Machine Learning Approaches for Traffic Flow Forecasting

By

Arsalan Ahmad Rahi

Submitted to the University of Hertfordshire in partial fulfilment of the requirement of the degree of Master of Philosophy (MPhil)

Submission Date 01/05/2019

Supervisor: DR. Sooda Ramalingam

School of Engineering and Technology
University of Hertfordshire

DECLARATION STATEMENT

I, Arsalan Ahmad Rahi, the author of this project, hereby declare that this research study titled "Machine Learning Approach for Traffic Flow Prediction" is my own genuine work from beginning to end. I certify that the work submitted with all the materials, sources and the form of published or unpublished work of other persons used in this thesis, were duly acknowledged by means of IEEE numeric referencing. (ref. UPRAS/C/6.1, Appendix I, Section 2 – Section on cheating and plagiarism)

Student Full Name: Arsalan Ahmad Rahi

Student Registration Number: 14111965

Signed: ……

Date: 01/05/2019

ABSTRACT

Intelligent Transport Systems (ITS) as a field has emerged quite rapidly in the recent years. A competitive solution coupled with big data gathered for ITS applications needs the latest AI to drive the ITS for the smart and effective public transport planning and management. Although there is a strong need for ITS applications like Advanced Route Planning (ARP) and Traffic Control Systems (TCS) to take the charge and require the minimum of possible human interventions. This thesis develops the models that can predict the traffic link flows on a junction level such as road traffic flows for a freeway or highway road for all traffic conditions.

The research first reviews the state-of-the-art time series data prediction techniques with a deep focus in the field of transport Engineering along with the existing statistical and machine leaning methods and their applications for the freeway traffic flow prediction. This review setup a firm work focussed on the view point to look for the superiority in term of prediction performance of individual statistical or machine learning models over another. A detailed theoretical attention has been given, to learn the structure and working of individual chosen prediction models, in relation to the traffic flow data.

In modelling the traffic flows from the real-world Highway England (HE) gathered dataset, a traffic flow objective function for highway road prediction models is proposed in a 3-stage framework including the topological breakdown of traffic network into virtual patches, further into nodes and to the basic links flow profiles behaviour estimations. The proposed objective function is tested with ten different prediction models including the statistical, shallow and deep learning constructed hybrid models for bi-directional links flow prediction methods. The effectiveness of the proposed objective function greatly enhances the accuracy of traffic flow prediction, regardless of the machine learning model used.

The proposed prediction objective function base framework gives a new approach to model the traffic network to better understand the unknown traffic flow waves and the resulting congestions caused on a junction level. In addition, the results of applied Machine Learning models indicate that RNN variant LSTMs based models in conjunction with neural networks and Deep CNNs, when applied through the proposed objective function, outperforms other chosen machine learning methods for link flow predictions. The experimentation based practical findings reveal that to arrive at an efficient, robust, offline and accurate prediction model apart from feeding the ML mode with the correct representation of the network data, attention should be paid to the deep learning model structure, data pre-processing (i.e. normalisation) and the error matrices used for data behavioural learning.

The proposed framework, in future can be utilised to address one of the main aims of the smart transport systems i.e. to reduce the error rates in network wide congestion predictions and the inflicted general traffic travel time delays in real-time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

This is an introductory chapter in which the initial undertaken topic of study with a little background is presented. The subject motivation and the research questions with the aim and the contributions are also discussed.

There has been a vast increase in the big data that is available through the advent of smart things, smart cities and smart transportation and internet of things. The next thing that arose in the organisations mind is to find ways on how to put this big data in use for meaningful use. This question motivated the author to further research using the big data gathered in the field of transportation in conjunction with AI techniques to build useful systems.

Intelligent Transport Systems (ITS) is all about providing the end users the innovative and advanced services to seamlessly use different modes of transportation and traffic management for timely effective planning and to empower users to make smarter choices in the latest multi modular transportation system. Having such a system can reasonably predict the transport changes and will be of great importance for the transportation authorities, government and the public institutions. Daily we use public and private transport on our everyday commute. We take it for granted that the buses arrive on the scheduled times. Due to the ever-increasing population there is a need for the public to have the better travelling experiences on commercial transports.

1.1 Problem Statement

The issues affecting a common native may include preparing for the right currency change beforehand in terms of cash to be paid for on board ticket purchase. The bus company eventually had to pay the cost for each extra time that the passenger takes while boarding the bus. Once boarded, the passenger constantly looks out for the desired stop out of the bus window. The bus takes a certain time to reach the destination that is affected by the variable congressional regions along the route. This cycle continues day in day out. There is nothing smart about this cycle of action; this is the normal life of the bus running operations.

Public transportation has thus not evolved much enough over the past years as we expect it to along with the growing technology, so that we can call it an efficient means of operation. Technology wise, fare prices may drop, flexible dynamic congestion-based routes may be provided and ultimately bus road driving behaviour and reliability issues can be solved. Having a system that can effectively predict the traffic behaviours this will not only reduce the costs but will also help in reducing carbon footprints as well. This thesis is written to address the improvements in some of the most distinctive fields of the transportation industry that eventually constitutes to the smart transportation industry.

The closest up-to-date system deployed in the England UK today is at this website[1] by Highway England. The system gives close to Realtime traffic information for most of the highways, motorways and major A roads in the England. The traffic information displayed includes the average traffic flows for each junction for clockwise and counter clockwise directions, on the corresponding motorways. The CCTV image is taken at a frequency of one minute. The data sources are the onsite deployed loop detectors and the microwave sensors at the specific locations on the road.

[1] https://m.highwaysengland.co.uk/#flow

There are some of the issues inferred from the current implemented system:

- The system relies on the data collection sensors deployed at certain major road locations, which makes it easy to ascertain the traffic average road traffic speeds but at present cannot make a meaningful elaborative system so to make the sense of the nearby links.
- At the current state the system cannot make any prediction from the current data and just displays the instantly averaged speed and generates the control signals specifying the delays for the e-signs on the roads. So, there is a need of latest AI based deep machine learning techniques employment to make the effective forecasting by analysing the behaviours of the closely related traffic flow links data.

1.2 Aims and Research Questions

Based on the problem statement, the research questions explored in this research are as follows:

RQ1:

What are the potential hindering challenges for the practical implementation of the road traffic parameter forecasting systems?

RQ2:

What conventional neural network-based techniques have to offer for the traffic parameter predictions?

RQ3:

What are the state-of-the-art traffic prediction machine learning architectures for traffic flow forecasting and what effects does the proposed methodology has on the chosen model performances?

RQ4:

What deep machine learning approaches have to offer when compared to conventional or shallow machine learning techniques considering the traffic flow data?

1.3 Research Method

The background study and state of the art literature review was performed to answer the first three research questions. Proposed methodology and the null hypothesis containing the flow prediction objective function was put to test in an iterative manner for different models. The experiments were then conducted on the state-of-the-art deep ML models comparing them quantitatively with conventional ML and statistical forecasting approaches. Result based hypothesis was then drawn. The hypothesis is then further tested with multi time dependent model predictions. Conclusions are then made in the end.

1.4 Contributions

This thesis contributes by gathering the most elaborative machine learning techniques from shallow to the state-of-the-art deep learning approaches to do the predictions for traffic flows while optimising for the basic junction level highway traffic flow proposed objective function. The bi-directional flow function of individual roads is reported considering the net inflows and outflows by a

topological breakdown of the highway network. The proposed approach is modular and can be adopted for network wide traffic flow behavioural learning. Further the technique can help in considering the bottlenecks for congestions analysis.

1.5 What is Machine Learning?

Artificial intelligence (AI) has become one of the hot buzzwords in the recent times. Artificial Intelligence is the broader term used to describe the control algorithms and systems that derive the machines to undertake their tasks that are considered smart. Machine learning (ML) is the application of AI on machines that is since the access of relevant data to the machines will make them learn for themselves. MLs can have different types that differ in various ways but one thing which is common is that they all operate on in data. So, the data relevancy and accessibility are the main keys in any ML performance. ML is also referred to as the subset of the AI. It's not wrong to think of ML as the current state of the art technology.

1.5.1 How Machine Learning Works

Machine learning is the set of techniques used to figure out and perform certain tasks from a given set of data. The data from a specific field that is available serves as the main driving force to the learning process of the classifier that later classifies and predicts in the future. The algorithm then devises its rules and functions either itself (unsupervised) or from users handpicked features (supervised). This phase of learning and training an algorithm is called the training phase. After this the learned algorithm instance is tested against the validation and test dataset in the training and testing phases respectively. The accuracy and performances are then categorised against standard benchmarking algorithms in a simulation environment. This approach is rather more feasible than the conventional programming approaches as humans don't need to put a lot of effort. As more and more data become available more generalisations could be extracted from the example data. However for machine learning application to be successful one has to have a good gripping of 'black art' that is very hard to find in textbooks [1].

1.5.2 Innovations in Machine Learning

Machine learning is a form of analytical solution that automates the process of data analysis. Machine learning consists of algorithms that iteratively look for patterns in data and learn some useful hidden insights that can ultimately make the computer aware without programming them explicitly. Today's machine learning has changed a lot from the machine learning of past. Now-a-days, Machine learning algorithms are being devised faster and faster due to the possibility of applying complex mathematical operations to big data that have become a reality.

1.5.3 Self-Driving Cars

Some of the widely implemented applications gaining much popularity in talks now-a-days are: The Google and tesla's self-driving cars. They are all practically possible because of capability of the machine learning algorithms.

1.5.4 Recommendation Systems

Line of interest product recommendation systems based on ML techniques according to the customer's buying power, taste and past order history. Purchase history drive the soul of these recommendation systems. The simplest of the price model called dynamic pricing model have been presented in [2], that predicts the possible sale purchase of the products based on the past sale data, according to the allocation of the dynamic price range group, to each individual customer and finally

predicting the likelihood of the products to be purchased by a particular customer. The set of products offered to the customers are selected according to their buying power or range of dynamic price group already assigned using k means-clustering technique. Finally, a binary linear-logistic regression trained classifier is called upon the test data to predict the product purchases by the potential customers.

1.5.5 Social Media Sentimental Analysis

Sentimental Analysis on the twitter tweets, a data mining and linguistic rules aggregation approach, is the form of learning algorithm that could predict the types of responses one holds towards others.

1.5.6 Online Credit Card Fraud Protection

Online Credit transaction merchandisers are currently employing machine learning techniques to detect and predict spam cases based on previous cases. This helps improve the service of these credit merchandisers and the satisfaction level of their customers.

1.5.7 Span Email Filtering

Email spam filtering is part of the classical machine learning example presented even today for the general understanding of the classification of the spam filtering from incoming emails. This helps reduce the human efforts required to check one email at a time and can safe guard the hacking prone PC's to be safe to some extent. Simple filtering techniques use basic decision tree-based approach while complex anti-viruses may employ the traditional-hybrid algorithm combinational approach.

1.5.8 Network Intrusion Protection

Since the beginning of the wireless local area network (WLAN) era, the concerns about the general network security and intrusion have long been discussed. Now the latest machine learning approaches are being made to detect the likely causes of the different types of the network intrusions. Intrusion detection is as highly accurate as the past intrusion data provided for the learning classifier to predict it. Detection of r2I, u2r, network probes and DoS Attacks require asset of different salient feature based trained classifiers. Attack data training features can be host based features or the network based, depending upon the type and the level of intrusion attacks [3]. Recently increased interest in the machine learning in general is since the classification, data-mining and prediction phenomenon have become more popular. Things like organisational developments, big data produced by big organisations, technological advances in the computing powers, more and more advanced CPU/GPU architectures and above of all ever-getting cheaper mass data storages have sparked the researcher's interest. This brings the researchers to get closer to use the gathered data and extract thoughtful insights from it.

1.6 Commonly Used Machine Learning Algorithms

Some of the most useful ML algorithms now-a-days, with the brief of what they are used for are listed as follows:

1.6.1 Artificial Neural Networks

A neural network (NN) is a form of popular classifier that represents the simplest human brain and mimics it's decision making power. The artificial neural networks (ANNs) formed of combinations of perceptron's connected in the form of different layers. Neural networks represent the complex input and output correlations that are in other ways difficult to learn and mimic in the real world. The weights assigned to each individual perceptron acts as the strengthening chain that controls the flow of the information in the form of weight values and the activation functions. Optical Electrical circuit recognition is one of the many applications of the ANN. To understand the symbols in the electrical circuit diagram, different number of ANN layers, activation functions are used to learn variety of

possible unique variation techniques in the training data. The test data is then utilised to measure final model performance based on its true prediction and classification power of hand drawn electrical circuit components [4]. A more general neural network architecture with deep hidden layers is shown in figure 1.1 with the generalised model equations given by equation 1.1.

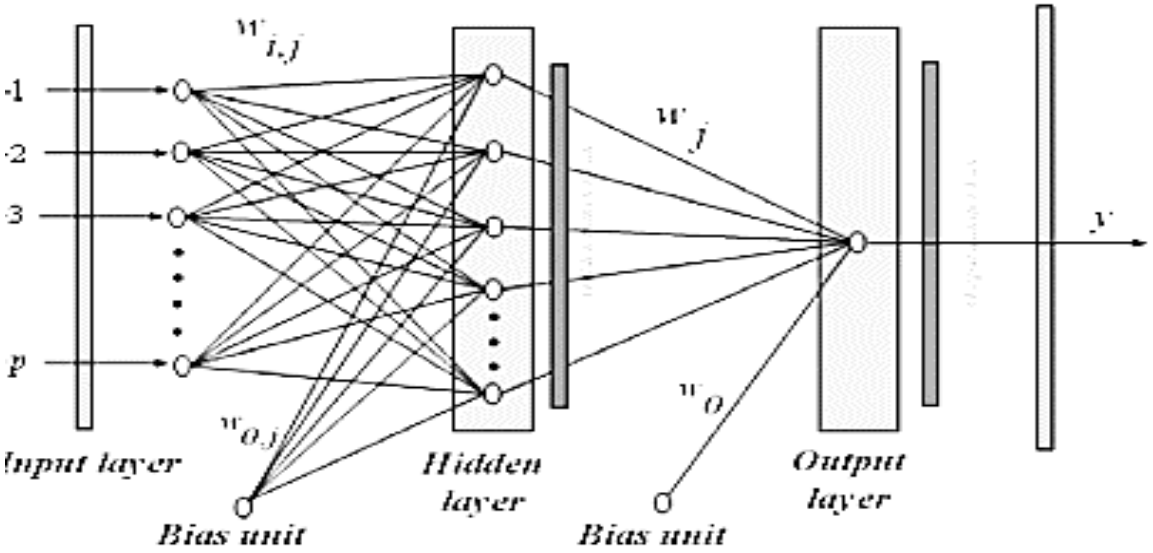


Figure 1.1 Multi-Layer Perceptron (MLP) Network.

$$y_t = w_0 + \sum_{j=1}^{q} w_j . g(w_{ij} + \sum_{j=1}^{p} w_{0j} \cdot y_{t-i}) + \varepsilon_t \qquad (1.1)$$

Given that ε_t is the bias term for each parameter calculations at different levels of layers.

