic volumes dataset contains monthly traffic counts recorded by detectors located at junctions, used for traffic signal control. These detectors also measure traffic volumes approaching a junction, making the data suitable for analyzing vehicle movement trends by focusing on key junctions that reflect overall traffic flows.

The July 2024 dataset was chosen for this study because it is the latest fully available dataset on the website. The dataset comprises 11,046,124 rows and 9 columns. Key features include End Time (indicating the conclusion of each hourly count period), Region (detector site location, such as North City, South City, West City etc.), Site (mapped to SCATS site files for precise locations), Detector(Sensor identifiers at each site), Sum Volume (total hourly traffic volume), and Avg Volume (average traffic volume calculated over 5-minute intervals in the preceding hour).

3.1.2 Data Preparation and Analysis

The data was loaded using Pandas library in Python. The first and last five rows of the dataset were viewed to inspect its structure. A summary of the dataset was analyzed to understand key statistics. Upon examining for null values, it was discovered that the last three columns ('Weighted_Avg','Weighted_Var', and 'Weighted_Std_Dev') contained only null values. These columns were removed using the drop function from Pandas library for a cleaner dataset. The next process is to transform the end time present in the 'End_Time' column into a 'DateTime' format to handle the data within a period more effectively. Other time segments including date, hour and the name of day of the week are derived from this 'datetime' object. This transformation helps to better analyze the dataset and improves the visualization of the data.

3.1.3 Exploratory Data Analysis

The graph in figure 2 represents the total sum of traffic volume across different regions and dates within the month, plotted as a line plot created using the Seaborn library .The traffic density demonstrate that Region 'IRE' consistently accounted for the largest traffic loads, with peak figures exceeding four million vehicles, whereas Region 'IRE 3' consistently recorded the lowest traffic loads, which did not exceed 0.5 million. The trends observed are quite different with traffic rising sometimes with a constant rise in the regions and clearly indicating troughs at the middle and end of the week. Regions like 'DCC1', 'CCITY', 'SCITY', and 'WCITY' exhibit similar traffic patterns with a moderate cluster of around 1.5 million to 2.5 million in traffic. In summary, these findings give an understanding of the traffic variation between the regions.

The second visualization shown in figure 3, represents the total number of vehicles per region throughout each hour of the day. This analysis highlights that the busiest hours for traffic in all regions consistently occur between 9:00 AM and 7:00 PM. These peak activity periods generally correspond to the daily commute period and probably the working hours. This information is useful for establishing the fact that the high traffic

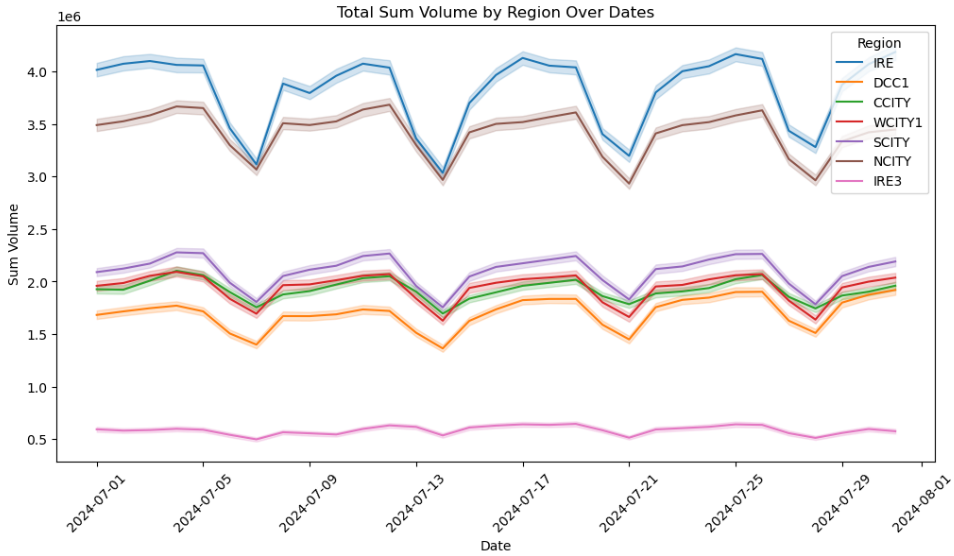

Figure 2: Sum of vehicles by Region Over dates

volume occurs during the day, compared to early morning and especially the late-night hours.

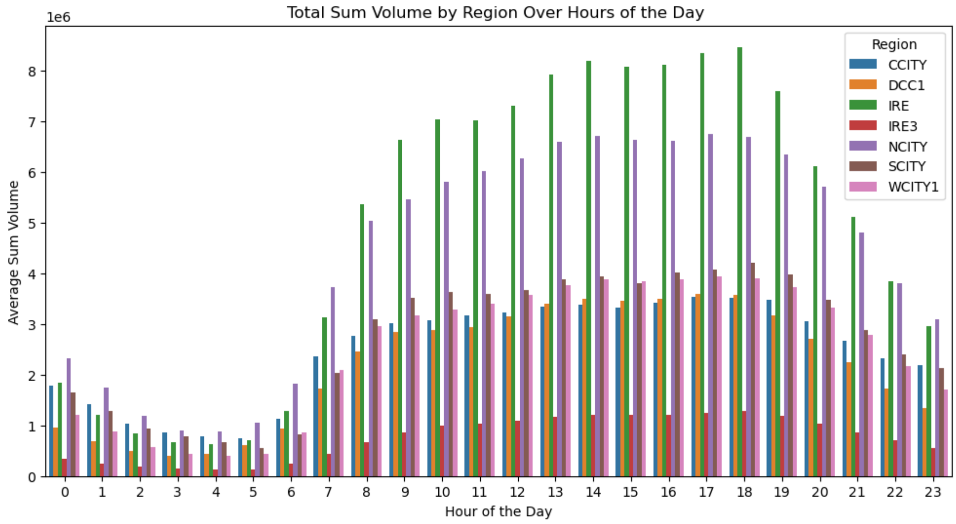

Figure 3: Sum of vehicles by Region Over hours

There are consistent daily patterns of traffic flow across certain locations, driven by the routines of daily activities. These systematic variations can be segmented into distinct time-based components, which are effectively visualized in the graph. To analyze these patterns, the traffic volume data are grouped into two categories: weekdays and weekends, as shown in figure 4. The plot reveals a notable difference in traffic behavior between these 2 periods.

