Real-time Traffic Flow Prediction using Temporal - Graph Convolution Network and external attributes

Hoang-Anh T. Nguyen*,†, Ha-Dung Nguyen*,†, Trong-Hop Do*,†

* Faculty of Information Science and Engineering, University of Information Technology, Ho Chi Minh City, Vietnam.
† Vietnam National University, Ho Chi Minh City, Vietnam.

***Abstract*—In intelligent transportation systems, predicting traffic flow is always a pressing concern, especially in light of the current big data boom. An intelligent traffic system must be able to anticipate large volumes of data in real time. The research on traffic predictions mentioned above have changed throughout the years using a variety of methods. Traffic flow at a fixed location is not only influenced by its historical data, but also by external ones. Nevertheless, relatively few studies use a wide range of variables, such as the weather and news, in the job of prediction. We thus employ a space-time graph convolutional network (T-GCN) coupled with external elements on the presumption that their inclusion can enhance the spatial accuracy of traffic prediction and increase interpretability. When compared to conventional traffic prediction algorithms, tests on actual data sets demonstrate the efficiency of external information considerations on traffic forecasting jobs. MSE and MAE scores received on the predictions are 0.006 and 0.0398, respectively. Our team experimented with the model based on big data to anticipate real-time traffic and achieve successful outcomes.**

***Index Terms*—traffic flow prediction, deep learning, big data, real-time, Graph Convolution Network, Temporal - Graph Convolution Network, external factors**

I. INTRODUCTION

In recent years, a progressive rise has been seen in transportation needs along with the steady rise in the development of Intelligent Traffic Systems (ITSs). More and more emphasis was being paid to traffic forecasting, especially the need for real-time prediction. It is an essential part of a sophisticated traffic management system and has a big impact on how well traffic is managed, planned, and controlled. Traditional traffic forecasting involves looking at factors like flow, speed, and density of urban road traffic as well as patterns of travel and predicting future traffic trends [1]. The process of forecasting traffic is difficult in part because traffic volume, speed, and pattern depend on a variety of complicated dynamic and static elements, which are sometimes referred to as spatio-temporal correlations and external events [2].

Additionally, a variety of outside variables, like the weather, the existence of transportation hubs, emergency situations, and holidays, may have an impact on traffic information. These external factors have direct or indirect connections to traffic data that might affect the city's traffic situation. However, important elements like the state of the weather and traffic incidents are frequently disregarded in prior studies as a multisource input [3]. Accurate traffic forecasting is challenging due to the unpredictability and multidimensionality these elements introduce into traffic situations. These issues with traffic forecasting have been the subject of several prior studies for decades. Statistical approaches have been widely used, but these methods do not effectively capture the complicated non-linear spatial-temporal relationship since they are still reliant on standard time series models or machine learning models [4]. After the advent of deep learning, several researchers have used recurrent neural networks, such as the long short-term memory (LSTM) network [5], the gated recurrent unit (GRU) network [6] or convolutional neural network (CNN) [7] to model the non-linear spatial dependency [8], but they are limited as they only use Euclidean data. And recently, researchers tend to graph theoretic approaches to model the traffic data collected from road sensors. Graph convolution neural networks (GCN) are frequently used in traffic assessments because traffic road networks generate non-Euclidean data. Nevertheless, these techniques only take into account the topological relationships between the roads while creating the graphs, excluding any other outside variables.

Feeling the need for an efficient traffic prediction system, We wish to investigate a Temporal Graph Convolutional Model (T-GCN) described in [9] that is applied to the problem of real-time traffic forecasting utilizing the urban road network in conjunction with external factors.

Organization of this survey: The remainder of this essay is structured as follows. Section 2 makes reference to relevant works. Then, Section 3 defines the definition

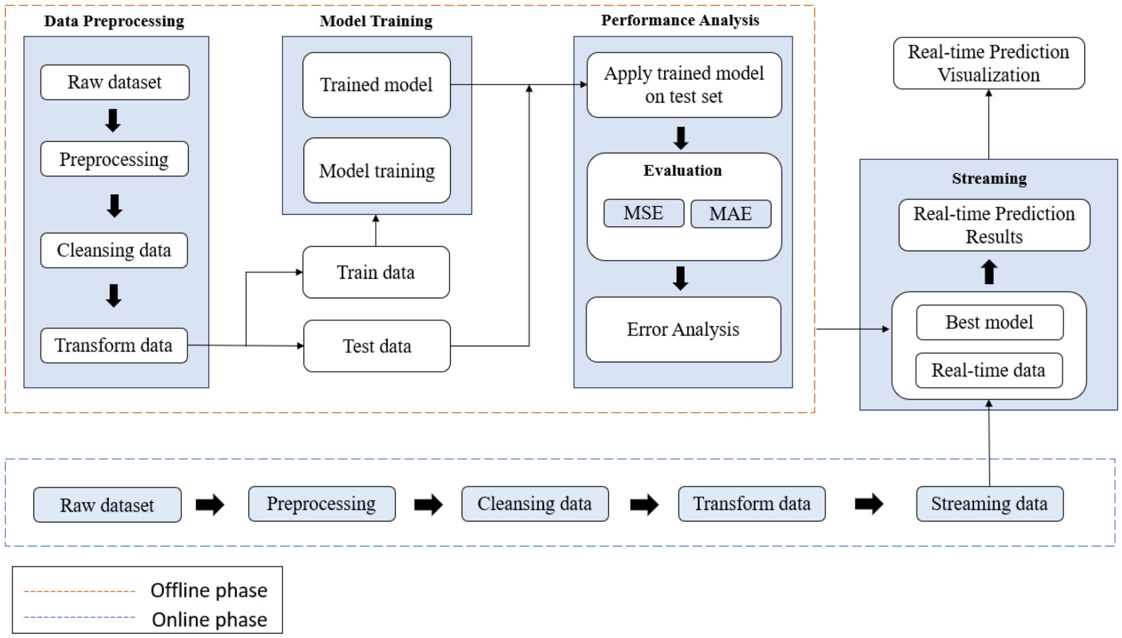


Fig. 1: Experimental procedure

and provides the details of our method. Subsequently, we present a few comparisons and evaluate the performance of the related methods in Section 4. The following section gives an overview of the real-time system. Finally, we conclude this article and discuss some future directions in Section 6.