1.6.2 Decision Trees

Decision tree is used to classify data in the classes inferred from the data itself. Dynamic Decision trees could be a more complex implementations that forms new classes or branches according to the data fed in real-time [5]. A traditional transport analysis solution could be a more fitted approach to know the decision trees in practice. In the case of transport service providers they need to form the solutions for the basic problems like: Real-time decision support, Handling incomplete data and human aware decision making powers [5]. To improve the quality of transportation planning machine learning techniques are used to read the rules from the data to provide offline planning solutions. Time consumption is a big issue in real-time planning systems by the operators. To allot the slot for the incoming request traditional optimal solutions are found to have a greater error rate in terms of the cost to avail that slot. So, an instant decision and feedback is provided to the customers by the system.

1.6.3 Other ML Techniques

Other most commonly used well established ML classification and prediction techniques include random forests (RF). Variable selection may be used for deep data interpretation, exploration and understanding. RF has shown better performance under different variable selection techniques since it reduces the correlation effect by ranking them in a special way of their importance [6].

Associations and sequence discovery in conjunction with other tools helps classifying the data. For an incoming data to classify it against the already existing data, it is necessary to transform the data records into ontology-based event graphs. These graphs are visual representations of event sequences through time. This mapping technique in terms of events would help in resolving data conflicts among aggregated records plotted in the form of event [7]. In an analogous manner the hybrid artificial neural network (ANN) - support vector machine (SVM) are put to use to forecast the building energy consumption with the ever growing human population [8]. Nearest-neighbour mapping, K-mean clustering, self-aware-organising maps, local optimal search techniques (genetics algorithms), expectation maximisation, Bayesian networks, principle component analysis (PCA), kernel density estimation, singular value decomposition, gaussian mixture models, sequential covering rule buildings are some of the developed ML algorithms that are often used singularly or in combination with other algorithms on a series of datasets to find the optimal solutions in terms of classifying, making predictions and fetching useful insights out of data. A breakdown of supervised techniques standardised, and basic ML techniques used in the literature previously are shown in the figure 1.2.

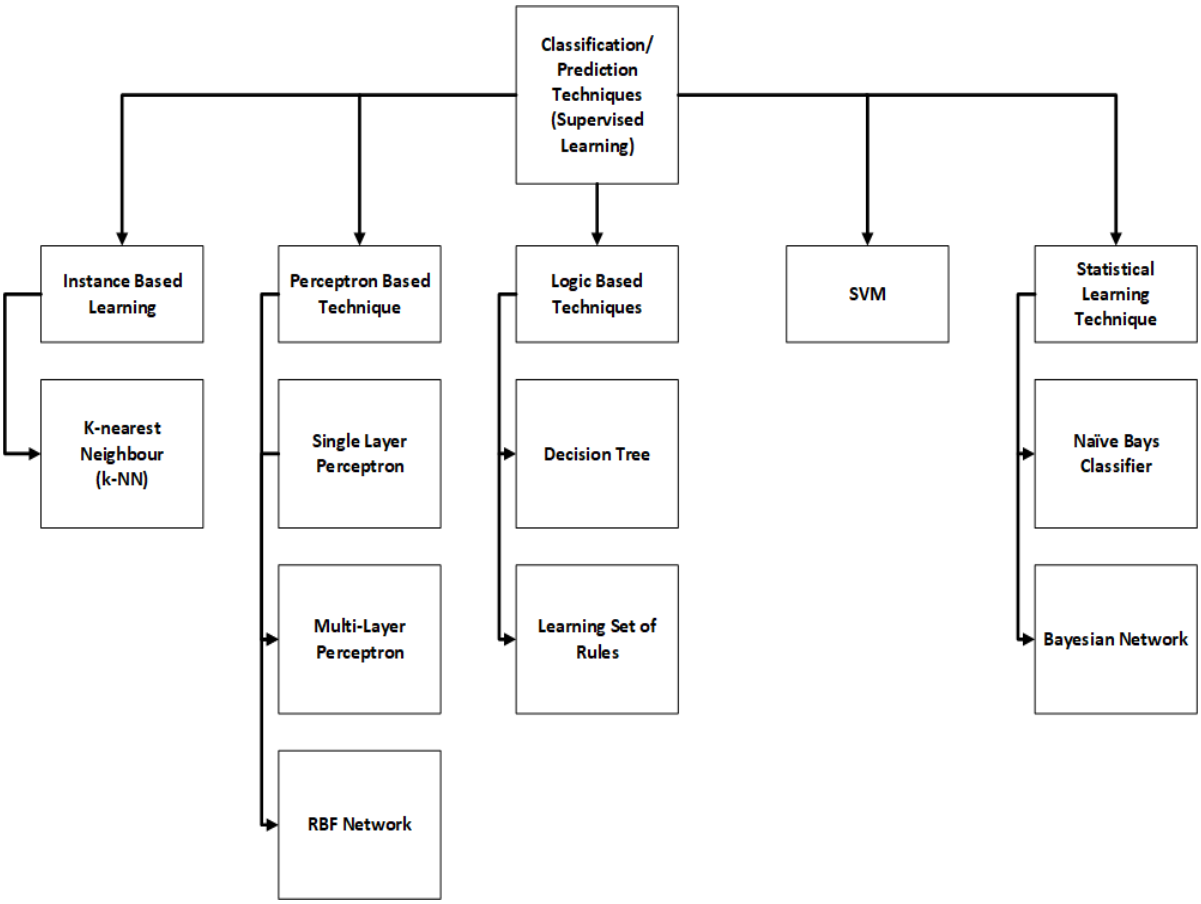


Figure 1.2 Supervised machine learning prediction algorithms breakdown.

Artificial Intelligence and Machine learning have become a popular subject in almost all the applied sciences in modern times. The single in hand capability of the AI and ML is its ability to generalise the behaviour of the process by and large through a large set of data gathered under various conditions, we call it Big Data. Different state of the art ML algorithms been developed over the period of the time that are considered as benchmarking standards. All newly proposed algorithms are bench marked against these standard algorithms in performance and accuracy. A dive into literature discussed in later chapters shows that ML can be developed or tuned for various parameters to

generalise the behaviour of data whether non-supervised or supervised and can be catered to a specific dataset or data driven application. The algorithm trained for one dataset may not be a feasible solution for the behavioural classification or prediction for another dataset.

1.7 What is Smart Transportation?

Transportation operations generate a lot of data on daily basis. Data generated by smart electronic ticket machines (ETM) and their backend servers consists of vital information from the commercial transportation providers. As deep and manual inspection is not enough, if at all, it will fail at very initial level due to the vastness of the data. There's a high need for a detailed insight in to the raw data.

1.7.1 From Commercial Transport Operators Point of View

The raw dataset contains a useful lot of information in terms of features: Bus departure time at all stops along the route, passenger dwell time (DT) at each stop, type and number of tickets used at each stop, smart tickets vs the cash transactions, concessionary passes and then there are other indirect features that can only be discerned by manipulating the known visible features in the dataset. The raw features in the dataset that are more decisive in machine learning behaviour may not be apparent visually at all the times, and there is always a need to manipulate them in one way or the other to get the indirect features that we are interested in. Figure 1.3 shows a typical wait time division taken into consideration from a commercial transport provider's point of view.

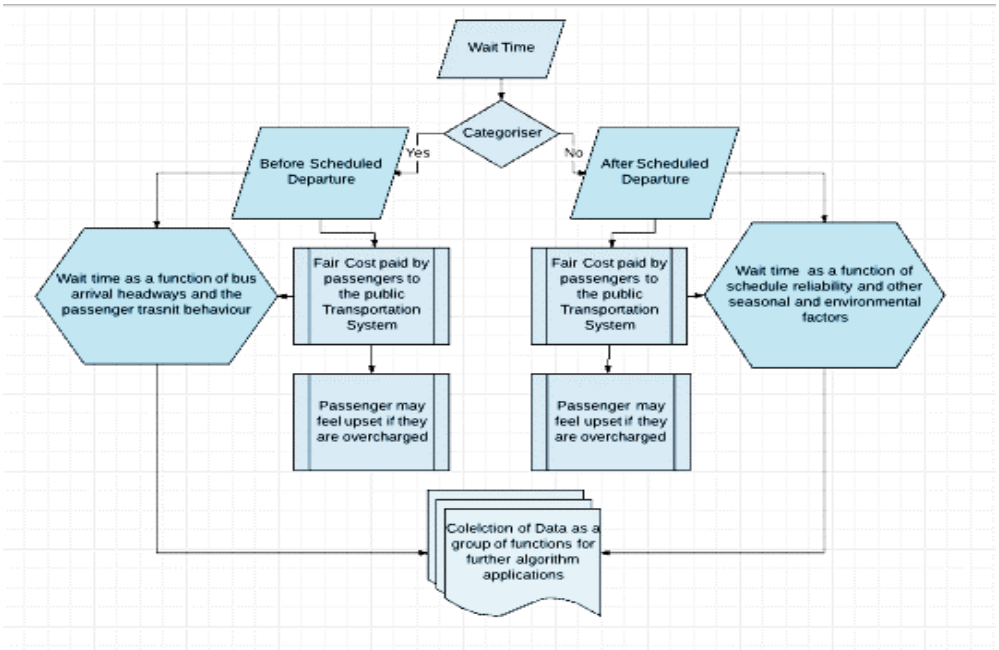


Figure 1.3 Typical Wait categories as seen by the transport operators.

An example of the sample smart transportation model which utilises the use of ML algorithms to model the bus deviation behaviour, let's say, is given in its general form as:

$$Bus\ Deviation\ Prediction\ Proposed\ General\ Model\ based\ Solutions = Artificial\ Intelligence\ (ML) + Statistics\ Models \quad (1.2)$$

Equation 1.2 is the general form of the model that describes mathematically how the practical ML architecture looks like for the bus deviation predictions scenarios. Equation 1.2 consists of the terms: Artificial Intelligence (AI) techniques like smart vision, data pre-processing, Internet of thing devices and ML like Neural Networks (NN), K-mean Clustering (KC) etc... The clarification of these terms would be much clearer along the way to the algorithm development and in the final phase of the thesis implementation.

The Aim of this research work is to adapt an AI algorithm catered specifically towards the development of the transportation problem solutions that could use the existing algorithms for their better performance. Multi modular, scheduling, improved and seem-less mode of travel based on congestion and real time travel time estimations, are the tipping points to be aimed for, for the smart transportation system as for the future development.

1.7.2 Congestion as a Cause of Flow Restriction

An Intelligent transportation system consists of many smart processes that adopt a modular approach but work in harmony with each other. Understanding vehicular traffic congestion is a key for effective mobility and high-quality traffic management and safety systems. The resulting traffic congestion have counter effect on traffic flows in networks. A more empirical approach tells us that the congestion on the road happens due to a sudden breakdown. The vehicle speed decreases sharply and the vehicle density increases instead in what was initially a freeway traffic road. Subsequent research pin points the very complex spatio temporal behaviour of the traffic networks. There is a need for a traffic model to explain the empirical features of the traffic breakdown and the resulting congestion. To explain this a huge number of model and theories have already been developed. The aim of this thesis is to study and discuss the traffic flows using conventional analytical and ML techniques and to build upon them a more accurate if not efficient model, to predict the phenomenon (flows) in real congested traffic. Thus, an assessment of modelling approaches to predict the traffic breakdown is need.

1.8 Thesis Structure

Different chapters with section and subsection are devoted to building the scenario from the small problem definition to the possible questions and proposed mythology, solutions through to the state-of-the-art machine learning model architectures and concluding the thesis with the conclusions and possible future work. Rest of the chapter in the thesis are organised as follows: Chapter 2 presents the comprehensive subject review regarding the conventional statistical, machine learning and deep learning approaches for the prediction of traffic road parameter forecasting. Chapter 3 presents the state-of-the-art traffic flow prediction frameworks. Chapter presents the theoretical details of the chosen and implemented architectures using in this thesis. Research methodology is presented in chapter 5. While chapter 6 discusses the experiments, results and evaluation of the proposed frameworks quantitatively. Finally, chapter 7 concludes the thesis with the final say on the prediction model performances and provides the future direction to be built upon this thesis.

2. Review of Traffic Flow Prediction Methods from Traditional to the State-of-the-Art Techniques

2.1 Introduction

In the previous chapter a general introduction of the traffic prediction problem was discussed. A more detailed background literature study in presented in this chapter. This chapter is organised in a way that it contains the literature review discussion on traffic parameter prediction studies, with an extensive comparison breakdown of each manuscript studied comparing their adopted approach, selected features for decision making, performance measures along with adopted experimental setup using statistical or data driven machine learning (ML) based algorithmic models, sample algorithm test simulations and results as reported in the original manuscript. Further in this chapter review of applications for data prediction in the general engineering domain closely related to those of traffic engineering are also discussed. Further, based on the thorough background study an understanding of the state of the art conceptual and implemented frameworks is envisioned at the end of this chapter which highlights the key research gaps in the existing literature.

2.2 Aims

Chapter 2 presents the background detailed study on road flow and closely related traffic travel time inference models and approaches. Later, this chapter aims to synthesise the types of statistical and machine learning methods presented in the literature study. Finally, the identified gaps trigger the selection of best algorithms for our model development that are further discussed in the next chapter 3.

2.3 History and Short Overview of Traffic Flow Analyses and Predictions from Literature Study

A survey of the recent literature suggests that many authors have contributed their well enough in the field of traffic Incident analysis, prediction and their relation in connection with the traffic congestions. A simple visualisation based approach to show traffic incidents from the past data as map overlay in the form of dynamic radial circles has been given in [9]. Traffic Origins are different coloured circles, each representing different road conditions i.e. heavy traffic, breakdowns and the congestions that are plotted on the map [9]. The traffic origins are the visual descriptors of the location of the incident, heavy traffic flow and the breakdowns whereas their radius determine the vicinity in which the traffic would be affected in one way or the other [9]. Once the area is cleared the circle recedes back and eventually vanishes at the central point of their origin [9]. According to [9], the traffic origin visualisation technique helps better in determining the effects that a cascaded accident or constricted traffic flow could potentially have on a particular road in a traffic network. From literature review traffic flows forecasting can be broadly classified into two distinct categories, Parametric approaches that are based on statistical methods for time series forecasting. Knowledge of data distributions are usually assumed in these approaches. These traffic process-based prediction methods mostly employ traffic systems simulations, road activities and driver behaviour parameters as part of the simulation process. The macroscopic traffic prediction models are based on the vehicular traffic flow analogies with fluid dynamics [10]. The major advantage of using the macroscopic simulations for traffic predictions is that in such methods traffic control parameters (e.g. delay at traffic lights, average time spent on bus stops etc.) can be used in the predictions process and the better understanding of the real traffic environment is achieved based on the locations. On the other hand, the disadvantage of

using much macroscopic prediction techniques are the complex parameter estimations and a real struggle to generate close to real world simulation test environment. Also, the predictions are highly influenced by the quality of the estimated traffic parameters [11]. Both the statistical ML and macroscopic approaches are useful for the ideal traffic flow prediction model development.