Weekdays: The morning commute (8 AM - 11 AM) and the evening commute (4 PM - 7 PM) show a sharp increase in traffic. Work-related travel is responsible for these peaks. After 9 PM, traffic flow declines steeply, indicating a reduction in activity during late night hours.

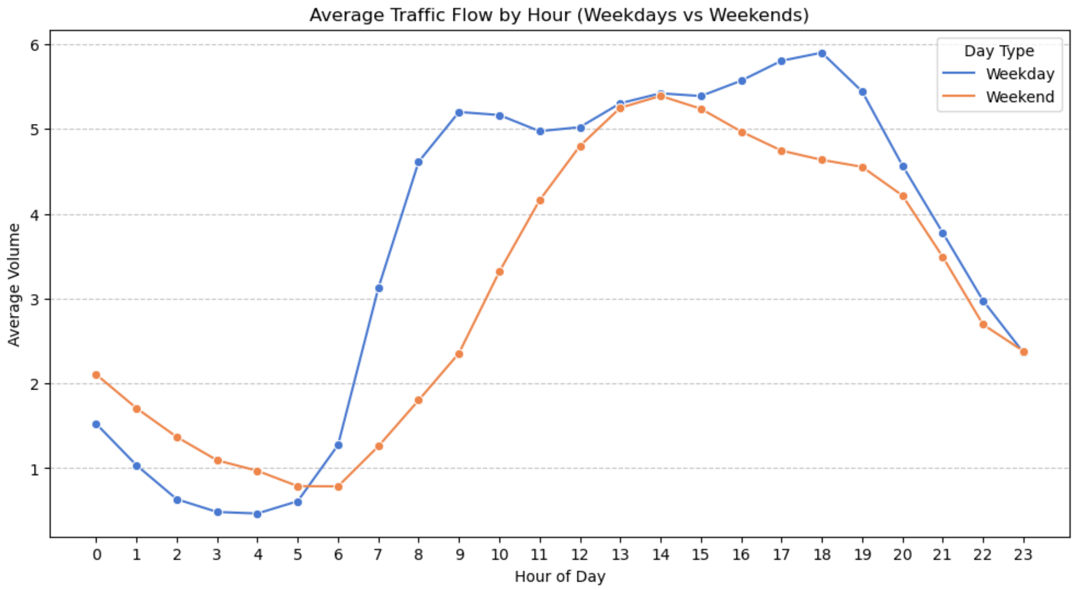

Figure 4: Weekdays vs Weekends traffic flow by hours

Weekends: There are more gradual changes in traffic patterns with a steady rise in the late morning building sharp peaks in the afternoon, followed by a gradual decline in the evening. These trends are due to personal activities, not work-related commutes. Through segmentation and comparative analysis, this analysis provides valuable insights into how traffic flow dynamics change throughout the week.

The traffic flow analysis in figure 5 shows that traffic flow differs on different days of the week. From this analysis we can see that Mondays, Tuesdays and Wednesdays show the highest morning peaks, and Saturdays and Sundays show the lowest morning peaks. All weekdays' peak traffic levels stay the same in the afternoon. Mondays, Tuesdays, and Wednesdays also have noticeable evening peak traffic hours compared to other days.

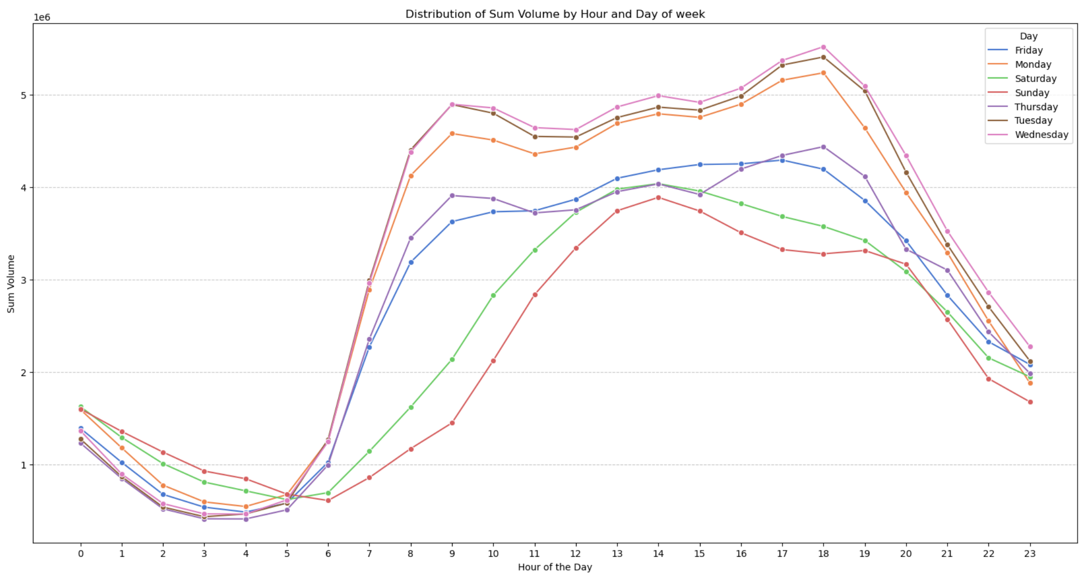

Figure 5: Total number of vehicles by hour and Day of week

3.1.4 Congestion Analysis and Visualization

In this phase, the traffic data is analyzed for periods of congestion in terms of predefined traffic volume thresholds. These thresholds are derived from statistical analysis of metrics, ‘Sum_volume’ and ‘Avg_volume’, and comparing them to statistical measures, such as percentiles and standard deviations. When these thresholds are exceeded, the traffic is classified as congested and labeled as 1, and non-congested traffic is labeled as 0. The aggregated resulting congestion data undergoes an examination of variation over different hours of day and regions. Those aggregations give directions towards times and places of congestion. In practical terms, plotting the data into a ‘heatmap’ as shown in figure 6 is an effective way to get an overview of potential patterns of congestion and identify specific times and locations that require the most attention.