II. Related Works

A. Traffic flow prediction

The approaches for predicting traffic have gone through many stages of development. Over the past fifty years, three main techniques to traffic modeling have been developed: deep learning, machine learning, and conventional statistical techniques.

In the early stage, statistical methods include historical averaging (HA) [10], vector autoregression (VAR) [11], and autoregressive integrated moving average (ARIMA) models and its several variations [12]. The strong nonlinearity and dynamism in traffic data, however, violate the linear and stationarity assumptions of these approaches, leading to subpar performance in actual use [13]. Later, machine learning-based traffic forecasting algorithms started to appear. The Bayesian inference approach [14] and support vector regression (SVR) [15] investigations came after the k-nearest neighbor algorithm (KNN) [16] was initially used to forecast traffic flow. High-dimensional data may be handled using machine learning techniques, which can also capture the complex nonlinearity that exists in the data. However, the manual feature selection, limited architecture, and segmented learning in these models are viewed as inadequate in big data settings [17].

The deep neural network models have drawn attention recently due to the rapid growth of deep learning and their ability to effectively capture the dynamic properties of traffic data. Models can be categorized into two groups depending on whether or not spatial dependency is taken into account. Some techniques solely take into account time dependency such as feedforward neural networks (FFNs) [18], deep belief networks (DBNs) [19], recurrent neural network (RNN) [20] and its variants long short-term memory (LSTM) and gated recurrent units (GRUs) [21]. Convolutional neural network (CNN)-based models have been used to simulate spatial dependency in order to capture it [22].

Although CNNs may partially adapt to the intricate topological structure of a road network, it was first created for spatial organization in Euclidean space. Traffic data are time-series data that are dispersed throughout a non-Euclidean road network. CNNs are unable to completely capture the spatial correlations of the traffic data as a result. In an effort to solve this problem, some recent works study the use of GCN (Graph Convolutional Network) for spatial modeling in road networks. Using GCN to learn the topological structure of the road network and GRU to learn the dynamic changes in traffic flow, T-GCN was proposed by Zhao et al. [9].

B. External factors

On the other hand, in addition to historical data and spatial linkages, the challenge of predicting traffic is also influenced by a number of outside variables, including weather, holidays, and traffic-related events. The majority of past studies only used one type of criterion to anticipate traffic flow. Recent studies have begun to suggest that additional external factors, such as the use of holidays and weekends [23], weather, and the distribution of nearby POIs [24], have an effect on traffic. In order to boost predicting effectiveness, a recent Oxford study added multidimensional meteorological data to a forecasting algorithm.

Another area of study investigated the connections between public opinion, indicated by tweet check-in kinds or numbers and content using geotagged social media as traveler evaluations. Ni et al. (2016) used Twitter user numbers and tweet counts related to the event to predict the number of people at a subway station during the event [25]. Lin et al. (2015) used information gleaned from tweets about adverse weather conditions to predict traffic speed [26]. However, the majority of prior strategies for creating social media news material are frequently focused just on the quantity of tweets with geotags, and seldom ever on the information included in the tweets themselves.

Based on this background, in this paper, we employ a graph convolutional neural network that can extract intricate spatial and temporal properties from traffic data provided by Ling et al. [9] and incorporate the use of external factors, such as weather conditions and traffic social media news, to predict actual traffic. Then, attribute data and traffic characteristics are combined to improve the model's perception of outside data and boost the precision of traffic forecasts.

III. Methodology

A. Problem Definition

The purpose of traffic forecasting is to make predictions about future traffic conditions based on past conditions and extra data. To anticipate future traffic, this paper's traffic forecasting work primarily relies on the volume of traffic over the prior period as well as external elements, including the weather and traffic news.

Definition 1: Road network G. The connection among road sections is represented in a road network as an unweighted graph $G = (V, E)$, where $V = (V_1, V_2, \ldots, V_n)$ represents the set of road sensors, n is the total number of road nodes, E is a collection of edges. The relationship between the roads is represented by the adjacency matrix A, which has the dimensions $A \in R^{N \times N}$. The common adjacency matrix used in earlier investigations only has entries 0 and 1. If there is no connection between the routes, the element is 0, and if there is, it is 1. The group combines the adjacency matrix with the distance matrix, which measures the separation between the locations, to boost the connection between the routes.

Definition 2: Traffic flow matrix X. Every node on the urban road network, illustrated by matrix X, is thought to have traffic flow as an intrinsic characteristic. We express the amount of traffic on the road network at time t as a vector with N-dimensions, where N is the total number of roads $X_t = [X_{1,t}, X_{2,t}, \ldots, X_{N,t}]$.

Definition 3: The external variables that influence traffic conditions are considered in this study as auxiliary characteristics of the route segments on the metropolitan road network. These elements might make up an attribute matrix $K = \{K_1, K_2, \ldots, K_m\}$ where m is the number of variable types. The attribute matrix K_j at time t for each attribute $j \in m$ is shown as a vector with n dimensions, where n is the total number of variables for this attribute $K_{j,t} = [K_{j_1,t}, K_{j_2,t}, \ldots, K_{j_n,t}]$.