This research however, focuses purely on the study for the data driven statistical to complex ML methods for traffic related predictions. The major difference between the ML and conventional analytical method-based models is that ML is considered as a black box which learns the relationships between the inputs and the outputs to predict traffic variables. While ML models are complicated to optimise for the learning, but they are less complicated and computationally efficient to calculate the final prediction once trained. Continued training allows ML models to adapt to the changing behaviour displayed in the data. A detailed description of the selected models can be found in the section 4. According to the literature, for comparison to be meaningful the same traffic data needs to be used for both the statistical and computational learning models. However, such a comparison of models across the literature with same data used in different comparison scenarios is difficult to be found.

2.4 Study of Factors Influencing Traffic Prediction Models in the light of the Literature Review

There are many factors that affect the traffic flow predictions for a model. Apart from hyperparameters of the models some of the factors are the context in which the input traffic parameters are treated, sample resolution of the input data, prediction steps, the relation between the different traffic parameters being used and the spatial based temporal dependencies hidden into the traffic variable data. Further seasonality and trend in the time series data can also influence the prediction performance. Each of these important factors are reviewed in the following subsections:

2.4.1 Context of Implementation for Road Traffic Predictions

According to the literature review traffic parameters are predicted for two main distinct types of roads that constitute the context of implementation for prediction models. One is the highway, freeway and motorways and the other being the urban road connecting roads. The major difference between the two is that urban or connecting road traffic dynamics are more difficult to understand due to uncontrolled connections and variable sized intersections. As highways form the backbone of the major long travelling road structures so the prediction models for highway predictions are majorly used for ITS applications [12]. Some examples of road predictions made in the context of highways and motorways can be found in [13][14][15][16][17][18]. The prediction models employed for the connecting and arterial roads are uncertain and structurally more complex involving more parameters and the urban networks lack the deployment of data acquisition equipment which is mostly on highway and freeways. Some examples of traffic prediction studies made in the context of urban networks can be found in [19][20][21][22][23][24][25]. One of the aims of this thesis is to find the best performing prediction model for the highway network roads.

2.4.2 Input variables for Traffic Prediction

Choice of variables may be critical and difficult for forecasting, but it is directly related to traffic flow forecasting model performance and efficiency [26]. Sometimes not just the raw feature values are used rather indirect methods of information extraction maybe employed like mutual information based on entropy theory has been used in [26]. The variable parameters that are commonly considered for the forecasting models include, traffic flow volume, travel time and speed data. Such variables data is collected with onsite sensors using loop detectors and or laser sensors. According to [27], traffic flows along with the traffic density and speed parameters are used to model complexities

in traffic flows in a piecewise switched linear manner, describing the model as an aggregated set of partial derivatives of the involved parameters [27].

General road delays and blocked lane duration (BLD) in [28], a queueing based model is compared considering the road delays, incident severity and road incident locations as the input parameters. For the proposed performance measure using decision trees (DT) blocked lane duration (BLD) and general road delays from a particular incident are considered in [28] as the final input parameter. Finally, the model quantifies the average delay per number of cars as the final performance parameter measure for the effective Traffic Incident Management (TIM) System. The fact that the proposed DT [28] does not require additional data makes it favourable for traffic predictions. Similarly, lane blocking incident data is used as an input parameter in automatic traffic incident detection system by considering the hybrid of Time series analysis (TSA) and machine learning (ML) techniques utilising the theory of fault diagnosis [29]. TSA is used to predict the normal traffic based upon the past normal traffic data. Likewise, ML is used to detect the traffic incidents from real-time behavioural learning, already existing normally predicted traffic data and the differences between the two [29]. The proposed approach in [29] claimed to have the better detection rate and lesser mean time for detection of incidents under the constant condition of same false alarm rate (FAR), when compared to other standard algorithms. Traffic features i.e. acceleration and other action based parameters recorded during the driving are used as input parameters and clustered together using k- mean learning in a supervised learning fashion, further tested to categorise the overall vehicle driving behaviour and later used to predict the potential traffic accidents in a driving simulation analysis [30]. Another automatic traffic incident severity classification system comparing different machine learning techniques has been presented in [31]. Input data not only contains the standard traffic incident parameters (e.g. incident location, date, time and affected lanes) but incident severity levels are also considered as an important deciding parameter to issue control commands. The proposed ML model approach in [31] is developed and tested to help manage the traffic incident management controllers to automate the network traffic control process instead of just doing them manually and breaking the information for classifying it into the pre-determined categories, and to minimize the effects an incident could have on the network [31].

2.4.3 Effects of Using Purely Machine Learning Approaches

Considering purely ML approaches, in [32], fuzzy logic–deep neural net learning (FDNN) approach have been adopted as an effort to detect the traffic road incidents, considering the traffic flow features in an urban environment. The proposed model in [32] is focussed primarily on the deep learning neural network with the aim to learn inherent spatial-temporal features from traffic flow data used an input parameter. Stacked Auto-Encoder (SAE) based layer by layer pre-training and fuzzy logic is used in conjunction with the back-propagation algorithm to control the over shoot of the learning rate and to avoid data over-fitting. Produced simulation results show an improved incident detection rate, considerably reduced false alarm rate and less learning time of FDNN compared to simple DNN [32]. Likewise, Convolutional Neural Network (CNN) based deep learning approach like in [34], with the aim to learn the spatiotemporal traffic dynamics by forming the time-space matrix images from the traffic flow data, for the network wide speed predictions, has been proposed in [35]. The results show an overall performance improvement of 42.91% with an acceptable execution time compared to the other widely used standard algorithms for predictions namely; ordinary least square method, k-nearest neighbours (KNN), artificial neural network (ANN) and long short-term memory networks neural networks (LSTM NN). The future extensions, as given by [35] could consider the possible combinations of CNN and LSTM-NN for feature learning and prediction for overall better network wide speed and flow prediction accuracy. Similar, deep learning Stacked Auto Encoder (SAE) model for the traffic data spatial temporal feature extraction and learning using a simple logistic Regressor activation

function based outer layer, for the network wide traffic congestion prediction is given in [36]. The SAE developed model prediction accuracy was compared with other algorithms performance accuracy which include; back propagation neural nets BP NN, random walk (RW) forecast method, support vector machines (SVM) and the radial basis function neural network (RBF-NN) [36]. An overall prediction accuracy improvement of 93% was shown by the SAE model when tested for 15 minutes average duration, traffic flow rates larger than 450 vehicles and different number of hidden layers, disregarding any other road parameters (weather, speed, density, traffic incident) had a direct or indirect effect on the traffic volumes [36]. Similarly, in [19], individual vehicular feature based behavioural study of four prominent features (speed, acceleration, and lane changing ratio) for pre-effective traffic incident detection are used in an urban environment using the mobile sensors data instead of fixed road sensors. Four different road scenarios of normal and incident traffic conditions, with having each variable passed through the Kolmogorov-Smirnov test (K-S). Final consideration of the empirical cumulative distribution function (ECDF) with an initial null-hypothesis revealed the relative importance of these variables in effectively detecting different types of traffic incidents [21]. However, [21] does suggests that for higher vehicular flow volumes (>500 veh/h) the chosen variables do not play a very significant role in differentiating between the normal and incident road conditions thus the better implementation of the incident detection system (IDS) must also consider the traffic volumes and flows.

Data Driven approach with GPS collected speed data to predict the traffic congestion evolution using spatial-temporal features learned using recurrent neural network and restricted Boltzmann machine (RNN-RBM), is given in [37]. To assist transport professional in congestion prediction and planning, [37] ruled out the common assumption that traffic flow dynamics over the networks follow the power law distribution all the times, which is generally assumed due to the lack of enough traffic sensors data. Further the proposed RNN-RBM is tested on different road networks and compared to the performance accuracy of SVM, conventional neural networks and the sensitivity analysis done with different speed thresholds, [37] reports an overall prediction of accuracy of 88%, training accuracy of 95% and finally the overall algorithm execution time to be less than 6 minutes. Further, the results are visualised on the GIS map for congested road planning. Proposed future recommended techniques includes are the model pre-training using hessian-free optimisation method for parameter rational initialisation and spatial road interaction to be considered for more precise training and prediction accuracies [37]. Support vector regressor (SVR), Bayesian classifier and linear regressor are used as main algorithms for the traffic flow estimations by predicting spatiotemporal traffic features in [38]. Traffic flow input parameter data and its relations are models into graphics form using 3D Markov random field in spatiotemporal domain. Based upon the cliques of cones obtained in the spatiotemporal domain and the overlapping between the successive cones, multiple SVR's and Linear Regressor were used to predict the dependencies on that defined cone [38]. Finally, to predict he traffic flow for future time stamps, the speed level was found by decreasing the energy function [38]. SVR based prediction (84.64%) showed higher accuracy than the linear based approach (~76%) when tested on the test data with multiple cone zones that are not even complex geometries but also represents noisy traffic flow conditions [38].

In [39], a real-time distributed VANET approach for not just detecting the road incident-based congestion in the urban environment but also to classify congestions into different types into recurrent and non-recurrent congestions NRC (road incidents, work zones, special events and adverse weather conditions). The proposed model considers the spatial temporal causality (cause/effect features) measures with the training data produced synthetically from a real case study [39]. The algorithms tested with their prediction accuracy include: Decision tree Classifier (DT) (88.63%), Naïve Bayes (87.83%), random, Random Forest (89.51%), and boosting technique (89.17%). Future add on

techniques as suggested in [39] can include a voting process, a likelihood evaluation or a model to value the density of information in the data. Also, in case of real-world data with the connected vehicles strategy, Ann Arbor automated vehicle operational test can also be performed in a test environment congestion estimation. Another novel statistical approach has been discussed in [18], to detect traffic congestions from the vehicle flow/density data. The unique and advance statistics model developed in [18], uses the piecewise switched linear (PWSL) traffic model to describe the traffic flow dynamics from the data, with the leftover deterministic features from PWSL fed to exponentially weighted moving average (EWMA) chart to detect traffic congestions. EWMA performance was degraded in the presence of the real noisy data so [18] suggests the multiscale filtering before the application of EWMA.

A detailed study of literature also tells us that a fair number of researchers have given their contribution to mitigate the traffic road congestions. Reinforcement learning technique had been used to control the variable speed limits to control congestions at the recurrent freeways bottlenecks [40]. A Q-learning (QL) model in offline and variable speed limit (VSL) model in online mode has been used in conjunction with one another [40]. The VSL controller agent, which works in an online mode, is already trained with the optimal speed limits to be observed under different traffic conditions. The modified cell-based traffic model is used to evaluate the prediction-based control of trained VSL controller on a freeway recurrent bottleneck [40]. According to the paper [40], the proposed QL-VSL controller-based strategy showed a much-improved congestion optimising (travel time reduction of ~49.34% in stable conditions, ~21.89% in fluctuating traffic conditions) model than a simple feedback based VSL controller on a long-term performance basis. According to [40], the future development may include the, sophisticated prediction models, with the combination of further traffic flow estimation techniques with RL-VSM model for better performance, bottlenecks related to the incident, lane reduction, work zone, event related, merged positions of road links or even multiple bottle necks together, and other single or multiple reward functions, as part of better RL, can also be considered. Furthermore, the future recommendations in [40], are of the view to include advanced deep learning techniques as part of the VSL and overall model development strategy to improve the traffic congestions.

2.4.4 Input Data Resolution for Traffic Prediction

Input data resolution is also considered an important element for the consideration of traffic model performance. Since it can affect the quality of information to be extracted by the prediction models. According to the recent survey of traffic flow prediction mechanisms [33] the time interval or data resolution range varies from 30 seconds or one minute [34] as the least to 1 hours or 60 minutes to the most peak resolution time considered. The traffic data should be available with the data resolution that is sufficiently enough to capture the traffic dynamics by the prediction model. Along with the mentioned literature, an overview comparison of the considered time interval for the data used in the respective proposed model along with their references are presented in table 2.1. Higher recorded data resolution means more noise and less temporal resolution means the loss of data. The resolution granularity of the data should be controlled dynamically according to the prediction model. Due to the measurement instruments used for recording the traffic data it tends to be fixed in most cases and the change in traffic conditions could be missed without increasing the recorded data resolution in such a case.

	Title & Reference	Data Resolution / Time Interval
1.	On the capacity of bus transit systems [35]	1-hour recorded data resolution
2.	Traffic origins: A simple visualization technique to support traffic incident analysis [9]	15-minutes recorded data resolution
3.	Traffic incident data analysis and performance measures development [28]	5-minutes aggregated data
4.	A Hybrid Approach for Automatic Incident Detection [29]	1-minute data resolution
5.	Traffic Flow Prediction with Big Data: A Deep Learning Approach [36]	5-minutes aggregated data
6.	Large-scale transportation network congestion evolution prediction using deep learning theory [37]	2-minutes recorded data resolution 5, 10, 30, 60 – minutes aggregated data
7.	Predicting Spatiotemporal Traffic Flow based on Support Vector Regression and Bayesian Classifier [38]	30-seconds & 1-minute recorded data resolution 1– minutes average aggregated data
8.	Effective Variables for Urban Traffic Incident Detection [19]	1-second recorded data resolution
9.	Automatic classification of traffic incident's severity using machine learning approaches [31]	1-day recorded data resolution
10.	Fuzzy Deep Learning based Urban Traffic Incident Detection [32]	100-seconds aggregated data
11.	Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction [39]	1-minute recorded data resolution 2-minutes aggregated data
12.	Distributed Classification of Urban Congestion Using VANET [40]	0.1-second recorded data resolution
13.	An Efficient Statistical-based Approach for Road Traffic Congestion Monitoring [16]	1-second recorded data resolution
14.	Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks [41]	30-second recorded data resolution 30-second to 5-minute aggregated data
15.	Using LSTM and GRU neural network methods for traffic flow prediction [42]	30-second recorded data resolution 5-minutes aggregated data
16.	Short-Term Traffic State Prediction Based on the Spatiotemporal Features of Critical Road Sections [43]	2-minutes recorded data resolution 2 to 15-minutes aggregated data
17.	Deep Transport: Learning Spatial-Temporal Dependency for Traffic Condition Forecasting [44]	5-minutes recorded data resolution
18.	Short-term traffic flow prediction using seasonal ARIMA model with limited input data [45]	1-minute recorded data resolution 10-minutes aggregated data
19.	Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification [46]	15-minutes aggregated data
20.	Spatiotemporal Patterns in Large-Scale Traffic Speed Prediction [47]	5-minutes aggregated data

21.	Bus Dwell Time Prediction Based on KNN [48]	--------
22.	A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting [49]	5-minutes recorded data resolution
23.	A distributed spatial–temporal weighted model on MapReduce for short-term traffic flow forecasting [50]	Variable-minutes recorded data resolution
24.	Urban Traffic Flow Prediction System Using a Multifactor Pattern Recognition Model [25]	15-minutes aggregated data
25.	Prediction of Bus Travel Time Using ANN: A Case Study in Delhi [51]	30 to 60 minutes recorded trip data
26.	Long short-term memory neural network for traffic speed prediction using remote microwave sensor data [52]	2-minutes aggregated data

Table 2. 1 Data Resolution Used by various Prediction Models Across Literature.