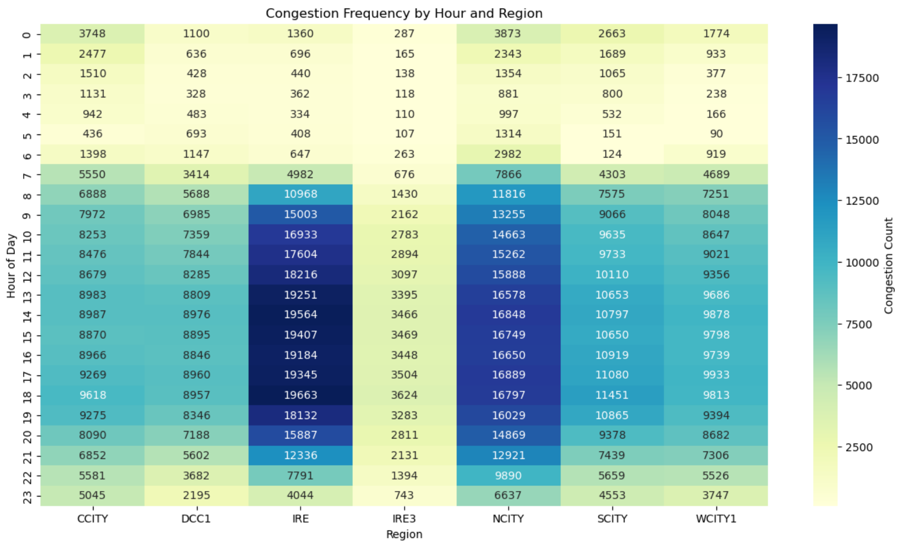

Figure 6: Heatmap of Congestion frequency

3.2 Feature Engineering

3.2.1 Label Encoding

In this case, the ‘LabelEncoder()’ function from the ‘Scikit-learn’ library is used to convert the categorical variables to numerical variables. For this specific case, the ‘Region’ column, which contain unique categories, are encoded using the encoder so that their values become integers. It is important to have numeric inputs because deep learning and machine learning models require them. The ‘fit_transform()’ function first trains the encoder on the unique values in the columns and then assigns numerical labels to those values based on the unique values it identified. This step primarily aims to transform the categorical data so that the model can successfully handle the transformed data.

3.2.2 Data Splitting

In this step, the data is divided into Train and Test using the ‘train_test_split()’ function from ‘Scikit-learn’ library.This is a very important step in machine learning as it provides a means for model validation and evaluation using the test dataset. Using 80% of the data for training allows the model to learn patterns, while the remaining 20% serves as a test set to measure how well the model performs on unseen data.This step makes the validation process more credible, avoids over-fitting, and promotes better generalization to new inputs.

3.2.3 Data Scaling

The next step is data scaling, which incorporates both the target variable and features. The 'StandardScaler' class from the 'Scikit-learn' library is used for scaling the features. Scaling is an essential step in deep learning and machine learning methods to reduce the impact of features with exceptionally large scales, such as those found in traffic dataset.

3.3 Model Training and Evaluation

3.3.1 Model Training

Two deep learning techniques, namely Artificial Neural Network (ANN) and a Convolutional Neural Network (CNN), were used to efficiently capture the underlying patterns in the data. These models were employed to forecast traffic flow across hourly, daily, and monthly time-frames, offering valuable insights into temporal variations. Additionally, a specific region from the dataset was chosen for this analysis, enabling accurate traffic flow predictions for hourly, daily, and monthly intervals.

- ANN: The ANN was chosen because its layers of connected neurons can effectively learn complex pattern and nonlinear relationship within datasets, which make them more appropriate in predicting traffic flow over different time intervals such as hourly, daily, and monthly forecasts. Because of the architecture of the ANN model, the model is capable of studying various patterns and dependencies contained in traffic data and various traffic conditions. For example, Kranti Kumar and Katiyar (2015) demonstrated the application of ANNs for short-term traffic flow prediction, relying on historical traffic information. This study shows that ANNs are particularly suitable for the analysis of traffic trends by being real-time and high-performance solutions that can be used to support next generation traffic management systems that seek to address traffic congestion and improve mobility.

- CNN: Convolutional neural networks are highly efficient at capturing spatial and temporal dependencies in traffic patterns and are flexible for traffic flow prediction across different time intervals, such as hourly, daily, and monthly. CNNs extract many key features using convolutional and pooling layers to identify spatial and temporal patterns in the data. As illustrated by Agafonov (2020), CNN-based methods, such as graph convolutional networks, are pivotal for traffic flow prediction. The research compared the architectures of graph convolutional networks with respect to traffic density predictions, incorporating daily and weekly trends.

3.3.2 Evaluation Metrics

The performance of the traffic flow prediction models was assessed using the metrics, R^2 score, Mean Absolute Error(MAE), Mean Squared Error(MSE) and Root Mean squared Error (RMSE) to ensure both accuracy and reliability(Duan et al. (2016)). These evaluation metrics are implemented using the 'Scikit-learn' library.

R^2 **Score**:The R^2 score is a metric used to assess the predictive performance of a model, reflecting how closely the predictions match the actual data. It quantifies the proportion of variance in the target variable explained by the model's predictions. This score was computed using the 'R^2_score' function from the 'Scikit-learn' library and served as a primary evaluation metric due to its ability to clearly indicate the model's effectiveness in capturing traffic flow patterns.
MAE: This error measures the average magnitude of prediction errors, providing a straightforward and easily interpretable evaluation.
It is calculated using the 'mean_absolute_error' function from the 'Scikit-learn' library.
MSE : MSE represents the average squared differences between predicted and actual values.By squaring this errors, it assigns greater weight to larger discrepancies, making it useful for identifying significant prediction errors. This metric was calculated using the 'mean_squared_error' function from the 'Scikit-learn' library.
RMSE: RMSE is the square root of MSE, providing error measurements in the same units as the predicted variable, enhancing interpretability. It is calculated using the 'NumPy' library.