In conclusion, the traffic prediction issue may be viewed as learning the function f based on the fundamental topology G, feature matrix X, and attribute matrix K of the road network in order to get traffic information in the future period T.

$$[x_{t+1}, x_{t+2}, ..., x_{t+T}] = f(G, X \mid K) \tag{1}$$

B. Data Preprocessing

General idea of data preprocessing steps are illustrated in Figure 2. The final dataset contains 4 types of data: traffic, distance matrix (weighted adjacency matrix), news and weather. Detailed preprocess of each data are demonstrated in the following subsections

1) Traffic: Traffic data are collected from the website Traffic Volume Viewer, Transport for NSW [1]. This website allows users to browse and search for available traffic count data in New South Wales (NSW). Data is available from 2006 up to the current year. Users can view and download traffic count data for the selected station for each hour of each day (for the date range where data is available).

First, our team selected all traffic sensors located in NSW. We also ensured that data from those sensors are available during the desired date range, which is a 3-year period from 07/07/2019 to 06/07/2022. We want a complete time series to feed into the model, so checking the missing values is an indispensable step. Unfortunately, most traffic time series data contain a large range of consecutive missing points, and the total number of missing days is up to more than 600, which can be seen in Figure 3(a). Stations which contain more than 200 dates of missing values were dropped from the dataset. The distribution of missing dates after dropping columns is shown in Figure 3.(b).

To build a graph of related traffic counters, every node of the graph should be close enough to generate meaningful connections. Therefore, the specific locations of each station were observed. There are several stations that are extremely far from the center of New South Wales. Traffic flow among those stations are considered not strongly affected by each other due to long distance

[1] Traffic Viewer website URL: NSW Traffic Viewer

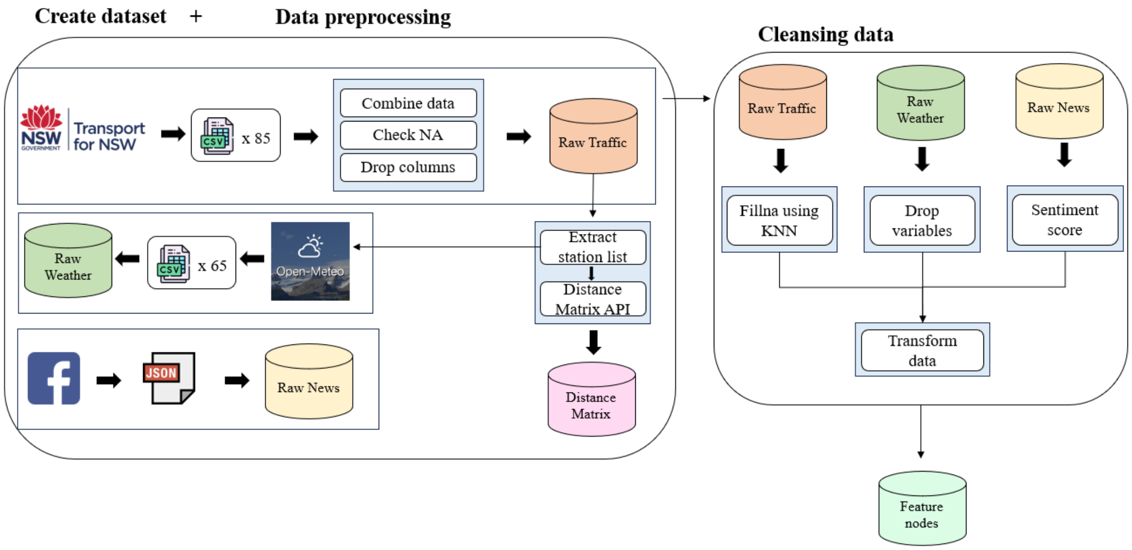
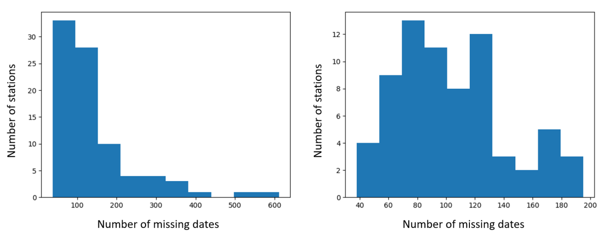


Fig. 2: Data Preprocessing Procedure



(a) Missing dates distribution before drop columns (b) Missing dates distribution after drop columns

Fig. 3: The distribution of missing dates before and after dropping columns

from the center, so they were omitted from the dataset. After dropping unhelpful columns, the shape of traffic is 26304 rows representing time axis and 65 columns, each one is a traffic flow series of each traffic station in 3 years.

The next step in traffic data preprocessing is the imputation of missing values. As the missing range is large and the nature of time series is complicated, simple imputation methods such as substitution of minimum, maximum, mean, median value, linear interpolation, etc,... are not applicable in this case. Python provides a package called 'imputena'[2], which allows both automated and customized treatment of missing values in datasets. The package can also recommend a treatment for a given dataset, inform about the treatments that are applicable to it, and automatically apply the best treatment. k-NN is recommended in this case, so it is applied on our dataset.

2) Distance Matrix: In order to utilize the spatial information of the traffic network, we construct a distance matrix, representing edge features of the road network graph. Travel distances between traffic stations in our network are collected using APIs provided by website Distancematrix.ai [3]. The weighted adjacency matrix is computed based on the distances between traffic counters and used threshold Gaussian kernel (Shuman et al., 2013) [27].

$$W_{ij} = \begin{cases} exp\left(-\frac{d(v_i,\ v_j)^2}{\sigma^2}\right), & if\ d(v_i,\ v_j) \leq \kappa \\ 0, & otherwise \end{cases} \tag{2}$$

W_{ij} denotes the edge weight between counter v_i and v_j, $d(v_i,\ v_j)$ represents the length of the shortest route between counter v_i and v_j, σ is the standard deviation of distances and κ is the threshold, which is assigned to 0.1 in our experiment.

3) News: All Facebook postings on the Live Traffic NSW [4] page were gathered to create our traffic news dataset. It was developed by a government agency in NSW and is a reliable news source. We conduct preprocessing using the gathered raw data in accordance with the below steps:

- Identify key characteristics for news data: We keep only the primary properties like post_id, post_text, post_time, and post_url since raw data includes many attributes but not all of them are pertinent to the issue.