2.4.5 Prediction Steps in Traffic Flow Prediction

The future time steps or intervals across which the prediction model predicts is referred to as the prediction step or prediction interval or prediction horizon. A generally accepted rule is that the prediction accuracy degrades with an increase in prediction horizon [47]. Although multi-step predictions are so common in the prediction model discussed across literature, but it comes with the prediction model accuracy compromise. In this research the intension is to do both one step and multi-step ahead predictions. An overview comparison of the considered prediction steps for the respective proposed model along with their references are pin table 2.2

	Title & Reference	**Prediction Types**
1.	On the capacity of bus transit systems [35]	multistep predictions
2.	Traffic origins: A simple visualization technique to support traffic incident analysis [9]	multistep predictions
3.	Traffic incident data analysis and performance measures development [28]	multistep predictions
4.	A Hybrid Approach for Automatic Incident Detection [29]	multistep predictions
5.	Traffic Flow Prediction with Big Data: A Deep Learning Approach [36]	multistep predictions (15, 30, 45, 60 minutes)
6.	Large-scale transportation network congestion evolution prediction using deep learning theory [37]	multistep predictions (4 -per day)
7.	Predicting Spatiotemporal Traffic Flow based on Support Vector Regression and Bayesian Classifier [38]	multi-predictions model
8.	Effective Variables for Urban Traffic Incident Detection [19]	multi-Step Multi-variable predictions
9.	Automatic classification of traffic incident's severity using machine learning approaches [31]	one-Step Multi-variable predictions
10.	Fuzzy Deep Learning based Urban Traffic Incident Detection [32]	one-Step Predictions
11.	Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction [39]	multi-Step Multi-variable predictions

12.	Distributed Classification of Urban Congestion Using VANET [40]	multi-Step (5-15 minutes) multi-variable predictions
13.	An Efficient Statistical-based Approach for Road Traffic Congestion Monitoring [16]	multi-Step multi-Scenario predictions
14.	Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks [41]	multi-step predictions
15.	Using LSTM and GRU neural network methods for traffic flow prediction [42]	one-step (5-minutes) multi-model predictions
16.	Short-Term Traffic State Prediction Based on the Spatiotemporal Features of Critical Road Sections [43]	multi-step predictions (15 to 40 minutes)
17.	Deep Transport: Learning Spatial-Temporal Dependency for Traffic Condition Forecasting [44]	multi-step (15, 30, 45, 60 minutes) multi-model predictions
18.	Short-term traffic flow prediction using seasonal ARIMA model with limited input data [45]	multi-step (15 minutes interval) predictions
19.	Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification [46]	single-step (15-minutes) multi-level predictions
20.	Spatiotemporal Patterns in Large-Scale Traffic Speed Prediction [47]	multi-step (10- per day) multi-model predictions
21.	Bus Dwell Time Prediction Based on KNN [48]	multi-step (each for stops) predictions
22.	A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting [49]	multi-step predictions (per hour) multi-model predictions
23.	A distributed spatial–temporal weighted model on MapReduce for short-term traffic flow forecasting a distributed [50]	multi-step predictions (per day) multi-model predictions
24.	Urban Traffic Flow Prediction System Using a Multifactor Pattern Recognition Model [25]	multistep prediction (15 or 30 minutes)
25.	Prediction of Bus Travel Time Using ANN: A Case Study in Delhi [51]	multistep prediction (12 intervals or 12 trips)
26.	Long short-term memory neural network for traffic speed prediction using remote microwave sensor data [52]	one-step prediction (2 min or one-time interval)

Table 2. 2 Data Prediction Interval Strategy Used by various Prediction Models Across Literature.

2.4.6 Seasonal Effects and Spatial-Temporal Patterns in Traffic Flow Prediction

The temporal and spatial relationship has been widely discussed in the context of traffic flow and general traffic predictions. Many researchers have mentioned well enough in their research [53][49][54][38][47]. The goal has always been to exploit the road traffic temporal data with respect to the spatial features. Traffic time series exhibit seasonal and periodic behaviour when analysed for the trendiness in the series data. Free flow ways and motorway roads have a strong connection with respect to the spatiotemporal features [50].

2.4.7 Various Road Conditions in Traffic Flow Prediction

Traffic flow are affected by different conditions the simplest of these are road side incidents with different levels of severities. Different researchers have tried classifying different road conditions in

their own ways. Generally, the traffic road conditions can be divided into two main categories: Normal and Abnormal traffic conditions. In [55] two extreme traffic conditions are considered for the proposed deep LSTM model i.e. peak hours and the post-accident conditions. Likewise, heterogenous traffic conditions are the focus for the proposed neural network (ANN) based model [15]. Other researchers have also focussed on exploiting the conditioning based traffic data for their proposed prediction models [44][50].

2.5 Traffic Predictions in Other Domains Closely Related to Traffic Flow

Passengers travel time, waiting time at bus stops are the most effectual measure of the public transportation as a substitute to automobiles [56]. Waiting time have been regarded as the most unlikely time to be avoided by the passenger during their transit journey which maybe an indirect measure of the traffic congestions. The waiting time is usually more than the in-vehicle times. This is particularly true in terms of the urban transit mode [56]. The main purpose of this section is to study and investigate in literature the waiting times of the passengers at the bus stops, general traffic road travel delays and to investigate the relationship between the two considering other predictive variables related to congestion studies and bus headways time and delays which are an indirect measure of the relative traffic flows. The idea is to focus on machine learning algorithms for passenger wait time and general road traffic travel time predictions.

Literature review suggests that the passenger wait time at the bus stop have been generally grouped into three main categories, based upon the time logic estimation or inference: waiting time with microscopic simulation level models, which involves different types of buses stopping at various types of bus stops [57]. The simulation model developed in [58] simulates stop operations, their working capacity with recorded delays that results in queues at the bus stops. Very basic level simulator has been used to create a virtual environment that simulates the cases under study in [57][35]. [57][35]. In the second category, the transit travel studies focus mainly on finding the statistical relationship between the actual waiting times and the ones recorded by the passengers [59]. And lastly the waiting times at bus stops are inferred by manipulating the vehicle's arrival information [60][61]. The wait times at the bus stops that the passenger have to encounter can be deduced using a classical probabilistic approach and queueing model [62]. The Queueing Model takes into effect the stop arrival headways, bus or service operating times and the total number of services along throughout the day serving that same bus stop. This is one way to estimate the passenger waiting time at bus stops.

The bus spends as much time at the stop berth as the passenger dwell time and none during the actual headway until the next bus comes into the berth. So it is rather more practical, to calculate the average waiting time to render an application of vacation queueing model rather than the classical queueing model [63].

According to [64], a more practical approach in order to model the wait time at bus stops is the use of vacation queueing model as the classical queueing model is incapable to model the periods of absence in between adjacent bus headways, which is the main time when the passenger actually waits for the bus to arrive. The vacation queueing theory allows to classify the wait times of the system in equilibrium state into two different types: Wait times at the stop modelled using the classical queueing model without the consideration of the vacant service periods and in addition the second being the derivative from the length of the vacant service period using the actual vacation model. As we know that a stochastic process can be termed as a sequence of events in which the outcome always depends upon certain probability [62]. Markov process is a kind of stochastic process in which the outcomes at any stage depends on just the previous event, outcome possibilities are always finite, and the probabilities being considered for the problem remain constant over time of the event until the next

event happens. In [65] the passengers wait time is estimated to be the duration, from the moment when the bus boarding starts, when the bus door opens up to the moment, when the last passenger boards the bus, and this last passenger instant is then termed as the 'achievement instant'. Considering the achievement instant as the last finite point of the Markov finite chain [62], the Laplace transform of the vacation queueing model helps deducing the wait time in [64]. The arrival rate of the passengers at the bus stops is considered according to a Poisson process in [64].

Bus headway probability prediction model using relevance vector machine (RVM) utilises the time series headways data, travel times and the passenger demands at previous stops [66]. With the relevance vector algorithm in [66], the upper and lower bounds of the probability of bus headway are predicted with up to a confidence level of 95% and the algorithm in general outperforms the SVM, genetic SVM, Kalman filters k-NN, ANN in terms of testing accuracy and confidence levels. Passenger and transit rider behaviour is greatly affected by the reliability of bus headway information. This allows the transit riders to plan their trips more effectively and to transit operators on the other hand to maintain the smooth transit flow by intelligent bus scheduling [66]. Predictions based upon the accelerated survival model proposed in [67] not only estimate the bus travel times down to the bus stops but also estimate the uncertainty associated with it. When predicting the time estimate of the travel times, simultaneous survival prediction models, when compared with the linear regression models showed relatively same root mean squared and (RMS) and mean absolute errors (MAEs) but the survival models on the other hand does much better in stating the uncertainty associated with each prediction [67]. A least square support vector machine (LS-SVM) is utilised to highlight the bus bunching with the headway pattern predictions [68]. The proposed model in [68] captures the bus headway irregularity based on the transit smart card data along with the past headway, passenger demands and travel times data to predict the travel time pattern based bus bunching on the following stops. LS-SVM in [68] exhibits 95% accuracy, more sensitivity and specificity in predicting bus bunching occurrences hence the travel times, when compared with KNN, ANN, RF and gaussian process regression (GPR) algorithms.

Based on the literature study [69], bus travel time predictions are classified as naïve approaches, data driven approaches and traffic flow based approaches. The algorithm proposed in [69] assume the spatiotemporal variations in the travel time data. Most studies just consider either the spatial or temporal data alone. According to the proposed model in [69], vehicle conservation equations are rewritten in terms of traffic speeds instead of flow and density, following a differential approach using the traffic stream models. Godunov scheme was then utilised to discretise the vehicle equation and fed to Kalman filters-based predictor to predict the travel times. Model developed in [69] outperformed the classical average, regression, ANN and simple temporal and spatial methods based on the past data.

Socio-economic conditions, weather conditions, trip specific characteristics including but not limited to, infrastructure facility usage have a great effect on the transit rider's total travel time. Comparison between the passenger perceived travel time and the actual travel time in [70] shows that passengers do perceive the travel time to be greater than the actual time at any stage of the transit journey. Three linear regression based models, when applied on the transit survey data at all stages of a journey to predict the perceived walking, waiting and in-vehicle time do authenticate the effect socioeconomic characteristics and trip stages have on the travel time perceptions [70]. Interval based sampling approach is also thought to be a better approach to estimate the uncertainties in the stop arrival and overall travel times of the buses. It does cover the ramifications of both the late and early bus arriving. Interval based model to estimate the travel time in [34], made use of the similar approach by generating the probability distributions to close to accuracy predictions of the arrival times at the

respective stops. Systematically thought of intervals (8km) based probability distribution of the travel times are found and compared with different distance interval thresholds. Lognormal and normal distributions were found to be the better estimates of the travel time behaviour for before and after the cut-off horizon intervals of 7-8km, respectively [34]. Similar approach of route stop based segmentation have also been adopted in [71] to predict the journey travel time using scheduled time data based on the combination of the queueing theory model and ML decision tree algorithm. According to the snapshot model (queueing theory) in [71], some segments use learning (hybrid-Queueing theory and ML) to predict while other are learning free. But learning free segments based on queueing theory do encounter some prediction outliers which are then effectively identified using decision tree ML based prediction trained on historical data [71].

The travel time predictions are introduced with a lot of variability in the urban environments. This variability in the travel time prediction is a night mare for the transit riders. The variability may arise due to the schedule and design of the transit lines themselves and because of different operators running different trip schedules due to their own needs and demands. Also, the buses using the same lanes as with the other public vehicles also introduce a factor of variability. Through recent advances in the data gathering techniques and technology made it possible to research for the root causes of variability. The experimental justification to the variability have been adopted in [72], in which Automated vehicle counter (VAC) and automated vehicle location (AVL) data is put to test for a basic analytical study. Results from [72], shows a strong similarity between the urban traffic temporal patterns and the travel times and the possibility of more short term travel time predictions in an urban environment.

Recent work [73], predicts the travel times based on the routes recorded traces of GPS trajectories data. In [73], the issue of sparsity in the GPS recorded data have been addressed, as some of the routes may not be even travelled by the GPS equipped vehicles once in the designated considered time. A tensor-based modelling approach is adopted, that models different driver's travel times on different road segments on different scheduled time. Tensor missing values are filled with the context aware decomposition approach with geospatial, historical and temporal data learned from past trajectories and map data [73]. An optimal estimation of the missing tensor trajectory value concatenation is finally used for the considered time slots [73]. Not directly related but a similar parametric learning spatial-temporal hidden Markov model (STHMM) is used in [74] to model the dependencies of different traffic time series GPS tracking data and to infer the future travel costs in a transportation network. Data sparsity, heterogeneity and spatial-temporal correlations are the major driving force for the models to learn the STHMM parameter [74]. In contrast to [74], Spatial-temporal random (STRF) based traffic future conditions, is implemented in [75] for the purpose of dynamic route planning. Travel time is the main factor is future predictions and dynamic route planning. Although both use non-ML techniques in their models for future traffic estimations. Further in [75], estimations of the incomplete data fields estimation done through gaussian process regression technique with conditioning of spatial regression on the intermediate predictions in discrete probabilistic graphical models for better inter dependency explanations though historical and real-time data. Real time traffic speed, flow, travel time measurements and statistical bundles of information is estimation in an IBM Infosphere Streams environment implemented applications, to be accessible for masses [76]. Probability distribution Function (PDF) of arrival times, PDF smoothing and classification techniques have been used for the estimation of the parameters for both online and offline mode utilising the historical sparse, noisy Automated Vehicle Location (AVL) data [76].

Missing time series traffic data imputation technique based on gap-sensitive windowed (GSW) k nearest neighbour (GSW-KNN) have been presented in [77]. There results show a 34% accuracy then

general KNN and benchmarking methods. This method can be used towards our aim to predict travel time from past series data. Short term travel time predictions between two points using conventional feed forward backpropagation ANN algorithms have been put to use in [78]. The data used consisted of two points as start and stop points that detected the vehicle movement through them, and the times were recorded. Not directly related in terms of aims but the mechanism presented in [79], denoising stacked auto encoders works on temporal and spatial factors for traffic data imputation, can be used for the travel time prediction. The proposed model [79], as reported, shows a better performance compared with ARIMA and BP NN models. A geographically weighted regression (GWR) model have been proposed in [80]. The proposed model [80] compares the least squares (OLS) multiple regression traditional models, for the application of forecasting at the train station. The similar approach can be used to forecast the passenger wait time and the total bus travel time. A Bayesian network and neural network based double star modular framework approach have been formulated considering the spatial and temporal relations and to predict traffic network speeds and compared with the seasonal autoregressive moving average model (SARMA) [22]s. Different Time series models provide the priori estimations of predictions to the Bayesian network [22] .