4 Design Specification

The design specification (refer to fig 7) relates to the design and implementation of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) for predicting traffic flow, as supported by studies such as (Kareem et al. (2021)).The design starts with traffic flow dataset sourced from SCATS detectors, which is analyzed and preprocessed using various Python libraries, such as Pandas for Data preparation, NumPy for numerical operations, and Seaborn for visualization and traffic trends exploration. The prediction framework, Using ANN and CNN models developed in Python with the libraries TensorFlow and Keras, was designed to forecast traffic volumes at hourly, daily and monthly intervals.

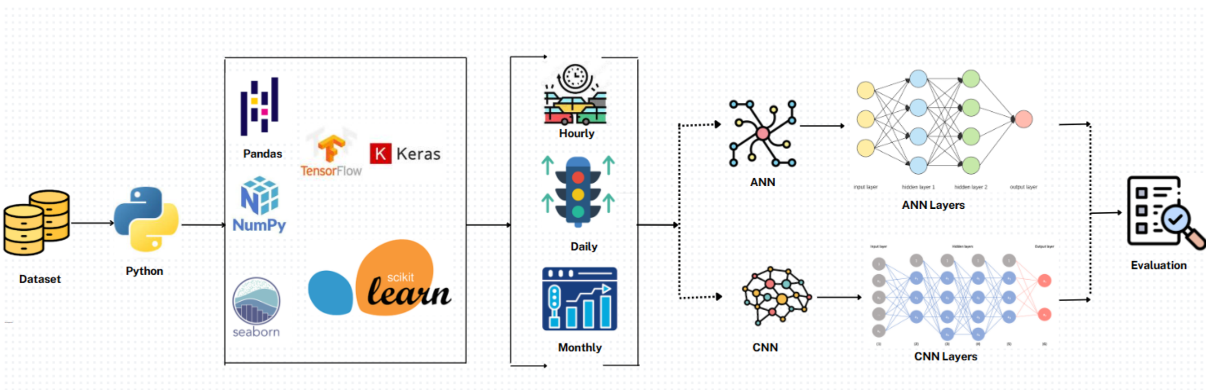

Figure 7: Design Specification Diagram

The ANN architecture comprises multiple dense layers utilizing 'ReLu' activation functions to introduce non-linearity, enabling the model to learn intricate patterns. Additionally, dropout layers are incorporated to reduce over-fitting, making it highly effective at capturing complex, nonlinear relationships within the data. The model was built using the 'TensorFlow.Keras.Sequential' API, which provides a straightforward way to define the network layer by layer, and trained using the 'Adam' optimizer, an algorithm that ensures efficient and stable updates to the model's parameters for faster convergence.

To accommodate the input requirements of CNN, the data was reshaped into a three-dimensional structure using the 'x_reshape' function, where the final dimension represented a single feature. CNN uses convolutional layers that apply a set of filters to identify temporal and spatial relationships in the input data, extracting essential features. Max-pooling layers then reduce the dimensionality of the data, lowering computational demands while preserving critical information. The pooling layer output is converted into a one-dimensional array using flatten layers to enable dense layers to analyze the extracted features for prediction. When combined with ReLU activation and dropout, dense layers enhance the model's ability to effectively process and generalize the data.
To provide flexibility and accuracy across various time scales, both the models are trained to forecast traffic flow at hourly, daily and monthly intervals.

One of the Key milestones of this work was the comparison of the ANN and CNN models, focusing on specific regions within the dataset. Both models were designed to use the same intervals, allowing for a detailed examination of their predictive capabilities at varying time scales. For both models, the dataset was divided based on a specific region, allowing the models to make localized predictions that accurately represent the unique traffic patterns and behaviors in those areas. With region-specific data availability, the resulting models were able to capture variations in traffic flow that are specific to various locations, such as differences in road usage, congestion levels, and traffic signal timings.
Every model was evaluated using metrics such as R^2, Mean Absolute Error (MAE), Mean squared Error (MSE), and Root Mean squared Error (RMSE), which were used to analyze the model's performance, providing a comprehensive comparison of the model's effectiveness.
The design successfully implemented ANN and CNN models for predicting traffic flow, highlighting their ability to generate accurate and localized forecasts. By incorporating region- specific data and comprehensive preprocessing, the models adapted to varying traffic patterns, offering practical insights for improving traffic management and reducing congestion.

5 Implementation

This section details the use of Artificial Neural Networks (ANN) and Convolutional Neural Network (CNN) (Çetiner et al. (2010)) for predicting traffic flow across three-time intervals: monthly, daily, and hourly. The models were developed using Python and TensorFlow, with comprehensive preprocessing applied to the dataset to enhance prediction accuracy.

5.1 Monthly Prediction

5.1.1 Implementation of ANN

An ANN model was designed to forecast traffic flow using aggregated monthly data. The architecture included an input layer with 64 neurons and ReLu activation, followed by a hidden layer with 32 neurons and ReLu activation. Dropout layers with a 20% rate were added to prevent over-fitting. A single output neuron was included to handle the regression task. The model was trained using the Adam Optimizer, with MSE as the loss function and MAE as the evaluation metric. The model was built using tensorflow.keras.Sequential function and trained on the monthly aggregated dataset for 10 epochs, with a batch size of 32 and 20% validation split. It was evaluated on the test dataset, producing predictions and performance metrics that provided valuable insights to the model's ability to forecast monthly traffic volumes accurately.