[2]Python imputena package: https://github.com/macarro/imputena/tree/master

[3]Distance Matrix website: https://distancematrix.ai/

[4]Facebook page: Live Traffic NSW

- Filter news by chosen timeframe: News is filtered by the period from 7/7/2019 to 6/7/2022 during which traffic data was collected.
- Filter the news by relevant location: Each traffic sensor includes location data, including suburb, road name, intersection, etc. The news is then filtered based on all of this data.

Post_id	Post_text	Post_time	Post_url
3409032679168415	BANKSTOWN: Canterbury Road is closed in both directions between Stacey Street and Punchbowl Road due to a building fire. Avoid the area and use an alternative route.	03-02-2020 11:15:14	https://facebook.com/livetrafficnsw/posts/3409032679168415

Fig. 4: Examples of Traffic Posts from Facebook

Afterwards, the information from the postings is extracted from the preprocessed raw data as follows:

- Event time extraction: They have a significant influence on traffic since the information in the data is frequently accompanied with the beginning or finish of the event. We specify the event time as the post time if there is no time specified in the post. The timings will then be rounded to the nearest sixty minutes. For instance, if an accident happened at 8:38 AM, we may determine that it will have an impact on traffic at 9:00 AM.
- Location information extraction: Our team extracts location-related keywords from station data for each post using a regular expression and then create a list of sensors impacted by the news after that.

Finally, determining the polarity of each news item is a crucial step before we can utilize it for anticipating task. The team in this study applies a straightforward method based on sentiment analysis to determine the impact of news. Finding the positive or negative aspects of the news is the fundamental objective of sentiment analysis. We employ a collection of words with positive and negative sentiments, known as the Sentiment140 Lexicon [5], to gauge the positive and negative inclinations of morphemes. It comprises weighted phrases from tweets; a positive score denotes a positive relationship, while a negative score denotes a negative association.

4) Weather: Hourly historical weather data is collected using the Open-Meteo Historical Weather API [6]. This API enables users to download .csv files of data based on specific coordinates and time interval. 65 files of weather data are downloaded corresponding to each traffic counter. Each file contains 26304 rows and 26 columns of weather features such as temperature, precipitation, humidity, wind speed,.... All weather dataset contain no missing values, so there is no need for data imputation. Similar to traffic data, Min Max Scaler is also applied on weather features before feeding it into the model.

C. Temporal - Graph Convolution Network

This study focuses on the T-GCN stacked with a Data Fusion block, which includes both traffic and external data, including news and weather aspects, utilizing a deep hybrid neural network structure. This structure is mainly based on Zhao et al.'s (2019) [9]. T-GCN design, which includes employing a GCN to handle the graph structure of the road network and an RNN to analyze the temporal properties of the data. Stellargraph is a Python package that includes a built-in function based on Zhao's research, which uses a GCN paired with an LSTM to anticipate traffic. Figure 5 depicts the whole model architecture, including the data fusion layer and the entire T-GCN.

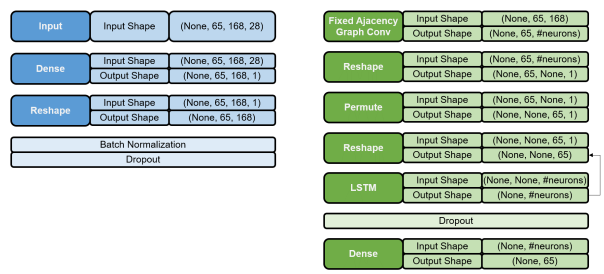


Fig. 5: Model Architecture: T-GCN stacked with a Data Fusion layer

1) Data Fusion: To include external data, a data fusion mechanism was devised. Following an examination of data fusion strategies, a data-in data-out (DAI DAO) data fusion technique was chosen (Castanedo, 2013) [27]. Essien et al. (2021) [28] used a traffic model that incorporated meteorological variables and traffic speeds in the first layer of a deep neural network. The second layer merged traffic data, news data, and meteorological factors using a multidimensional input. As a result, the mistakes caused at the prediction level are avoided, resulting in more dependable results.

The traffic data and external data were integrated into a four-dimensional input matrix: time, number of nodes, the input length and along with the extra dimension being the 28 features (one traffic feature, one news feature, and 26 weather features). This multidimensional array may be fed into a neural network using a five-dimensional input layer. The second layer lowers the 28 dimensions to one, thus "fusing" the data, as seen in Figure 6.

After passing through a dense layer of neurons, the data is dimensionally reduced, and batch normalization and a dropout layer are used to prepare it for feeding into the T-GCN.

2) T-GCN: For spatiotemporal modeling, the T-GCN simply combines the GCN and LSTM cells. To summarize, the T-GCN model is capable of dealing with

[5]Sentiment140 Lexicon dataset: Downloads

[6]Historical Weather API: Open-Meteo Historical Weather API

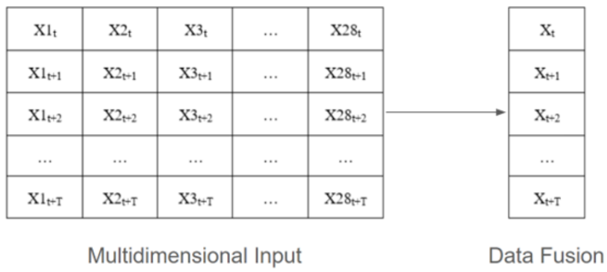


Fig. 6: Data Fusion layer

complicated spatial dependencies and temporal dynamics. On the one hand, the graph convolutional network is employed to acquire spatial dependency by capturing the topological structure of the urban road network. The gated recurrent unit, on the other hand, is used to record the dynamic fluctuation of traffic information on roadways in order to establish temporal dependency and, finally, to perform traffic prediction tasks.