2.6 Various Approaches for Traffic Flow and Congestion Behaviour Modelling and the Associated Limitations in the Light Of Literature Review

Broad division of prediction models across the literature review falls into the following categories:

2.6.1 Parametric, Naïve and Macroscopic Simulation based Approaches:

Conventional approaches that use statistical methods for time series forecasting are normally termed as parametric model approaches. The prior knowledge of data distribution is assumed in parametric approaches. These model approaches mostly perform well in short term forecasting. Also called naïve approaches as they provide simple estimates of the traffic based on average means, weighted average in the simplest form using the previous interval data etc. Parametric approaches require a fixed set of parameters defined as part of their mathematical and statistical equations for example: analytical methods hence exhibit a poor performance generally in long series data structures and for bigger future horizons. Some of the literature gathered popular parametric models and techniques are given in table 3.1 along with their limitations in terms of traffic flow predictions.

Models	Limitations
Autoregressive Integrated Moving Average (ARIMA) Model	Could be used for more than one-time interval predictions but the prediction performance degrades with the increase in prediction horizon.
Seasonal Autoregressive Integrated Moving Average (SARIMA)	Like ARIMA but incorporates the seasonality of the time series. Works best for non-stationary seasonal series as stationary seasonal component in difficult to tackle.
Kalman Filter model	Cannot predict well enough due to stochastic and non-linear approach of traffic data. As simple biases update for the net value is incorporated which is lacks the behavioural modelling.
Auto Regressive Moving Average (ARMA) Model	Better for short term forecasts, same as ARIMA model. Better suited for stationary time series. So, the integrated part has not

	much of the effect on the final forecasts. Performance can be enhanced by integrating the seasonal component to it.
Exponential Smoothing, Simple Smoothing, Complex Time Series Analysis and Filtering methods	Biased significance towards the most recent observations, cannot handle the real-time trends well. Poor performance for unbalanced class or the seasonal components
Weighted average method, Weighted moving average (WMA), Geographical Weighted Regression (GWR)	Importance given to those observations with heavy weights, custom weightiness decision is complex to make in traffic flow data and so the trends filtering.
Mean speed based on two-dimensional linear interpolation, Spatial-Temporal Hidden Markov models (STHMM),	Macroscopic approach, estimated inference for the estimated variable is better in accuracy with higher resolution in data observations.
Piecewise Method, Bivariate Movelets	A pattern library must be constructed based on the data intelligence for this technique to work effectively, difficult to do in multi trend traffic environment.
Gaussian Model for Flow Data Imputation	Additional parameters inferred from data are required e.g. mean, covariance among data, which accounts for bad estimations for data that is already incomplete. Prior data knowledge is difficult to estimate.
Tensor based multi-dimensional modelling, Linear, Polynomial, Power, and Exponential curve data fitting	Modelling approach needs an objective function that considers the trades off between the concerned variables.

Table 2. 3 Summary of Parametric Models.

The problem with most of these parametric approaches is that they can effectively be employed for only one-time interval prediction and cannot predict well enough due to the stochastic and nonlinear nature of traffic data. this can better suit short term forecasts only which are well biased towards the recent observations in the data, thus this makes the parametric approaches incapable of handling real world trends.

2.2.2 Non-Parametric and Data Driven Data Driven Machine Learning Methods:

Models with not fixed structure and not pre-defined number of fixed parameters falls into this category Deep learning long short-term memory (LSTM), gated recurrent units (GRU), neural network (NN), and recurrent neural network (RNN). Non-parametric approaches mostly constitute of the data driven models as well. They utilise the empirical underlying algorithms to provide the predictions. They can assume any assumptions based on the data formation and uncertainty as they estimate the model parameters, a classic example of this approach is the neural networks.

A few years ago machine learning (ML) strategy based traffic parameter prediction algorithms [48] have been utilised. These data driven approaches are also termed as non-parametric approaches. These have been utilised with not fixed structure and not pre-defined number of fixed parameters. The most commonly tested non-parametric approaches for spatiotemporal traffic forecasting includes the k-nearest neighbours (KNN) [49][50] and support vector regression (SVR) [47]. However, these shallow ML algorithms mostly wok in a supervised manner which makes their performance dependent upon the dataset manual features selection criteria.

Models	Limitations
K-Nearest Neighbour (KNN) Models, K-means and Hierarchical clustering, Linear Regression, Random forest (RF).	Exhibits a better performance if data correlation is too low. Also, algorithm performance diminishes significantly in high dimensional data classification. But traffic series data is a highly correlated data.
Support Vector Regression (SVR)	
Back-Propagation Neural Network (BPNN), Multi-Layer Feed Forward Neural Networks (ANN) and variants of NN	Out performs conventional linear parametric models but struggles with time series data during data learning phase for finding the absolute global minimum since there might be multiple minimums for the trendy non-stationary data traffic series.
Recurrent Neural Network (RNN) and it's variants (Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU))	Designed to cope with time series data prediction problems, gives the options to learn across varying time steps at once. Due to the recurrent feedback the relative prediction performance is better than the simple NNs.
Deep Learning Models, Convolutional Neural Network (CNN) with multiple layers	Raw deep learning methods are devised mostly for image learning purposes and need some sort of modifications for time stamp data learning as the dependency might be a big issue due to high data co-relations and scattered trends which in cases of image classification in not so obvious (same group of pixels might appear together that makes the classification job easier). Similar concept is employed to exploit the traffic series spatial data, if input to CNN is structured properly.
Bayesian Networks, Naïve Bayes, and Self-organizing maps	Naïve Bayes, to mine spatiotemporal performance trends at the network level and for individual links. Bayesian networks along with naïve Bayes solve computational complexity by considering correlated features as mutually independent happening events to calculate their probability distributions. Further on link level they can be employed for conditional probability-based flow predictions
Principal Component Analysis (PCA)	PCA is dimensionality reduction technique in first place primarily based on finding the most relevant eigen matrices of data variable. PCA based models might consider those parameters which have less or no decision in final prediction and hence it is not favourable for traffic predictions and mostly used in feature dimensionality reduction purposes.

Table 2. 4 Data Driven Models

With the advancement in the ML algorithms, a bit more sophisticated dense supervised learning approach is applied for traffic predictions by using back propagation techniques in artificially connected neural networks (ANN) [25][51]. Although ANN out performs conventional linear parametric models but struggles with simple time series data learning and finding global minimum. Recently, deep recurrent neural networks (RNN) have shown some great promises for dynamic sequential modelling especially in the field of speech recognition [81][82]. Simple RNNs however suffer from gradient explosion for extra-long sequence training which results in information loss and reduced performance [83]. Fu R et al [42], have used the RNN variants called long short term memory (LSTM) [84] and gated recurrent units (GRU) for the traffic forecasting because of their ability to retain and pass on the information that is necessary and forget what is redundant using the output and forget gates.

2.6.2 Hybrid Models

In general, parametric models are sensitive to parameter tuning and generally exhibit less prediction accuracy. Hybrid models on the other hand combines the power of both parametric and non-parametric models and inherit properties of both for a better performing accuracy model. Deep sophisticated models can also be classified as deep learning models where each different model exploit some different feature or the problem in the data.

Models	Limitations
Extremely randomised Trees (ET), Spatial-Temporal Random Field (STRF), Spatial-Temporal Hidden Markov Model (STHMM) for parameter learning	The assumed additive (and uncorrelated) structure of the segmented model is less accurate for high loads of correlated data. Where previous data resembles the current one more for example (evening and seasonal easily predictable trends).
Least Square Support Vector Machine (LS-SVM), Temporal Window based Support Vector Machine (v-SVR), Relevance Vector Machine Regression (RVM)	LS-SVM fails to predict the extreme high flow rates future predictions and the possible reason causing such failures is the sudden increase in traffic congestion and un-anticipated trends. RVM sometimes needs to have the upper and lower boundary values estimated beforehand for the variable to be predicted.
Fuzzy Logic Controlled Deep Neural Network (FCDN),	FCDN uses the fuzzy login in back propagation weight training and the fuzzy rules needs to be defined beforehand that makes it a semi-unsupervised learning and difficult to implement in traffic problems.
Denoising Stacked Auto encoders, Stacked Auto encoder (SAE)	Over fits the input data but lacks the inherent generalisation ability of time series-based data. The learning of SAE can be enhances by further recurrent layers.
AdaBoost	Deals with an ensemble built iteratively by reweighting the learning samples based on how well their target variable is predicted by the current ensemble. The worse the prediction is, the higher the weight becomes. So, the overall performance accuracy is dependent upon prediction algorithm

	itself. The working principle of AdaBoost resembles to that of reinforcement learning (RL)
Gap-Sensitive Windowed KNN (GSW-KNN), KNN-PCA, KNN-RFE, Random Forests-PCA, Random Forests-RFE	Combining the power of two ML algorithms and their short comings as well. GSW-KNN seems a promising approach for missing data imputation for time series traffic parametric data.
Convolutional Neural Network (CNN) – Recurrent Neural Network (RNN) (DeepTransport) Convolutional LSTM NN (ConvLSTM NN)	The deep learning models that combines the spatial feature exploiting power of CNN and temporal using LSTM. Generally Good for dealing structural missing data. Can Learns adjacent roads spatiotemporal features based on the impact on a road section and its local network. Critical road sections can be ranked according to the distance and thus can be distinguished based on their order. Good for mining learning road topology.

Table 2. 5 Summary of Hybrid Models.

2.7 Established Theoretical Relevance for the Proposed Methodology from Literature Review

From literature review the closely related mathematical explanation to the later proposed methodology is reported here. In [85] road link traffic travel time is estimated using the exit and entry information for the network. According to [85], road link travel times are inferenced based on the likelihood principle. This technique of sampling a traffic network for individual road link travel times is close to being similar like the one proposed in this research. In contrast, this research applies unsupervised and supervised machine learning techniques considering the road intersection and junction points. A detailed explanation of the proposed methodology is discussed in chapter 4. Let A represents the set of road links and n being the total number of links. I being the overall number of observations with x representing the observed travel time of trip i. The set of observations is denoted by D . i.e. $D = \{ x_1 , x_2 , \dots x_I \}$. Assuming all trips have known paths, $\delta_{i,a}$ is the incidence indicator which is equal to 1 when link A is on trip i and zero otherwise. Considering the assumption that all link travel times considered on the network are independent and normally distributed random variables denoted by $N (\mu_a , \sigma_a)$ for each individual link a.

$$Maximise \quad \mathcal{LL} (\eta , \tau | D) = \sum_i \log\left(\frac{1}{\sqrt{2\pi}\,\tau_i} e^{-\frac{(x_i - \eta_i)^2}{2\tau_i^2}} \right) \tag{2.1}$$

Where η_i & τ_i respectively are the mean travel time and the standard deviation of trip i. Equation 2.1 serves as the bare minimum equation to serve the concept of finding traffic flow predictions on a network level.

2.8 Summary

In this chapter the series data predictions techniques for the traffic variables in literature are explored. The focus was on traffic flow modelling techniques using machine learning, but the latest methods of various statistical flow forecasting are also explored. The traffic prediction model and techniques are adaptable hence some areas of these closely related fields are also explored. The closely related field are passenger wait times forecasting at bus stops, bus headway stops times and congestion predictions. In the next chapter the literature extracted techniques are compared for their advantages and disadvantages in terms of their adaptability towards the proposed methodology.

This chapter reviews the state-of-the-art traffic prediction techniques. Closely related models are grouped into three categories based on how well they are constructed and how well they treat the input traffic time series data. Finally, their usability and performance limitations are compared.

3. Models and Architectures

In this chapter the specific chosen models are discussed in detail. The models are then implemented in the experiments. The reasons for choosing these specific models are explained in section 3.1 whereas section 3.2 lists the complete implementation details with the frameworks used along with each individual model pipelines for searching best performing model parameters.

3.1 Selected Models Theory

In this section the implemented models are explained. As researchers suggest non-parametric models are better suited for problem learning part when compared to parametric models as they happen to be generally better in generalising complex data and have the better ability adapt to its patterns, like forecasting traffic data. As parametric tests and methods assume underlying statistical distributions in the data. Parametric approaches are usually the first choice as the input and output traffic variables are noisy and the relationship between each other is nonlinear and poorly understood. Pattern recognition-based approaches, a subset of the non-parametric approaches, seem to be more appropriate as they are effective in identifying similar traffic conditions needed to generate a prediction. The ten data driven models with a mix of parametric and non-parametric approaches are Historical Average (HA), Seasonal Autoregressive Integrated Moving Average (SARIMA), Random Forest Regressor (RFR), Support Vector Regressor (SVR), Feed Forward Backpropagation Neural Network (FFBNN), more complex Deep Belief Network (DBN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), along with complex hybrids of Backpropagation Long Short Term Memory with Neural Network (B-LSTM-ANN) and Deep Convolutional Neural Network with Long Short Term Memory (DCNN-LSTM). The reasons for the chosen models and a detailed mathematical explanation is presented in the relevant sections.

3.1.1 Historical Moving Average (HA)

Historical Moving average is a naïve and effective approach in times series forecasting. This model is used as very basic performing models and is considered as a baseline in the set of experiments. Considering HA as baseline performing model the error difference from HA gives a general idea of the inherent temporal variations in the data. For this reason, the moving average is always performed using a windowing operation. Window based moving average often called trailing moving average uses the past and future observations to be considered for the average and is slid along the whole time series. Although a univariate model but predications can be made for the next day as the new observations are made available. Window sizes of one, two, and three are used in this experiment for short, medium and long interval forecasting respectively.