5.1.2 Implementation of CNN

A convolutional Neural Network was implemented using the TensorFlow and Keras libraries to model traffic flow. The architecture featured a convolutional layer with 32 filters, a kernel size of 2, and a ReLU activation, designed to capture spatial and temporal patterns in the traffic data. This was followed by a max-pooling layer with a pool size of 2. The output from these layers were flattened and fed into a dense layer with 64 neurons and ReLU activation, accompanied by a dropout layer with a 20% rate. The final layer comprised a single neuron for regression, enabling traffic flow value prediction. The model was compiled with Adam optimizer and the training was conducted with a batch size of 32 for 10 epochs, while validation data was used to monitor performance.

5.2 Daily Prediction

For daily traffic flow prediction, the data was aggregated by grouping it based on attributes such as region, site, detector, year, month, and day. The resulting dataset included features like 'Avg_volume', representing the mean traffic volume and the 'Sum_volume', representing the cumulative traffic volume for each day. To prepare the data for modeling, it was split into training and testing sets using the 'train_test_split' function. To ensure uniform input values, both target variable and features were normalized.

5.2.1 Implementation of ANN

The ANN model was developed to analyze data aggregated at a daily level. The target variable was the total traffic volume (Sum_Volume), while the dataset included features such as region, site, detector,and temporal attributes like year,month, and day.The ANN architecture comprised an input layer with 64 neurons, a hidden layer with 32 neurons and dropout layers to mitigate overfitting. The model was compiled using the Adam optimizer, with MSE as the loss function and MAE as the evaluation metric. The model was trained for 10 epochs and after training, predictions were made on the test set and rescaled back to their original scale for better interpretation.

5.2.2 Implementation of CNN

The CNN model for daily data followed the same architecture and implementation approach as described in the CNN model for monthly prediction.It began with a convolutional layer of same filters and size and ReLU activation.The model included a max-pooling layer, followed by a flattening, a dense layer, and a dropout layer, with a final layer for regression.It was compiled with Adam optimizer and trained for 10 epochs with validation data to assess performance.

5.3 Hourly prediction

The data was aggregated for hourly prediction by grouping it based on Region, Site, detectors, and time attributes such as year, month, day, and hour. To ensure the consistency and enhance model performance, feature scaling was applied.

5.3.1 Implementation of ANN

The architecture of ANN model, comprising multiple dense layers with ReLU activation and dropout for regularization, was the same that used form monthly predictions. The model was trained for 10 epochs with a batch size of 32, using the test dataset for validation. Following training, the model was evaluated on the test set to generate hourly traffic flow predictions.

5.3.2 Implementation of CNN

To maintain consistency and scalability across various time periods, the CNN model for hourly forecasts used the same architecture as the models created for daily and monthly predictions. By leveraging its robust architecture, the CNN effectively captured the intricate patterns present in hourly traffic data. This implementation will provide valuable insights such as peak and non-peak hours.

5.4 Traffic Flow Prediction in a Specific Region: 'IRE'

ANN and CNN models were created for a specific region 'IRE', selected from the seven region's in the dataset, to predict traffic flow at hourly, daily and monthly intervals. The region 'IRE' was filtered from the dataset, with the columns divided into features (Region, site, detector, year, month, day, and hour) and the target variable(Sum_volume).

5.4.1 Implementation of ANN

ANN models were developed for region 'IRE' using TensorFlow and Keras to predict traffic flow on a monthly, daily, and hourly basis.
The target variable, 'Sum_Volume', was forecasted using features such as region, site, detector, year, month, day, and hour for monthly predictions. The model, comprising dense layers with ReLU activation and dropout, was trained for 10 epochs and produced accurate predictions.
Similarly, daily predictions, based on factors like average volume and day- specific attributes, were trained over 20 epochs, yielding precise results.
For hourly forecasts, the model, with dense layers of 64 and 32 neurons, was trained for 10 epochs to hourly aggregated data.

5.4.2 Implementation of CNN

For monthly prediction, the data was scaled, and reshaped to meet CNN input requirements, and processed through convolutional, max pooling, flattening, and dense layers with ReLU activation and dropout for regularization. Trained for 10 epochs with the Adam optimizer and MSE loss, the model produced accurate monthly predictions after rescaling.
Similarly, for hourly predictions, the model applied the same architecture to aggregated

data. The model was trained for 10 epochs with a batch size of 32 and the predictions are made.
For daily predictions, the preprocessing steps were consistent, with input features including region, site, detector, average volume, and day. After 20 epochs of training, the model delivered precise daily forecasts.

6 Evaluation

This section represents an in-depth assessment of the ANN and CNN model, emphasizing their capability to predict traffic flow across different intervals: hourly, daily, and monthly. Furthermore, the analysis also specifically targets a specific region, referred to as 'IRE', and evaluated the model's prediction efficiency at a regional level. To ensure a thorough evaluation of the model's effectiveness, standard performance metrics from the Scikit-learn library were employed, including Mean Absolute Error(MAE), Mean Squared Error(MSE), Root Mean Squared Error(RMSE), and R-squared (R^2).These metrics were selected to provide a well-rounded understanding of the model's accuracy, both absolute and relative, and to assess the overall quality of the model's fit to the data. To check the presence of overfitting, I compared the model's R^2 values on the training and test sets. The R^2 value represents the degree to which the actual data is fitted by the model, thus showing how valid the model is as a predictor of variability in the dependent variable. The model showed high accuracy on both training and testing sets. The small difference between the accuracies indicates that the model generalizes effectively without significant overfitting.

6.1 Experiment 1: Monthly Prediction

The ANN model for monthly predictions excelled in delivering accurate monthly predictions, achieving an R^2 of 0.9918, which highlights its ability to capture a significant proportion of the variance in the data. More than that, the model highlighted remarkable precision, as reflected in its performance metrics: a Mean Absolute Error of 2.8964, a Mean Squared Error of 89.6810, and Root Mean Squared Error of 9.47. These results emphasize the model's strength in generating reliable and accurate forecasts, making it a valuable tool for predictive analysis.
The CNN model demonstrates outstanding performance in monthly traffic flow predictions, as evidenced by its evaluation metrics. A mean Absolute Error (MAE) of 2.0538 indicates that, on average, the model's prediction differs from the actual values by just units, which is exceptionally low relative to the data's scale. The MSE of 54.2533 and RMSE of 7.3657 further validate the model's accuracy, with the RMSE showing that even larger errors are minimal. Additionally, the R squared value of 0.9951 confirms that the model accounts for 99.51% of the variance in the data, leaving only a small fraction unexplained. These results highlight the model's ability to effectively capture the underlying patterns in the data, making it a reliable tool for precise monthly predictions, particularly in applications such as traffic management.