Graph Convolutional Network - GCN: Obtaining complicated spatial dependence is a critical problem in traffic forecasting. Traditional convolutional neural networks (CNNs) may retrieve local spatial information, but only in Euclidean space, such as photographs, a regular grid, and so on. Because an urban road network is a graph rather than a two-dimensional grid, the CNN model cannot accurately describe spatial dependence due to the complicated topological structure of the urban road network. Recently, there has been a lot of interest in generalizing the CNN to the graph convolutional network (GCN), which can handle any graph-structured input. Many applications have employed the GCN model successfully, including document classification [29], unsupervised learning [30], and image classification [31]. The GCN model builds a filter in the Fourier domain, which works on the nodes of the graph and their first-order neighbors to capture spatial characteristics between the nodes, and then the GCN model is formed by stacking many convolutional layers. A visual representation of a GCN is illustrated in Figure 7.

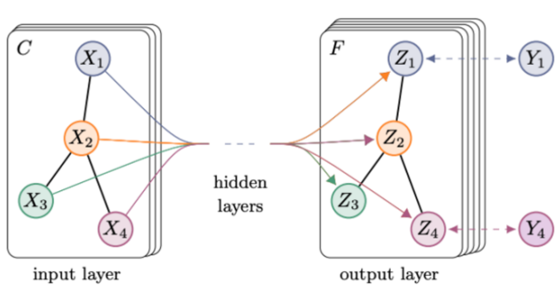


Fig. 7: A visual representation of a GCN (Kipf and Welling, 2017)

The structure of a 1-layer GCN can be represented as the following

$$f\left(X,\ A\right) = \sigma(\hat{A}\left(\hat{A}XW_0\right)W_1) \tag{3}$$

where, $\hat{A} = \ \tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}$, W_0 is the weight matrix $W_0 \in R^{P\times H}$, P is the feature length, H is the number of hidden units, W_1 is the weight matrix $W_1 \in R^{H\times T}$, T is the prediction length, $f\left(X,\ A\right)$ is the output with prediction length T, $ReLU$ is the activation function for the GCN layer, which can be customized.

The adjacency matrix must be fed to the GCN layer in addition to the feature matrix in order for the GCN to grasp the relationships in the temporal graph. The GCN layers' output is then sent to the LSTM.

Long-Short Time Memory - LSTM: Another major issue in traffic forecasting is acquiring temporal dependency. The recurrent neural network (RNN) is now the most extensively used neural network model for processing sequence data. However, the typical recurrent neural network has limits for long-term prediction due to flaws such as gradient disappearance and gradient explosion [32]. The LSTM model is a recurrent neural network variation that has been shown to handle the difficulties listed above. The essential ideas of LSTM are that it uses a gated mechanism to memorize as much long-term information as possible and that it is equally successful for varied jobs.

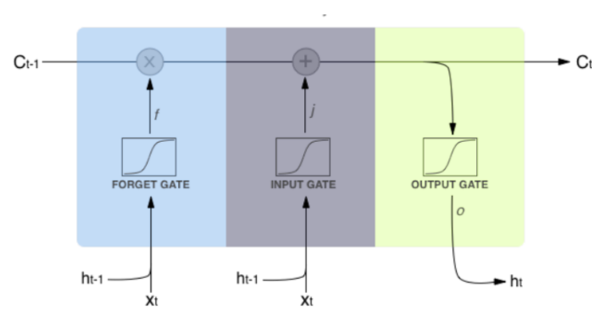


Fig. 8: A LSTM memory cell

To produce predictions, an LSTM employs three gates: an input gate, an output gate, and a forget gate, as shown in Figure 8. Each gate employs a sigmoid function to generate a value ranging from 0 to 1, with "0" shutting the gate and "1" opening the gate (Hochreiter and Schmidhuber, 1997) [33]. Each LSTM cell state is calculated by evaluating what information from the previous cell state must be discarded and what information from the present cell state must be considered. The LSTM network can recall essential long-term patterns while simultaneously evaluating short-term inputs by repeating this procedure over multiple time steps (Staudemeyer and Morris, 2019) [34]. The following are the three equations for each gate:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{4}$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (5)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (6)$$

where c_{t-1} is the cell state from previous LSTM cell $t-1$, h_{t-1} is the output from previous LSTM block at timestep $t-1$, x_t is the input at the current timestamp, c_t is the cell state from current timestamp, i_t is the input gate, f_t is the forget gate, o_t is the output gate, σ is the sigmoid function, w_x is the weight for the respective gate neurons, b_x is the bias for each gate.

Before the final forecast is formed, the data is sent via a dropout layer and one additional neuron layer after the LSTM layer. The outcome is a traffic forecast N steps in the future.

IV. Experiments and Results

A. Evaluation Metrics

We assess the performance of different models using two metrics below:

Mean Squared Error (MSE): MSE measures the average of the squares of the errors, which means the average squared difference between the estimated values and the actual value.

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}^i - y^i)^2 \quad (7)$$

Mean Absolute Error (MAE): MAE measures the average magnitude of the errors in a set of predictions, without considering their direction.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}^i - y^i| \quad (8)$$

B. Baselines

We conduct multiple experiments on different external datasets. The T-GCN model without using external also was considered for this study, mainly for comparison purposes. **Experiment 1**: The T-GCN does not include a data fusion layer, and simply includes the GCN and LSTM cells for spatiotemporal modeling. This is to determine the impact that external data has on the traffic prediction problem, a comparison against an identical model without external data would allow for critical analysis **Experiment 2**: The full architecture of data fusion, GCN and LSTM layers are applied on traffic data, distance matrix, news and weather features. **Experiment 3**: We drop weather features from the experiment 2 to examine the impact of news on traffic prediction. **Experiment 4**: We drop news feature from the experiment 2 to examine the impact of weather on traffic prediction.

For a better understanding of the efficiency of T-GCN architecture, its performance will be compared with that of other traditional time-series predictive models.