For the univariate training data ($X_T = x_1, x_2, \ldots, x_t$), with windows size greater than zero (k >0), the k^{th} moving average or HA at t is given as:

$$HA(t) = y_{t-k+1} = mean\,(\,x_{t-k+1} + x_{t-k+2} + \cdots + x_t\,) \tag{3.1}$$

3.1.2 Seasonal Autoregressive Integrated Moving Average Model (SARIMA)

The ARIMA model is the most popular statistical model and is a good fit for baseline model for comparison with other machine learning models. The ARIMA model is composed of three different simple models: the order of auto-regressive model (AR), the degree of differencing and the order of the moving average (MA) model. Where p, d and q are used to represent each feature in the model so the overall model is often written as $\mathrm{ARIMA}\,(\mathrm{p}, \mathrm{d}, \mathrm{q})$. The integrated term d corresponds to the non-stationary univariate series data. The term d determines the lag for the response before the actual computation for the difference. It's helpful with seasonal non-stationary data. Ideally Seasonal-

ARIMA expects the time series input to be seasonal stationarity so the time series is put for the stationarity test for null hypothesis as stationary using Augmented Dicky Fuller (ADF) test [86, p. 9] is performed in the experiments section. $SARIMA\ (\mathrm{p, d, q, P, D, Q, m})$ as the name suggest deals with the seasonal trendy stationary data, contains three more parameters, which are P, D,Q and m. P is the order of the seasonal component for AR model, D is the order of the integration seasonal model, Q is the order of seasonal component belonging to MA model and m represents the cyclic seasonal period or time steps for the seasonal lag consideration. According to [87, p. 208], SARIMA model applied to the time series y_t is given by the following expression:

$$SARIMA\ (p,d,q)\ x\ (P,D,Q)_m = (\ \Phi\ (L^m)\phi\ (L)\ \Delta^d\ \Delta_m^D\ \ y_t = \ \theta_0 + \ \Theta(L^m)\theta(L)\epsilon_t\ ) \tag{3.2}$$

Where m is the seasonal length, L is the lag operator and ϵ_t is the gaussian white noise process with zero mean and variance. Δ^d and Δ_m^D are the difference and seasonal difference operators with d and D representing their orders respectively. The difference operations help transform the non-stationary time series into a stationary time series. The AR part is this model is derived by multiplying autoregressive lag polynomials $\phi\ (L)$ and $\Phi\ (L^m)$. And the MA part if represented by the moving average lag polynomials $\theta(L)$ and $\Theta(L^m)$.

3.1.3 Random Forrest Regressor (RFR)

The data driven Random Forrest Regressor (RFR) is selected as on the comparison models from the practical point of view as it's quite fast to estimate the parameters as they are very few, fits quite well and the fit model is quite intuitive which allows visualisation of the class ranking of variables in terms of their importance compared to other machine learning models. RFR is the set of regression-based trees and the result is the average of those trees. Each tree has n number of nodes from a selection of m variables in the data. The node leaves are constructed with each independent variable value such that the average of the dependent variable on either side of that value minimises the sum of squared with the actual value of the data point. When a prediction is made using RFR it simply predicts the average from all the predefined number of trees and they can be traced back to each individual leaf depth formed from each individual tree during the training process. RFR training and prediction algorithm routines are listed below as pseudocode 1 & 2 routines respectively.

Pseudocode 1: RFR Training

Data: Training samples ***X***, features ***m***
Result: Trained model
1 Random selection of ***k*** features from ***m,*** given that ***k*<<*m***;
2 From ***k*** features, calculate the node **d** with best split point;
3 Node split into daughter nodes using further best splits;
4 Repeat 1 to 3 steps, until given ***max number leaf split*** equals nodes;
5 Repeat 1 to 4 steps for ***n*=max *number of trees split*** to create ***n*** number of nodes;

Pseudocode 2: RFR Prediction

Data: Test samples ***Y***, features ***p***
Result: Predicted Value
1 Select ***p*** features from ***Y*** each sample in an iterative manner and use the rule of created trees in trained model to predict for store the predicted outcomes of each individual tree;
2 Calculate the **votes or close to similar votes** for each predicted outcome;
3 Consider the **most voted** predicted outcome or the average of closely related outcomes as the **outcome** of the trained model;

3.1.4 Support Vector Regressor (SVR)

Support Vector Regressor (SVR) is chosen to deal with the nonlinear data prediction with its capability to fit regression functions to the set of data points. SVR model is a non-parametric model and can be applied without any prior data knowledge with ease. Support Vector Machines (SVM) theory, the predecessor of SVR with its methods for estimating the indicators of a real valued function were first discussed in [88, p. 443]. The SVR is a classical statistical theory-based learning models that works implements the structural risk minimisation principle from computational learning theory. It works like a pattern recognition; the basic aim is to map input data vector x into a high dimensional feature space F using a non-linear mapping function Φ. The linear regressions are carried out in higher space F which corresponds to the non-linear regression in the low dimensional input space. The corresponding regression function is given as:

$$f\,(x) = (w\,.\Phi(x)\,) + b \quad with\,(\,\Phi:\,R^n \to F, w\,\epsilon\,F\,) \tag{3.3}$$

w represents the vector in the feature space, $\Phi(\mathrm{x})$ is the mapping function to map input x and b is the threshold. The mapping function is usually a kernel function. Four different kernels linear, poly with third order degree, sigmoid and radial basis function (RBF) considered. Replacing the dot product with kernel function enables the higher dimensional feature space mapping easy. RBF is usually among the most popular choice for nonlinear mapping, chosen because of its robustness short state predictions [66] and is defines as:

$$k\,(x, y) = exp(-\gamma\,|\,x - y|^2\,) \tag{3.4}$$

γ in equation 3.4 represents the gaussian bandwidth. With the aim to find the optimal weight w and bias b . To consider the regression problem the flatness of weights and the error generated though empirical risk estimation process are considered [88, p. 473]. The w value is optimised by minimising the sum of empirical risk $R_{emp}(f)$ and the complexity term $|w|^2$. The regression function is given as:

$$R_{ref}(\,f) = \,R_{emp}\,(f) + \frac{\lambda}{2}\,\,|w|^2 = C\,\sum_{i=1}^{N}\Gamma\,(\,f(x_i) - \,y_i\,) + \,\lambda\,|\,w\,|^2 \tag{3.5}$$

Where C is a curve fitting number usually defined beforehand, N is the sample size and Γ is the loss function with λ being the regularisation constant. Different loss functions are considered which when input into equation 3.5, it can be reduced to a quadratic statistical problem defines as:

$$minmise(\,\frac{1}{2}\,\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*\,)\big(\alpha_j - \alpha_j^*\big)\,\,k\big(x_i\,, x_j\big) - \,\sum_{i=1}^{N}\alpha_i^*\,(y_i - \varepsilon) - \,\alpha_i(y_i + \,\varepsilon)\,) \tag{3.6}$$

Given that $\quad \sum_{i=1}^{N}\alpha_i - \alpha_i^* = 0 \quad \alpha_i\,, \alpha_i^*\,\epsilon\,[0, C]$

α_i and α_i^* are Lagrange multipliers and are found through the constraints of equation 3.7.

$$w = \,\sum_{i=1}^{n}(\alpha_i - \alpha_i^*\,)\big(\alpha_j - \alpha_j^*\big)\,\,\Phi(x_i) \tag{3.7}$$

Equation 3.7 represents the weight term in terms of the data which when input back into the original equation 3.3 gives it the form given as:

$$f(x) = \,\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\,\,k(x_i, x) \tag{3.8}$$

3.1.5 Feed Forward Backpropagation Neural Network (FFBNN)

Neural Network (NN) is another popular non-parametric model which in the basis of Artificial Intelligence science. The basic idea of NN is to mimic the human brain and its decision-making power. Feed Forward Backpropagation Neural Network (FFBNN) is one of the forms of NN other being the recursive NNs. FFBNN can harvest useful nonlinear mappings and insights from input data features and approximate them with close to real value. Neurons are the basic building blocks of the FFBNN. Each neuron receives an input signal, processes it through an activation function to generate an output according to the weight value associated with the neuron. The neuron weights are then calibrated in the feedback training process using the weight error difference from network output hence the name back propagation is given to the model. These neurons are arranged in the networks of layers making a feed forward network. The simplest of the neurons in the perceptron. With a set of inputs $(x_1, x_2, \dots, x_n)$ with the network output y the weighted sum of perceptron inputs $w_{ij}x_j$ with a threshold bias value ϑ_i , subject to an activation function $\Phi(.)$ is given as:

$$y_i = \Phi\left(\sum_{j=1}^{N} w_{ij}x_j - \vartheta_i\right) \tag{3.9}$$

A detailed discussion of an MLP and its graphical illustration is also given in section 1.1. The group of different perceptron form different layers of the network with the inner to the visible input and output layers are termed as hidden layer perceptron. A network with multiple perceptron is often called multi-layer perceptron (MLP). At each hidden layer of the network the data features are computed. There are many different types of activation functions. The set of activation functions considered in this thesis are given in table 3.1. The FFBNN training involves two phases: forward and backward passes.

	Activation Function (Φ)	**Mathematical Implementation**
1.	sigmoid	$\Phi(x) = \frac{1}{1+e^{-x}};\ \Phi(x) \in [0,1]$
2.	softmax	$\Phi(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}};\ j = 1,2,\dots,K;\ \Phi(x)_j \in [0,1]$
3.	tanh	$\Phi(x) = \frac{1-e^{-2x}}{1+e^{-2x}};\ \Phi(x) \in [-1,+1]$
4.	relu	$\Phi(x) = \max(0,x)\,;\ \Phi(x) \in [0,\infty)$
5.	linear	$\Phi(x) = \mathrm{x}$

Nomenclature: softmax represents the normalised exponential function for multiclass logistic function flow values in our case, that makes K-dimensional vector x to have values in range [0, 1] that all add up to 1.

Table 3. 1 Layer Activation Functions.

1) Forward Pass

This is the first step in the feed forward network where training data is passed through the network and the error estimate Δf is calculated based on the loss or cost function and the final network output.

2) Backward Pass

Given the calculated error estimate in forward pass. The weights of the network in the backward pass are altered in an iterative manner. Different network weight update or optimisation techniques have been used. Table 3.2 gives the optimisation techniques considered in the scope of this thesis. In table 3.1, SGD is the most common technique mostly used which involves the calculation of second order gradient descent which is then adjusted back into the weight's matrix. Adaptive SGD or Adagrad on

the other hand inherits the gradient descent with adaptive qualities. Whereas in RMSprop the root mean square of the second order gradient descent is taken and then adjusted back into the weights.

	Optimisation Function (f)	Mathematical Representation
1.	Stochastic Gradient Descent (SGD)	$w_{t+1} = w - \eta \left[\sum_{i=1}^{N} \frac{\nabla Q(w_i)_t}{\mathrm{N}}\right] + \alpha \Delta w;$
2.	Adaptive Gradient Algorithm (Adagrad)	$w_{t+1} = w_t - \frac{\eta}{\sqrt{G_t + \varepsilon}} \odot g_t$
3.	Root Mean Squared Propagated Gradient Descent (RMSprop)	$w_{t+1} = w_t - \frac{\eta}{\sqrt{E(G_t) + \epsilon}} \odot g_t$

Nomenclature: $\boldsymbol{w_i} = (\bar{\boldsymbol{y}}_{\boldsymbol{i}} - \boldsymbol{y}_{\boldsymbol{i}})^{\boldsymbol{2}}$ *,* η *is the learning rate,* α *is the learning momentum factor,* g_t *is the iteration gradient,* $G_t = \sum_{i=1}^{N} g^2{}_{t,i}$ *is the diagonal.*

Table 3. 2 Training Optimiser functions.

FFBNN Convergence

After several successful forward and backward passes the FFBNN starts to converge to find the local minimum in the error curve. The single variable best performing optimizer may differ from the multi variable bets performing optimizer this is because of the most suitable to make the network converge for learning the data. For this reason, each optimizer uses the learning rate η which is definable and can make a difference while the optimizer goal is to converge. A large η can sometimes make the network step over the local error minimum towards the direction opposite to direction of convergence whereas the least η can make the network to take it much longer to converge. The amount of training batch considered for training for training can also make a difference. Different optimisers performed differently with different batch size. Where epochs are the number of iterations carried out to go through each data sample feature once.

3.1.6 Deep Belief Network (DBN)

The type of deep neural network (DNN) with at least some hidden layers and a significant number of hidden units in each layer. The simplest DBN is made up of stacks of Restricted Boltzmann Machine (RBM) models with a neural layer at the top as the output layer. A typical DBN is trained using a layer by layer greedy algorithm for the supervised data. RBM [37] is an energy based model. Each RBM unit has two layers, visible layer v and the hidden layer h, both of which are connected by untrained weights. In the stack of RBM the hidden layer of previous RBM is the visible layer of the next RBM. The RBM parameter set $\theta = (w, b, a)$ where w_{ij} represents weights between layer v_i and h_j and the biases associated with each of the layers are $b_i$and a_j respectively, as shown in figure 3.1. The RBM energy function is given as:

$$E\ (v, h\ |\theta) = -\sum_i b_i v_i - \sum_j a_j h_j - \sum_i \sum_j w_{ij}\ v_i\ h_j \tag{3.10}$$

From this the joint probability distribution between the hidden and visible layer is given as:

$$p\ (v, h\ |\theta) = \frac{\exp(-E\ (v,h\ |\theta\ ))}{\sum_{v,h} \exp(\ -E\ (v,h\ |\theta\ ))} \tag{3.11}$$

And the marginal probability distribution of layer v is given as:

$$p\ (v|\theta) = \frac{\sum_v \exp(\ -E\ (v,h\ |\theta\ ))}{\sum_{v,h} \exp(\ -E\ (v,h\ |\theta\ ))} \tag{3.12}$$

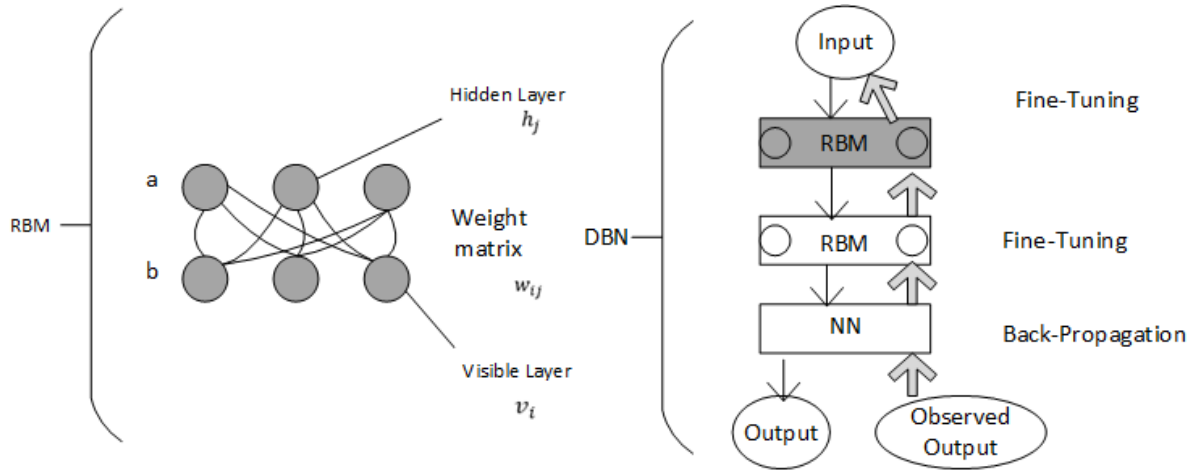


Figure 3. 1 RBM Structure (left) and DBN Model (right).