6.2 Experiment 2: Daily Prediction

The Artificial Neural Network model captures daily traffic flow trends, but it struggles to achieve the same level of accuracy as the CNN model for data aggregated over larger timeframes. This is evident in the evaluation metrics, with the ANN recording a higher Mean Absolute Error (MAE) of 94.5117 and a significantly higher RMSE of 215.32. The RMSE indicates larger prediction errors, particularly during traffic surges or anomalies. Although the ANN achieves a strong R^2 values of 0.9892, explaining 98.92% of the variance in the data, its performance is less effective for daily traffic forecast.

The graph 8 visualizing the true versus predicted daily traffic flow values for ANN was created using 'Matplotlib' library, which allowed for the comparison of true versus predicted values reveals that the ANN's predictions tend to deviate more noticeably during periods of high traffic, failing to fully capture the sharp peaks seen in the true data. This suggests that the ANN model lacks the robustness and stability required to generalize effectively for daily traffic flow predictions at this level of aggregation.
For daily traffic flow predictions, CNN demonstrates superior generalization capabilities. With an R^2 value of 0.9925, CNN successfully captures 99.25% of the variance in the data. Its low RMSE of 14.3648 and MAE of 4.7157 indicate that the model consistently delivers predictions with minimal deviations from actual values. The true vs. predicted graph for the CNN (refer fig 9) illustrates a much closer alignment, particularly during periods of high traffic. This performance underscores the CNN's ability to leverage its architecture to uncover patterns and relationships in the data over time, making it an invaluable tool for accurate daily traffic predictions.
From a model perspective, ANN model is limited in its ability to capture all the temporal and spatial details required to accurately predict daily traffic flow, as it relies on dense layers, which are not as effective at recognizing complex temporal patterns. In contrast, the true vs. predicted graph for CNN shows a closer alignment, particularly during high-traffic periods, indicating the CNN's ability to handle sudden spikes and fluctuations.
Some external factors such as weather conditions, road accidents, construction works, or events, were not considered in this research, but which can affect traffic flow. Also, the daily data aggregation may have missed out some valuable information, reducing prediction accuracy. Changes in traffic patterns over time, such as urban development or new policies, might also affect the model performance.

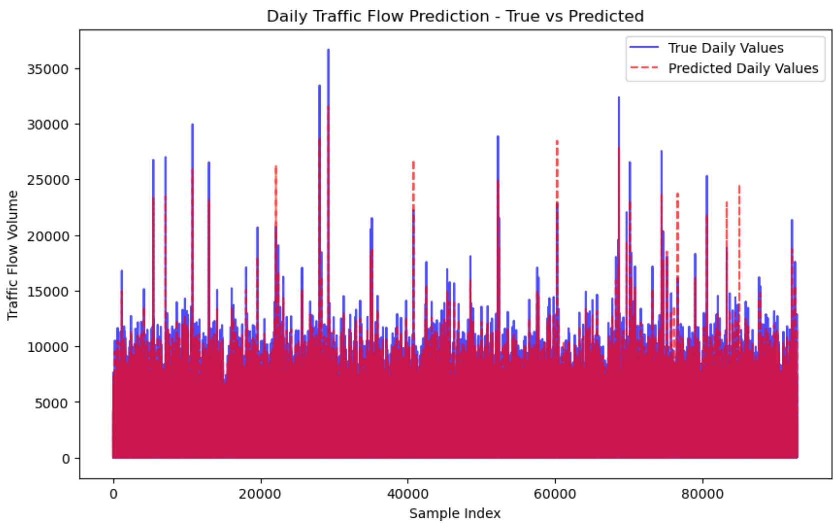

Figure 8: True vs. Predicted daily traffic flow values for ANN

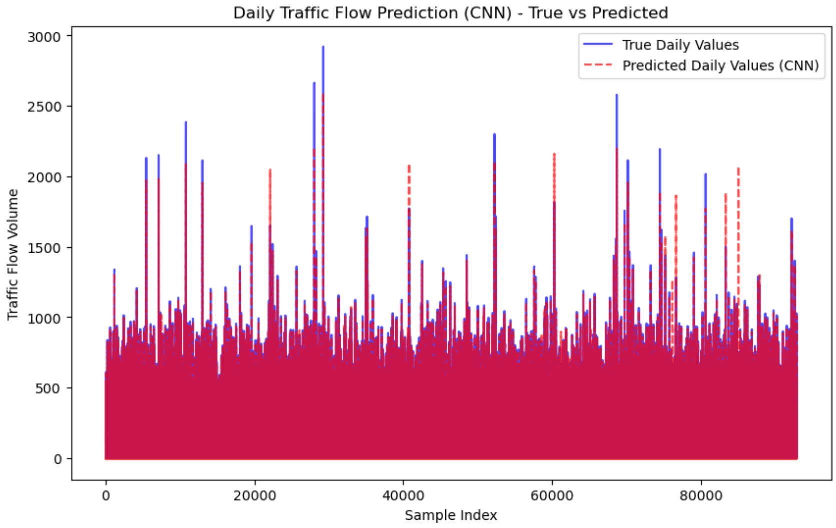

Figure 9: True vs. Predicted daily traffic flow values for CNN

6.3 Experiment 3: Hourly Prediction

The findings reveal that for hourly traffic flow predictions, the ANN model outperforms the CNN model. With low error metrics (MAE: 3.013, MSE: 74.1687, RMSE:8.6121) and a higher R^2 value 9932, the ANN demonstrates its effectiveness in capturing complex relationships within the data. This suggests that the ANN's architecture is particularly well-suited for structured, tabular datasets, where feature interactions can be effectively modeled without the need for advanced pattern extraction.
In contrast, while the CNN also delivered strong performance with an R^2 value of 0.991, its slightly higher metrics (MAE: 4.3418, MSE: 108.3294, RMSE: 10.4081) indicate that the added complexity of convolutional layers offers limited benefits for this type of predictions. These findings suggest that simpler models like ANNs can provide more accurate and efficient hourly forecasts, especially when the data does not necessitate complex feature extraction techniques.