History Average model (HA) [35]: this model calculates the average traffic flow in the historical periods and uses them as predictions

Autoregressive Integrated Moving Average model (ARIMA) [36]: The ARIMA forecasting equation for a stationary time series is a linear equation in which the predictors consist of lags of the dependent variable and/or lags of the forecast errors

Support Vector Regression model (SVR) [37]: The SVR algorithm aims to find the hyperplane that passes through as many data points as possible within a certain distance, called the margin. This approach helps to reduce the prediction error and allows SVR to handle non-linear relationships between input variables and the target variable using a kernel function.

Long Short Time Memory model (LSTM) [33]: this is a recurrent neural network (RNN), aimed to deal with the vanishing gradient problem present in traditional RNNs. Its relative insensitivity to gap length is its advantage over other RNNs, hidden Markov models and other sequence learning methods.

Localized Spectral Graph Convolution - Long Short Time Memory (LSGC-LSTM): this model combines a localized spectral graph convolution layer [29] and an LSTM layer. The proposed spectral filters aggregate information not only from the direct neighbors of a node, but from its k-hop neighborhood. In our experiment, we set k = 16.

Graph Convolution Network model (GCN) [30]: this is an approach for semi-supervised learning on graph-structured data. It is based on an efficient variant of convolutional neural networks which operate directly on graphs

C. Experimental Settings

There were hyperparameters for the T-GCN that needed to be optimized. The learning rate, activation function, number of layers, number of neurons in each layer, batch size and more parameters were all included, as summarized in Table I . This was a time-consuming operation, and for the sake of this study, hyperparameter adjusting was limited to experiment 2, optimizing the activation function and number of neurons in each layer. Both the GCN and LSTM layers have a maximum layer count of 1.

The number of epochs specified for hyperparameter tuning was 10 to identify which technique had the highest chance of performing well. Normally, a considerably greater number for epochs is utilized, however owing to computer restrictions, the number has to be set relatively low in order to provide timely results. As is common practice with the optimizer "adam," the learning rate was likewise kept at 0.001 and the batch size is set to 64. Because K-fold cross validation is difficult to apply to time series prediction, it was not addressed in the hyperparameter tuning procedure.

The purpose of our study is to predict one day forward traffic flow, so the pre_len is always set to 1 (24 timesteps). It is important to choose which is the best

Hyperparameter	Value
Learning Rate	0.001
Epoch	10
Batch size	64
seq_len	7
pre_len	1
Data Fusion Layer Activation	linear
GCN Activation	linear
Number of GCN layer	1
GCN Layer Neurons	10
LSTM Activation	linear
Number of LSTM layer	1
LSTM Layer Neurons	linear

TABLE I: Summarization of Hyper-parameter tuning

sequence length that the model will look back to predict future values. We choose a number of different sequence lengths (1, 2, 7, 14, 30) and analyze the change of error score. The results are illustrated in Figure 9. It can be seen that a 7-day sequence length provides the lowest training errors, so seq_len is set to 7 in our experiments.

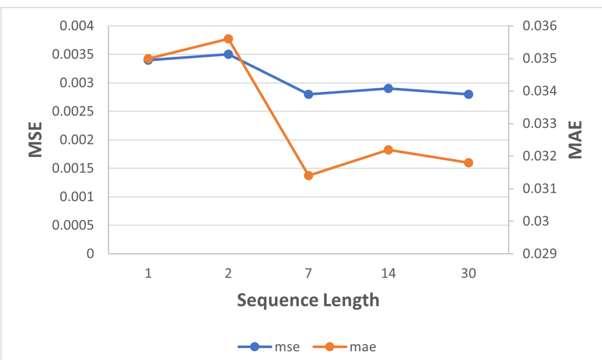


Fig. 9: Error score of T-GCN training on different sequence lengths

The training period of data was from July 07 at 00:00 am - November 19, 2021 at 23:00 pm, and the testing period was from November 20, 2021 at 00:00 am - June 27, 2022 at 23:00 pm. Altogether, 867 days of training data and 220 days of testing data. All model training and hyperparameter tuning was carried out in Google Colaboratory notebooks.

D. Experimental Results

Table II summarizes the training and testing errors using model T-GCN on different external datasets. Experiment 2 which utilizes both news and weather features have the best performance in general. Although learning

Experiment	External Data	Train		Test	
		MSE	MAE	MSE	MAE
1	No	0.0032	0.0368	0.0073	0.0585
2	News, Weather	0.0028	0.0314	0.0060	0.0398
3	News	0.0037	0.0368	0.0068	0.0505
4	Weather	0.0033	0.0346	0.0066	0.0346

TABLE II: Error scores of T-GCN training and testing on different external datasets

Model	Dataset	MSE	MAE
HA	Traffic	0.0899	0.2586
ARIMA	Traffic	0.0823	0.2392
SVR	Traffic	0.0602	0.1214
LSTM	Traffic	0.0202	0.1018
LSGC-LSTM	Traffic, adjacent matrix	0.0551	0.1999
GCN	Traffic, adjacent matrix	0.0194	0.0993
T-GCN	Traffic, adjacent matrix	0.0073	0.0585
T-GCN	Traffic, adjacent matrix, external data	0.0062	0.0398

TABLE III: Comparison of T-GCN stacked with data fusion with other predictive methods

with no external data has a good training result just after experiment 2, the error score of it is the highest among 4 experiments. This indicates that the model trained in experiment 1 is the most overfitting one. Between two types of auxiliary attributes, weather seemed to be more helpful in traffic prediction task as both MSE and MAE on train and test phase are lower than those of experiment 3. However, another explanation for better influence of weather attributes could be its large number of features (26) compared to news feature (1).

Table III shows the performance of several traditional techniques and component deep learning models on predicting traffic flow. All statistics imply that T-GCN is the best method to extract both spatial and temporal information from the data compared to its nuclear models, including LSTM, LSGC-LSTM and GCN. Traditional models such as HA, ARIMA, SVR, which utilize only traffic flow time series, obtain the least accurate predictions. All empirical experiments in this study prove that using spatial-temporal network models to learn the complexity of the dynamic traffic flow is an effective approach for traffic prediction. At the same time, add auxiliary attributes such as news and weather do help enhance the model performance.