To obtain the optimal parameters for the set θ for a given data vector v, the derivative approach is used. For this the gradient log-likelihood estimation is calculated as below:

$$\frac{\partial \log p(v|\theta)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$$
$$\frac{\partial \log p(v|\theta)}{\partial a_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model} \quad (3.13)$$
$$\frac{\partial \log p(v|\theta)}{\partial b_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model}$$

Where $\langle . \rangle$ is expectation of the distributions. There are no corresponding connections between the RBM layer units themselves. So, the layer distributions are easily estimated though conditional probability distributions, given as:

$$p(h_j|v,\theta) = \frac{1}{1+\exp(-\sum_i w_{ij} v_i - a_j)}$$
$$p(v_i|h,\theta) = \frac{1}{1+\exp(-\sum_j w_{ij} h_j - b_i)} \quad (3.14)$$

The weights in the RBM components are fine-tuned by the contrastive divergence (CD) [89, p. 3] by default though fast and greedy unsupervised method, and with one additional layer of neuron at the end overall model weights are trained with the backpropagation using supervised learning approach. An activation function is also used in the last output layer.

3.1.7 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) belongs to the deep learning family and it helps exploit the spatial structure of data (e.g. images) to learn the features of the data for the final model to learn something useful. CNN helps to learn the local features in the data. The CNN model is the deep model extension of FFBNN. With having more hidden layers and additionally the convolutional layers at the input. A typical CNN has three components, convolutional layers, pooling layers and a fully connected layer. The CNN inherits all the features of an FFBNN model except that convolutional layers are applied at the start of the normal ANN model and pooling layers are applied in between the ANN layers. Convolution is an operation between two functions, continuous or discrete which in practice has the

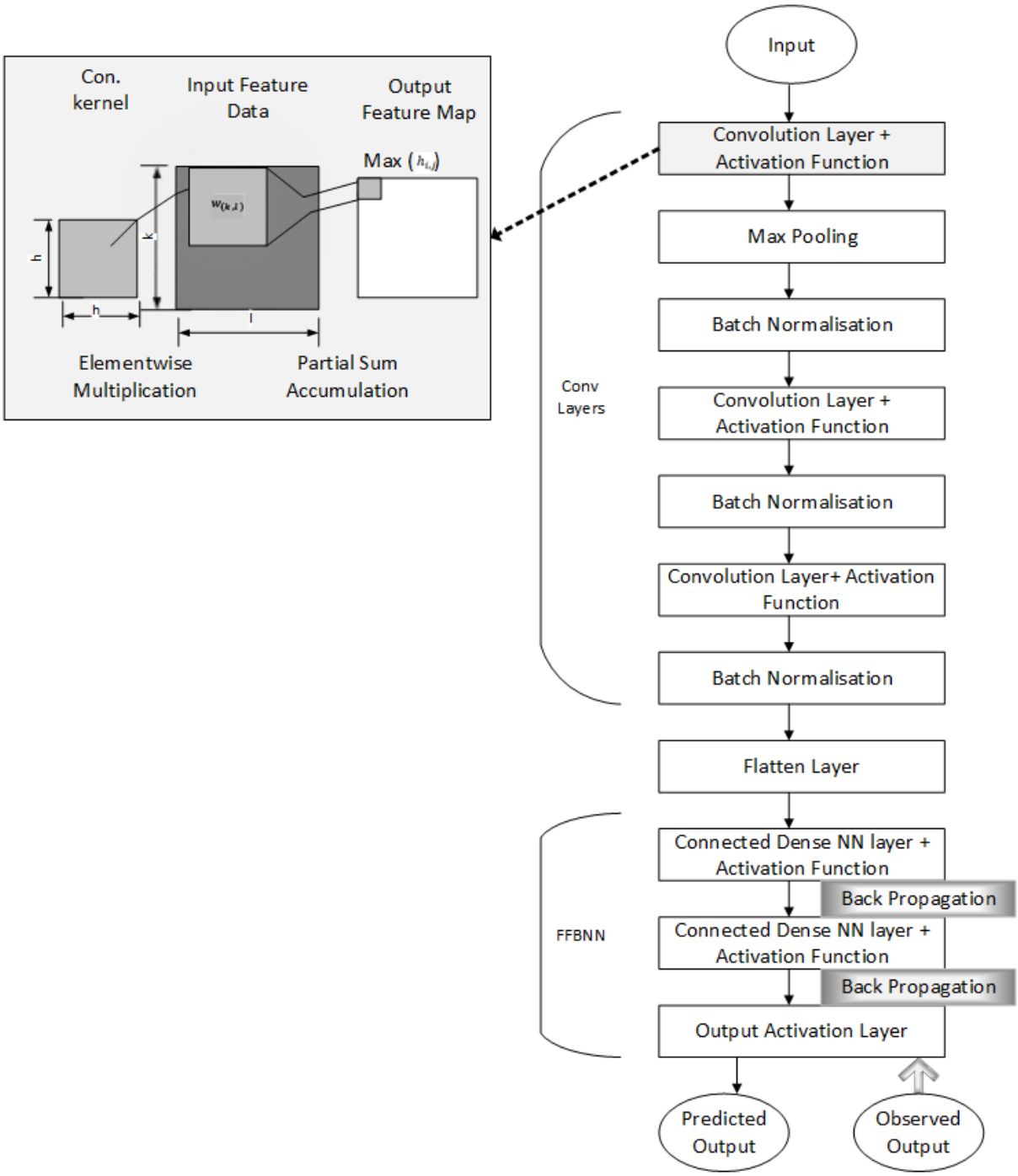


Figure 3. 2 Steps in Convolution Operation (left) and CNN (C-FBNN) Model (right).

effect of filtering one of them by another. The general convolutional operator when applied two discrete functions f and g can be given as the summation of the following form:

$$(f * g)\,[n] = \sum_{m=-\infty}^{+\infty} f\,[m]\,g[n-m] = \sum_{m=-\infty}^{+\infty} f\,[n-m]\,g[m] \qquad (3.15)$$

The input for CNN is an image representing one state, the pixel values of which represents the scaled input values. The convolutional layers act as a filter which when built into the model does emphasize on certain characteristics of the input feature vectors. It behaves like an automatic custom detector of feature patterns to create a feature map. For an input feature matrix, the commonly applied

functions are of two to five elements per dimension and are declared zero on the remaining elements, when strides onto the input feature matrices. The resulting small matrices which represent the groups of filter functions are called kernels. At a given position of the convolutional kernel, the element-wise multiplication of each kernel cell value and the corresponding feature value that overlaps the kernel cell to take the sum of that. For the kernel stride m of width and height h, convolutional output x, input w, the kernel outputs or sub matrices when slid on the input are given as:

$$h_{i,j} = \sum_{k=1}^{m} \sum_{l=1}^{m} w_{(k,l)}\, x_{(i+k-1,j+l-1)} \tag{3.16}$$

Pooling layers after the convolutional operation makes the CNN output as translational invariant. Two pooling mechanism are commonly used as average and max pooling. Average pooling or batch normalisation is used in the CNN model in this thesis. The max pooling operation lists the maximum values as outputs from all the values input by the kernel operations if they fall into the kernel range compensating the number strides used for sliding the kernel. This is mathematically given as below:

$$h_{i,j} = \max\{ x_{(i+k-1,j+l-1)}\ \forall\ 1 \le k \le m \text{ and } 1 \le l \le m \} \tag{3.17}$$

$$y_{i,j} = f_a \sum(h_{i,j} + b_{i,j}) \tag{3.18}$$

The CNN model final output $y_{i,l}$ is given by equation 3.18 where f_a is the output layer activation function and $b_{i,j}$ is the bias term. As shown in figure 3.2, the generated feature map from convolutional layers in then passed through simple FFBN network that gets activated for certain pattern or feature values as actually present tin the input. For the training, each convolutional kernel layer parameters are optimised for the involved parameters to reflect the useful features to the FFBNN. FFFBNN is trained using the BP algorithm with the suitable optimiser as already discussed in section 3.1.5.

3.1.8 Long Short-Term Memory (LSTM)

Long time series have long data dependencies. To learn these long-term dependencies the conventional neural based network is not enough to do the job. For this a recurrent neural network (RNN) based Long Short-Term Memory (LSTM) are used. An RNN is similar in structure to a feed forward neural network except that the output of each neuron unit is fed back to its input which makes it a recurrent value learner. LSTM is a more advanced form of RNN. LSTM were first introduced almost two decades ago [90] for language processing where it proved to be useful in exhibiting better performance in memorising the long term dependencies in the data. An LSTM is a memory block structure controlled by memory cells through their respective input, output forget gates and peepholes connections. Data flow and operations in Long Short-Term Memory (LSTM) unit structure which contains the forget, input, output, and update gate are given in figure in appendix B.

$$i_t = \sigma\,(x^t W_{xi} + h_{t-1} W_{hi} + c_{t-1} W_{ci} + b_i) \tag{3.19}$$

$$f_t = \sigma\,(x^t W_{xf} + h_{t-1} W_{hf} + c_{t-1} W_{cf} + b_f) \tag{3.20}$$

$$o_t = \sigma\,(x^t W_{xo} + h_{t-1} W_{ho} + c_t W_{co} + b_o) \tag{3.21}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(x^t W_{xc} + h_{t-1} W_{hc} + b_c) \tag{3.22}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{3.22}$$

$$y_t = W_{yh}\, h_t + b_y \tag{3.23}$$

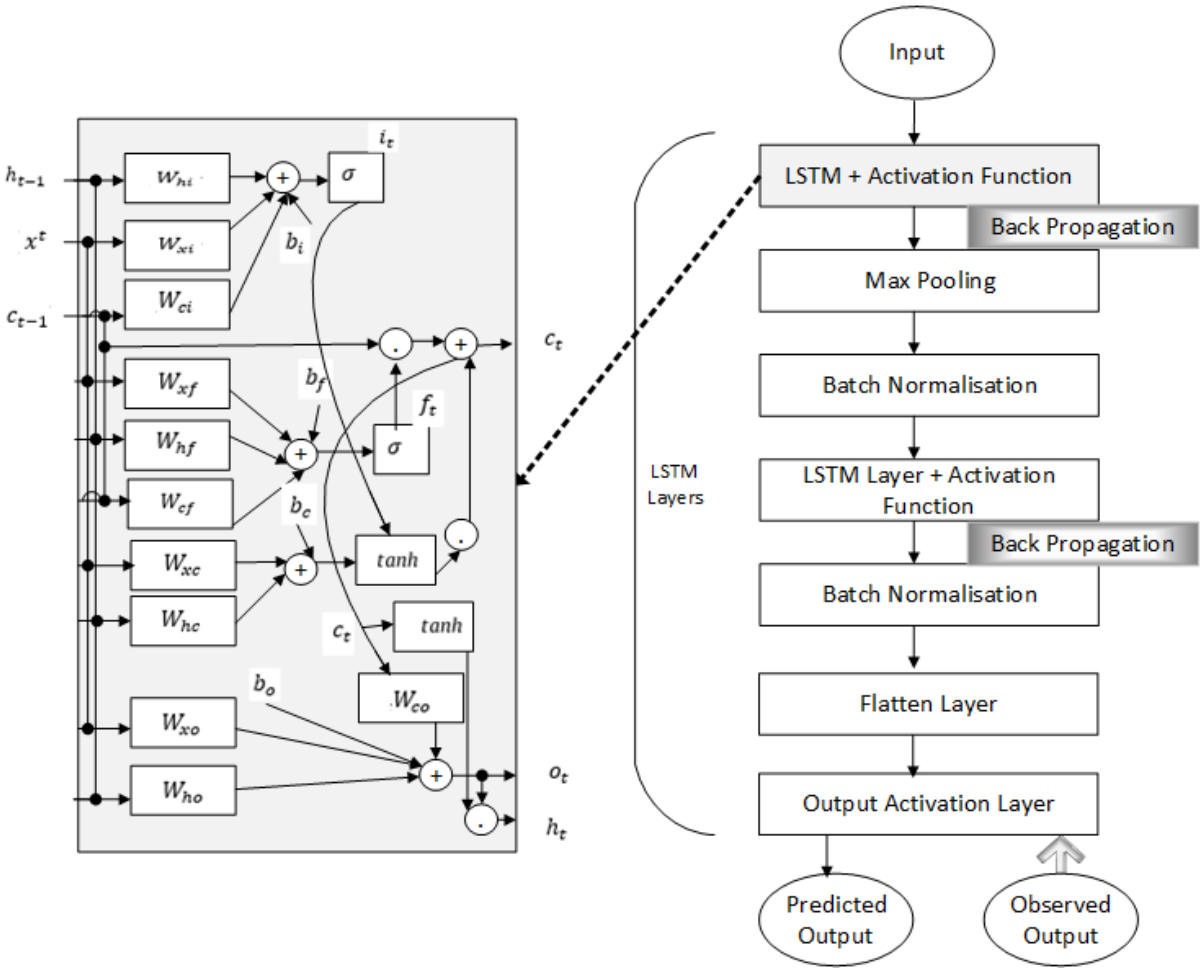


Figure 3. 3 LSTM Memory Unit Structure (left) and Stacked LSTM Model (right).

x^t is the feature input to the memory unit whereas i_t, f_t, o_t, h_t, c_t represents the output of the input gate, output of the forget gate, output of the output gate, the final cell state output and the final memory unit output, respectively. W_{xi}, W_{xf}, W_{xo} represents the weights between the input layer and input gate, input layer and forget gate and input layer and output gate, respectively. Similarly, W_{hi}, W_{hf}, W_{ho} are the weights assigned between the recurrent hidden layer and the input layer, forget gate and the output gate, respectively. Likewise, as the subscript suggests, W_{ci}, W_{cf}, W_{co} the weights associated with the cell state and the input gate, forget gate and the output gate, respectively. All the variables represented by b are the associated with each of the gates as given in equations (3.19 – 3.22). σ represents the sigmoid activation function used. The hidden recurrent unit output h_t is passed from the previous LSTM memory unit to the next LSTM unit and from final LSTM memory unit of one layer to the next layer memory unit as an input. THE LSTM model layers are trained using backpropagation for different optimisers and layer parameters and the best performing parameters are chosen as the final model forecasts. The model structure used for calibrating the parameters is shown in figure 3.3. Like CNN model one max pooling layer is inducted. And every LSTM layer is succeeded by a batch normalisation layer that normalises the batch vectors during each training iteration. Equation 3.23 represents the overall model output when iteratively calculated by following the equations from (3.19 – 3.23).

3.1.9 Backpropagation Long Short-Term Memory - Neural Network (B-LSTM-ANN)

Another non-parametric, data driven, and a bit deeper B-LSTM-ANN learning model is considered. After training the LSTMs using backpropagation for time dependent learning, B-LSTM-ANN utilises the

combined power of LSTM for time dependent recurrent sequence learning and the usual feed forward network (ANN) for state space learning. B-LSTM-ANN exhibits a much better performing model than just the LSTM model for the time series predictions [52]. The B-LSTM-ANN model can learn using the optimal time lags when combined with the recurrent memory for the pattern determination. LSTMs are good in recurrent value adaptive learning using gating to control information flow, on the other hand ANN also help memorising for the overall pattern attributes in a series, when passed though ANN after LSTM layers. Other reasons to test B-LSTM-ANN model is that simple RNN and typical LSTM does suffer from a little gradient explosion problem when trained for long data series which make the model a little unstable and unreliable, but the combination of LSTM and ANN makes it a more reliable model by keeping the network error constant. Also, there are very few instances where B-LSTM-NN is applied to the transportation problems. Figure 3.4 shows the stacked LSTM layers with the subsequent ANN layers attached into become the B-LSTM-ANN model.