6.4 Experiment 4: Prediction in 'IRE'

In region 'IRE', the CNN model demonstrated outstanding performance for monthly predictions, accounting for over 99% of the variance in traffic flow data with R^2 value of 0.9939. Its ability to generalize effectively and capture long-term trends is reflected in the metrics, MAE of 4.5309 and RMSE of 9. 0402.The ANN model also performed well for monthly predictions in this region, achieving an R^2 value of 0.9582, slightly lower than the CNN. However, the ANN exhibited larger discrepancies between predicted and actual values, with an MAE of 9.3309 and an RMSE of 23.758. These results suggest that although the ANN is effective at capturing general trends, it is not good as the CNN at addressing the temporal intricacies present in monthly traffic flow data.
For daily predictions in region 'IRE', both the CNN and ANN models predicted traffic flow with R^2 values of 0.9269 and 0.9288, respectively. The other evaluation metrics such as MAE and RMSE revealed considerable discrepancies between predicted and actual values, indicating challenges in accurately modeling aggregated daily trends. These results suggest that the models may struggle to generalize effectively to daily data, due to

complexity of traffic flow patterns.
The CNN and ANN models both demonstrated strong performance for hourly predictions in 'IRE', with R^2 values of 0.9935 and 0.9882, respectively, showing their ability to explain the majority of the variance in the data. CNN achieved slightly better accuracy, with lower error metrics (MAE of 2.00 and RMSE of 9.50) compared to ANN's MAE of 4.869 and RMSE of 12.775. These results indicate that both models are highly effective for short-term traffic flow predictions in this region, with the CNN providing a slight edge in precision.
The graphs below show the CNN model's performance in predicting monthly (fig 10) and hourly (fig 11) traffic flow prediction values.

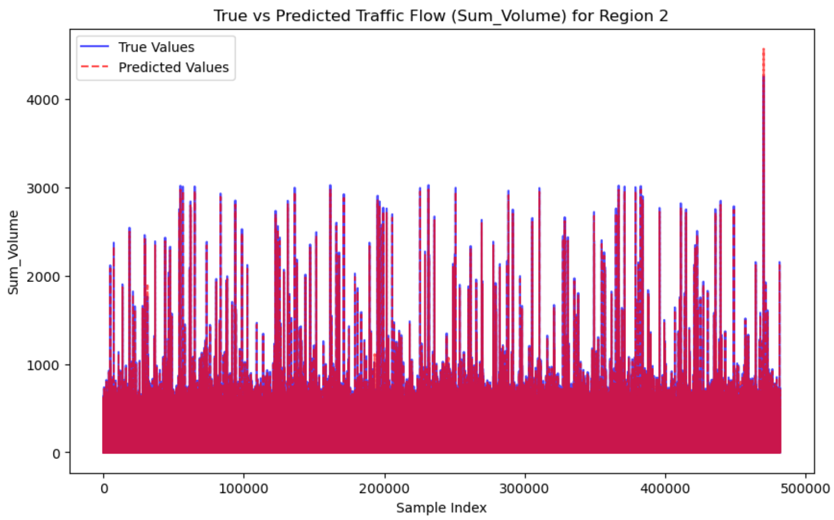

Figure 10: True Vs. Predicted traffic flow values- CNN Monthly

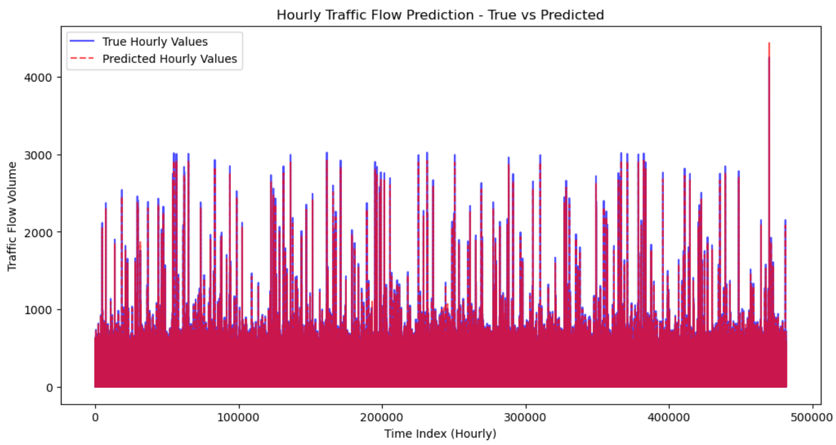

Figure 11: True Vs. Predicted traffic flow values- CNN Hourly

The close alignment between the true and predicted values in both monthly and hourly predictions demonstrate the CNN model's ability to effectively capture overall trends and fine- grained temporal patterns. However, occasional mismatches, especially during sharp or sudden peaks, show that the model has some difficulty handling abrupt changes in traffic flow.

6.5 Discussion

Both the ANN and CNN demonstrate notable strengths in predicting traffic flow across different temporal scales. The figure 12 is a table that presents the R^2, MAE, MSE,and RMSE values for all models.The consistently high R^2 values highlight their effectiveness in explaining a significant portion of the variance in traffic data, affirming the use of deep learning approaches in this field. The CNN excelled in daily and monthly forecasts with R^2 values of 0.9925 and 0.9951, effectively capturing intricate temporal patterns, while the ANN performed exceptionally well for hourly predictions with an R^2 value of 0.9932, leveraging its suitability for structured, tabular data that does not require complex temporal modeling.In the region 'IRE', while both the models excelled in hourly traffic predictions and showed moderate success in daily forecasts, the CNN consistently outperformed the ANN, especially in capturing long-term trends for monthly predictions.