Figure 10 illustrates some prediction results at several stations in 1 day. The predicted lines look impressive as they fit the ground truths at an acceptable degree. The proposed model has the ability to catch the daily pattern of traffic although the predictions tend to be less accurate at peak time.

V. Real-Time Traffic Prediction

A. Real-time system architecture

Fig. 11 shows the architecture of the proposed external data-based traffic flow prediction system. To build the data transmission source, the raw data is gathered.

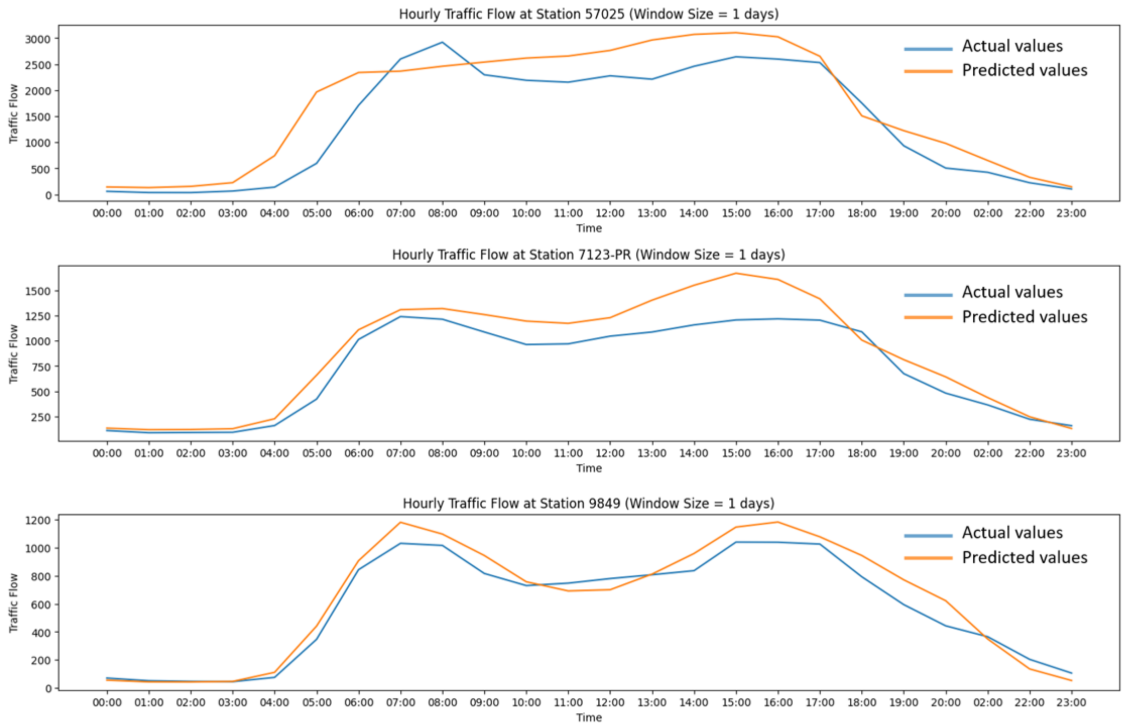


Fig. 10: Visualization of traffic prediction at several station in 1 day

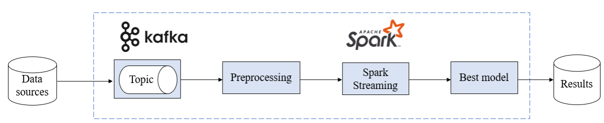


Fig. 11: System architecture

These data are streamed via Kafka and delivered into Spark streaming-integrated components. Preprocessing and data cleaning procedures resemble those of offline processing. After being put into the trained model, the clean data produces the best predictions for the data coming from Kafka.

1) Apache Kafka: A distributed messaging system called Apache Kafka was created and is maintained by Apache; as a result, the message broker is also known as Apache Kafka. It is created using the public/subscribe approach, just as other message broker programs or applications. The party that publishes the data is referred to as the producer, and the party that gets the data in accordance with the themes is referred to as the consumer.
2) Spark Streaming: Apache Spark Streaming is a scalable, fault-tolerant streaming processing solution that supports batch and streaming workloads out of the box. Data engineers and data scientists can analyze real-time data from a variety of sources, including (but not limited to) Flume, Amazon Kinesis, and Kafka. Combining Kafka and Spark Streaming is done so that we may read data from a single topic or from a number of topics fast while synchronously preserving all of the data you get from Kafka for simple recovery

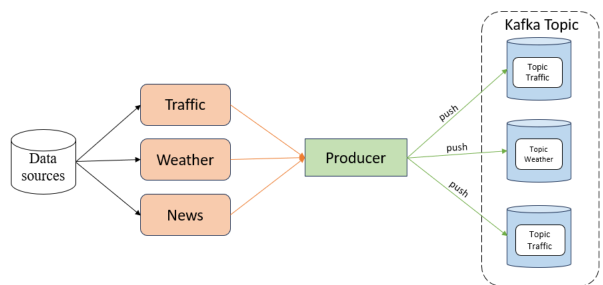


Fig. 12: Kafka's producer system in our experiment.

B. Real-Time Prediction

The minimum sequence length of historical data fed into T-GCN is seq_len + pre_len (8 days). To predict 1-day length hourly traffic data, we add 1 day more to the minimum time (9 days), so the model vsn slide through 24 data points and give 24-point predictions.

As our team cannot collect real-time data from NSW Transport Viewer, we do the data streaming with the assumption that real-time data is the pre-downloaded data located on Google Drive platform. Therefore, 9-day data from June 28, 2022 at 00:00 am to July 06, 2022 at 23:00 pm is collected as historical data for real-time prediction.

We choose the best model from our experiment, T-GCN trained on news and weather data, to perform real-time predictions. The model will forecast the real-time hourly traffic flow at all traffic stations in NSW in 07/07/2022.

VI. Conclusion and Future Work

In short, our project has successfully handled three main approaches. We are able to extract useful information from external data, namely news and weather, for traffic prediction task. Our team also succeeded in implementing a hybrid deep neural network to extract both spatial and temporal features of the data. Model T-GCN combined with a layer of data fusion performs well in our datasets and provides promising results. In particular, MSE scores are 0.0028 and 0.006, MAE scores are 0.0314 and 0.0398 in the training and testing phase, respectively. Finally, we build a real-time traffic prediction framework using Spark streaming.

During the research process, there appear several challenges. Raw traffic flow time series obtained from NSW Traffic Volume Viewer contain huge and consecutive missing range, which strongly affects the performance of predictive models and requires advanced imputation techniques. The amount of news related to each traffic station is limited. Our team members' also face several difficulties when implementing and modifying the original T-GCN Python code to fit our high-dimensional dataset.

In future works, our team will improve data preprocessing steps to drop noises and extract more useful information from the datasets. We also continue to discover more advanced and latest spatial-temporal models to solve the traffic prediction problem more effectively. We also aimed to build an interactive tool for real-time traffic prediction and visualization.

References

[1] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," 2018.

[2] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, and A. Khokhar, "Reinforced spatiotemporal attentive graph neural networks for traffic forecasting," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6414–6428, 2020.

[3] D. Zhang and M. Kabuka, "Combining weather condition data to predict traffic flow: A gru based deep learning approach," *IET Intelligent Transport Systems*, vol. 12, 03 2018.

[4] S. Sun, H. Wu, and L. Xiang, "City-wide traffic flow forecasting using a deep convolutional neural network," *Sensors*, vol. 20, no. 2, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/2/421

[5] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 2015, pp. 153–158.

[6] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328.

[7] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with conv-lstm," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2017, pp. 1–6.

[8] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," 2018.

[9] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[10] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: Comparison of modeling approaches," *Journal of Transportation Engineering*, vol. 123, no. 4, 8 1997. [Online]. Available: https://www.osti.gov/biblio/522384

[11] S. Chandra and H. Al-Deek, "Predictions of freeway traffic speeds and volumes using vector autoregressive models," *Journal of Intelligent Transportation Systems*, vol. 13, pp. 53–72, 04 2009.

[12] H. Mei, A. Ma, S. Poslad, and T. Oshin, "Short-term traffic volume prediction for sustainable transportation in an urban area," *Journal of Computing in Civil Engineering*, vol. 29, 04 2013.

[13] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *Trans. Intell. Transport. Sys.*, vol. 23, no. 5, p. 3904–3924, may 2022. [Online]. Available: https://doi.org/10.1109/TITS.2020.3043250

[14] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[15] C.-H. Wu, J.-M. Ho, and D. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[16] L. Hua-pu, "Short-term traffic flow forecasting based on k-nearest neighbors non-parametric regression," *Journal of systems engineering*, 2009.

[17] J. Liu, T. Li, P. Xie, S. Du, F. Teng, and X. Yang, "Urban big data fusion based on deep learning: An overview," *Information Fusion*, vol. 53, 06 2019.

[18] D. Park and L. R. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 357–367, 1999. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/0885-9507.00154

[19] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[20] Z. Zhene, H. Peng, L. Lin, X. Guixi, B. Du, M. Bhuiyan, Y. Long, and D. Li, "Deep convolutional mesh rnn for urban traffic passenger flows prediction," 10 2018, pp. 1305–1310.

[21] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328.

[22] S. He, Q. Luo, R. Du, L. Zhao, G. He, H. Fu, and H. Li, "Stgc-gnns: A gnn-based traffic prediction framework with a spatial–temporal granger causality graph," *Physica A: Statistical Mechanics and its Applications*, vol. 623, p. 128913, 2023.

[23] B. Liao, J. Zhang, C. Wu, D. G. McIlwraith, T. Chen, S. Yang, Y. Guo, and F. Wu, "Deep sequence learning with auxiliary information for traffic prediction," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[24] J. Zhu, C. Tao, H. Deng, L. Zhao, P. Wang, T. Lin, and H. Li, "Ast-gcn: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," 2020.

[25] Z. Zhang, M. Ni, Q. He, J. Gao, J. Gou, and X. Li, "Exploratory study on correlation between twitter concentration and traffic

surges," *Transportation Research Record*, vol. 2553, no. 1, pp. 90–98, 2016.
[26] L. Lin, M. Ni, Q. He, J. Gao, and A. W. Sadek, "Modeling the impacts of inclement weather on freeway traffic speed: Exploratory study with social media data," *Transportation Research Record*, vol. 2482, no. 1, pp. 82–89, 2015. [Online]. Available: https://doi.org/10.3141/2482-11
[27] F. Castanedo, "A review of data fusion techniques," *TheScientificWorldJournal*, vol. 2013, p. 704504, 01 2013.
[28] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "A deep-learning model for urban traffic flow prediction with traffic events mined from twitter," *World Wide Web*, vol. 24, 07 2021.
[29] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *CoRR*, vol. abs/1606.09375, 2016. [Online]. Available: http://arxiv.org/abs/1606.09375
[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907
[31] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2014.
[32] Y. Bengio, P. Y. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5 2, pp. 157–66, 1994.
[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.
[34] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM - a tutorial into long short-term memory recurrent neural networks," *CoRR*, vol. abs/1909.09586, 2019. [Online]. Available: http://arxiv.org/abs/1909.09586
[35] L. Jing and G. Wei, "A summary of traffic flow forecasting methods," *Journal of Highway and Transportation Research and Development*, vol. Vol. 21, pp. 82–85, 03 2004.
[36] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," 1979.
[37] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 08 2004.