Intervals	Model	R2	RMSE	MAE	MSE
Monthly	ANN	0.9918	9.4700	2.8964	89.6810
	CNN	0.9951	7.3657	2.0538	54.2533
Daily	ANN	0.9892	215.32	94.511	46364.09
	CNN	0.9925	14.3648	4.7157	206.3462
Hourly	ANN	0.9932	8.6121	3.0133	74.1687
	CNN	0.9901	10.4081	4.3418	108.4081

Figure 12: Comparisons of ANN and CNN Models for Monthly, Hourly and Daily implementation

The challenges faced by the models in daily traffic predictions in this study, indicated by higher RMSE and MAE values, are consistent with the observations made by **?**.They underscore the critical role of effective input data representation in capturing the spatiotemporal dependencies inherent in the traffic data, particularly for short-term traffic flow prediction. The paper highlighted that while deep learning models are highly effective at capturing complex patterns, daily forecasts can suffer due to insufficient representation of spatial and temporal relationships or the absence of standardized benchmark datasets. To overcome these limitations, incorporating temporal features such as weather, holidays and traffic incidents into the model design is essential. Furthermore, creating hybrid architectures that leverage CNN's strength in extracting temporal patterns and ANN's efficiency with structured data could improve performance across all prediction levels.

7 Conclusion and Future Work

High R^2 values across all intervals demonstrated strong predictive capabilities, confirming that both models effectively assessed the performance of ANN and CNN models across different intervals. The ANN outperformed in short-term (hourly) predictions due to its strength in processing structured tabular data, while the CNN excelled in capturing complex temporal patterns, particularly in daily and monthly forecasts. Regional analysis, such as in the 'IRE' region, further highlighted the model's effectiveness, achieving

high accuracy in hourly predictions but encountering challenges with aggregated daily patterns due to the complexity of traffic dynamics.
The findings of this study have important implications for urban traffic management. The ANN's strong hourly prediction performance highlights its potential for real-time monitoring and control. In contrast, the CNN's accuracy in daily and monthly predictions demonstrates its values for long-term strategic planning, such as optimizing infrastructure and guiding policy decisions that improve traffic management and urban planning.

Although the models delivered strong results, certain limitations are evident. Both models faced challenges with aggregated daily predictions, indicating the need for more robust architectures or enhanced datasets. Additionally, the absence of external factors such as weather, public events, and incidents limited the model's ability to capture more complex traffic dynamics effectively.

Future Work: To enhance performance across all prediction intervals, future research could focus on developing hybrid models that integrate CNN's ability to capture temporal patterns with ANN's efficiency in handling structured data. Incorporating additional contextual factors, such as weather, holidays, and traffic incidents, could improve accuracy for regional variability and aggregated trends. Improving model explainability through attention mechanisms would enhance transparency and build trust in predictions. Extending the models for scalability and real-time applications would support intelligent traffic management systems, with potential commercialization providing valuable insights for policymakers and urban planners. These advancements will further expand the role of deep learning in addressing the complexities or urban traffic forecasting.

References

Agafonov, A. (2020). Traffic flow prediction using graph convolution neural networks, pp. 91–95.

Çetiner, B. G., Sari, M. and Borat, O. (2010). A neural network based traffic-flow prediction model, *Mathematical and Computational Applications* **15**(2): 269–278.

Duan, Y., Lv, Y. and Wang, F.-Y. (2016). Performance evaluation of the deep learning approach for traffic flow prediction at different times, *2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 223–227.

Fouladgar, M., Parchami, M., Elmasri, R. and Ghaderi, A. (2017). Scalable deep traffic flow neural networks for urban traffic congestion prediction, *2017 international joint conference on neural networks (IJCNN)*, pp. 2251–2258.

Kareem, S., Hamad, Z. J. and Askar, S. (2021). An evaluation of cnn and ann in prediction weather forecasting: A review, *Sustainable Engineering and Innovation* **3**(2): 148–159.

Kranti Kumar, M. P. and Katiyar, V. K. (2015). Short term traffic flow prediction in heterogeneous condition using artificial neural network, *Transport* **30**(4): 397–405.

Liu, Y., Zheng, H., Feng, X. and Chen, Z. (2017). Short-term traffic flow prediction with conv-lstm, *2017 9th international conference on wireless communications and signal processing (WCSP)*, pp. 1–6.

McCann, B. (2014). A review of scats operation and deployment in dublin.

Medina-Salgado, B., Sánchez-DelaCruz, E., Pozos-Parra, P. and Sierra, J. E. (2022). Urban traffic flow prediction techniques: A review, *Sustainable Computing: Informatics and Systems* **35**: 100739.

Panda, R. and Nguyen, X. V. (2016). *Large scale real-time traffic flow prediction using SCATS volume data*, PhD thesis, MS thesis, Dept. Comput. Inf. Syst., Univ. Melbourne.

Polson, N. G. and Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction, *Transportation Research Part C: Emerging Technologies* **79**: 1–17.

Qi, X., Mei, G., Tu, J., Xi, N. and Piccialli, F. (2022). A deep learning approach for long-term traffic flow prediction with multifactor fusion using spatiotemporal graph convolutional network, *IEEE Transactions on Intelligent Transportation Systems* **24**(8): 8687–8700.

Shao, H. and Soong, B.-H. (2016). Traffic flow prediction with long short-term memory networks (lstms), pp. 2986–2989.

Ta, X., Liu, Z., Hu, X., Yu, L., Sun, L. and Du, B. (2022). Adaptive spatio-temporal graph neural network for traffic forecasting, *Knowledge-based systems* **242**: 108199.

Wang, Y., Jing, C., Xu, S. and Guo, T. (2022). Attention based spatiotemporal graph attention networks for traffic flow forecasting, *Information Sciences* **607**: 869–883.

Xu, D., Dai, H., Wang, Y., Peng, P., Xuan, Q. and Guo, H. (2019). Road traffic state prediction based on a graph embedding recurrent neural network under the scats, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29**(10).