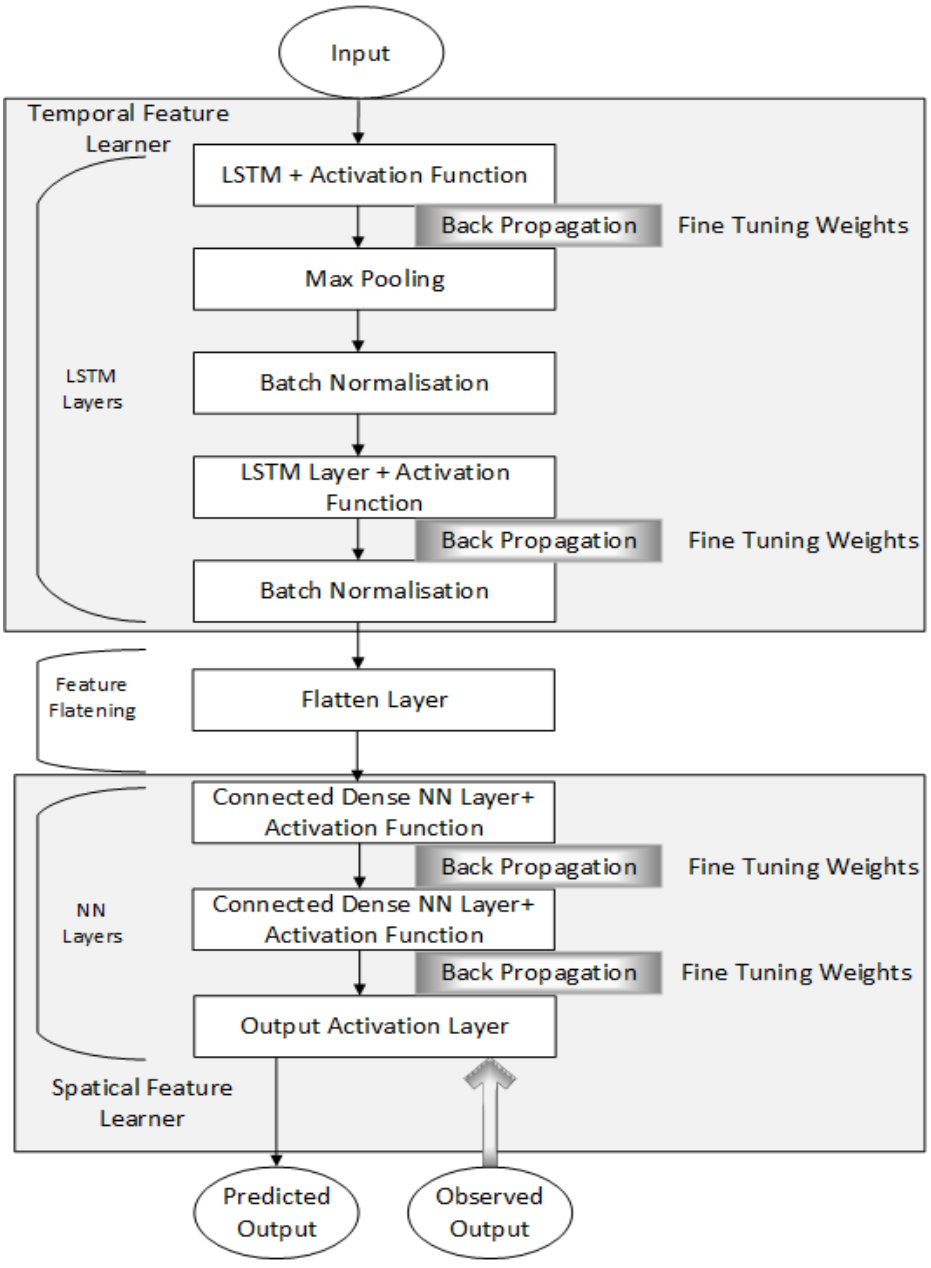


Figure 3. 4 Stacked LSTM Layer Combined with NN layers (B-LSTM-ANN).

LSTM's purpose can be defined as the estimation of the conditional probability $p(y_1, y_2, \dots, y_{T'} \mid x_1, x_2, \dots, x_T)$ given that $(x_1, x_2, \dots, x_T)$ is an input sequence and $(y_1, y_2, \dots, y_{T'})$ is the corresponding output sequence. The lengths of T' and T may differ. The deep LSTM computes the conditional probability by first computing the fixed dimensional input representations v, of the input sequence, from the last hidden memory state of the LSTM layer [81, p. 3]. The hidden state h_t for each individual LSTM unit is calculated as given by the equation 3.22. Accordingly, the standard LSTM network for the i^{th} node with internal hidden states v of corresponding inputs $(x_1, x_2, \dots, x_T)$ is given by equation 3.24.

$$y_j = p(y_1, y_2, \dots, y_{T'} \mid x_1, x_2, \dots, x_T) = \sum_{t=1}^{T'} P(y_t \mid v, (y_1, y_2, \dots, y_{t-1})) \quad (3.24)$$

With a set of inputs $(y_1, y_2, \dots, y_{T'})$ for dense ANN from stacked LSTM output, final network output y_0 for the weighted sum of ANN inputs $w_{ij}y_j$ with the corresponding threshold bias value ϑ_i, subject to an activation function $\Phi(.)$ is given as:

$$y_{0i} = \Phi\left(\sum_{j=1}^{N} w_{ij}y_j - \vartheta_i\right) \quad (3.25)$$

B-LSTM-ANN training is done in a truncated Back propagation Though Time (BPTT) manner. It's the same as the normal back propagation except that it involves the gradient descent optimisation across time intervals specifically for the recurrent networks. Error rates tend to decay forever as they are truncated when they pass through the output gate of memory cell, this process makes the error decay to follow the exponential process outside the memory cell. Due to this reason B-LSTM-ANN have the ability for learning arbitrary long dependencies [52, p. 191].

3.1.10 Deep Convolutional Neural Network - Long Short-Term Memory (DCNN-LSTM)

To dive further into deep complex deep learning CNNs are adapted after an initial anticipation of an even more improved performing B-LSTM-ANN model. CNNs have been quite successful in extracting features in the form of a feature maps as shown in figure 3.2 (left). The shallow convolutional layers help mining the near side time dependent data features and deep convolutional layers on the other hand help learn the more generic or distant related features. The fact that the distant features are effectively mined is due to the convolution and pooling layers which make the features co-relate with the near side features thus a complete pattern learning feature map by DCNN is achieved. This phenomenon is shown in figure 3.2 (left), the kernel filter during the convolution operation makes the features in input feature data appear closer than before in the output feature map. The graphical representation of DCNN-LSTM model is given in figure 3.5.

If input to the DCNN is $A^T = \{x_{k,l}\}$, where k and l represent the input data dimensions. k being the number of samples and l the number of features, respectively and the output $X^T = \{x_t\}$, at time t is given as in equations [3.16-3.18].

$$X^T = y_{i,j} = \Phi\left(\sum(\max\{\sum_{k=1}^{m}\sum_{l=1}^{m} w_{(i,j)}\, x_{(i+k-1,\, j+l-1)}\}) + b_{i,j}\right) \quad (3.27)$$

$$\forall\ 1 \le k \le m \text{ and } 1 \le l \le m$$

Φ represents a nonlinear activation function. After convolution max pooling is employed for the more prominent feature selection and to reduce the number of learning parameters for the densely connected NN layers. The output of DCNN is the input of the LSTMs and the output of LSTM is the final output of the model. For the temporal features the cell states and the output states of the gates from the DCNNs output are calculated by following the sequential equations (3.19-3.23). The final predicted outcome of the model is given as:

$$Y^{T+1} = \Phi\left(p\left(Y^{T'} \mid X^{T}\right)\right] = \Phi\left(\sum_{t=1}^{T'} P(x_t \mid v, (x_1, x_2, \ldots, x_{t-1}))\right) \quad (3.28)$$

Where v represents the fixed representations calculated using the hidden LSTM units. Φ represents the output activation function. Like B-STM-ANN training, DCNN-LSTM model is trained end to end using the BP training mechanism. The feature vectors after the DCNN are reshaped to make the input compatible with the first LSTM layer. By default, an LSTM unit requires time steps or interval as one of the input dimensions. So, inducting one-time step with the reshaping feature vector was mandatory.

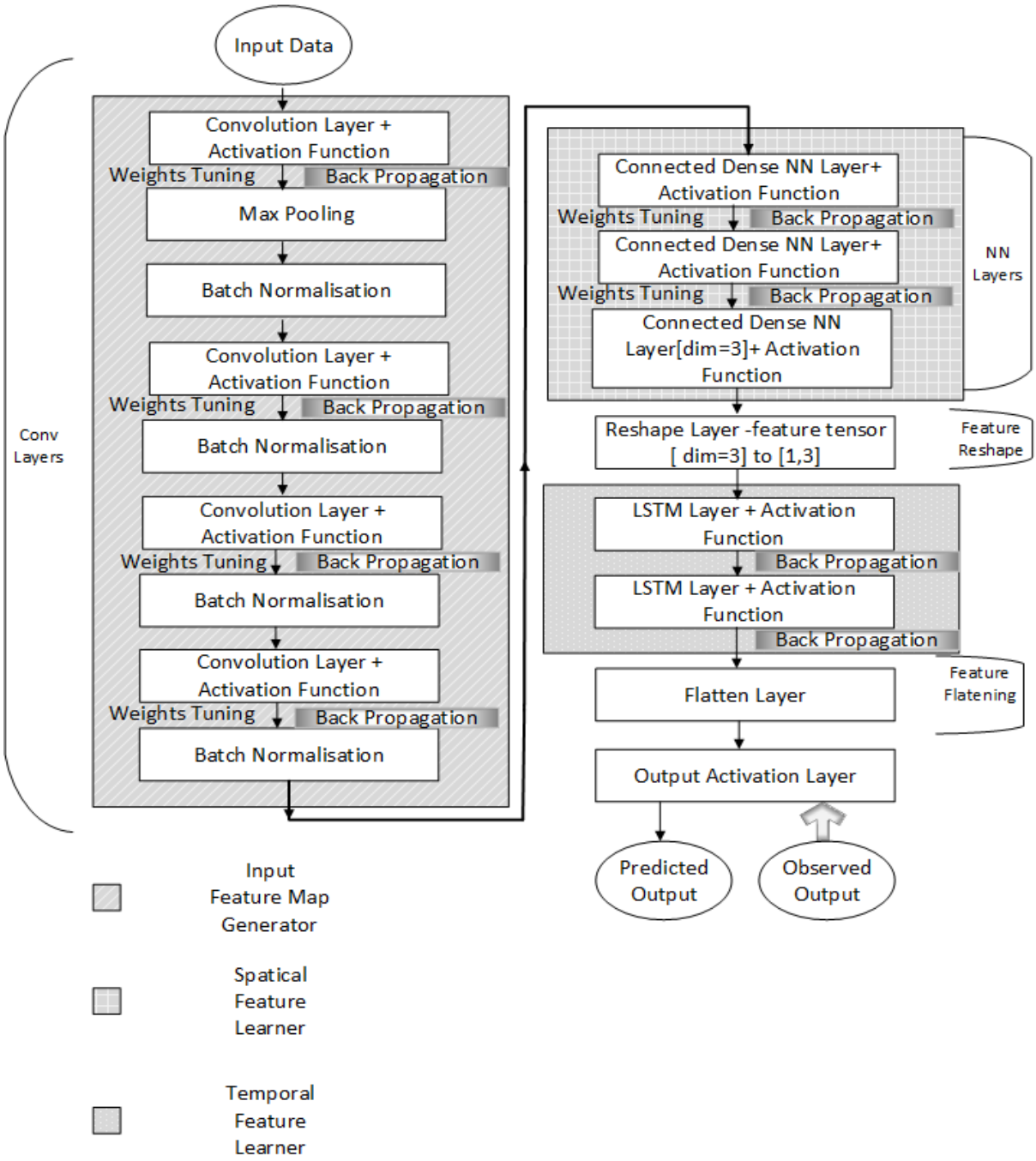


Figure 3. 5 Deep Convolutional Neural Network- Long Short-Term Memory (DCNN-LSTM) Model.

3.2 Hardware and Software Implementation Details

All the development and experiments are carried out using the popular programming and simulation language Python 3.7.1[1]. Anaconda which is an open source distribution compiling the data science and machine learning libraries, dependencies and binaries was used. Being a popular dynamically interpreted language python is fast and suitable for Realtime processing applications. The popularity of python has risen since the existing libraries for scientific computing and heavy processing written in C are easily integrate able with python. Among the other factors is the already growing vast community support of python contributors.

3.2.1 Data Exploration Library

Data exploration and pre-analysis was done using the python pandas library. Pandas deals with the data exploration by converting it to the tabular data structures and frames with columns. Pandas[2] is an effective scientific tool which makes it easier to resample time series data, re indexing the data frame or group by any column headers for better understanding the visualising the data plots directly from data frames.

3.2.1 ML Implementation Library

Already discussed models architectures, in the previous sections, are implemented using the various available open sourced library packages. The main machine learning libraries used are Keras[3] and TensorFlow[4]. Keres is a high-level machine leaning API which is written in python and runs on top of TensorFlow library. Keras is preferred over TensorFlow due to its fast experimentation ability to build the complex network in minutes from scratch which would have taken more time with the TensorFlow library. Keras also supports convolutional and recurrent neural networks as well as the combination of both along with the processing can be accelerated by specifying the workloads on each of the CPU or processor cores. Other features of Keras library includes user friendliness, modularity, easy extensibility and works with python.

TensorFlow the low-level python API used by Keras is developed by Google in 2015. TensorFlow deals the model computation in the form of graphs this makes it possible to developed new architectures based on the basic unit constructs. Some of the TensorFlow features include easy model development, graphs computations to be carried by both the CPU and GPUs which makes it more easier using Keras API on top of it.

To pre-process the data before training and for preparing training and validation datasets scikit-learn[5] python library was used. For end to end model training with best parameters models flow pipelines were written which made it easy to find the best performing parameters for each individual model though grid search function in scikit-learn library. The best performing model parameters are given in appendix A. Interestingly scikit-learn functions are written to be compatible with Keras high-level prediction functions.

---

[1] https://www.anaconda.com/distribution/
[2] https://pandas.pydata.org/
[3] https://keras.io/
[4] https://www.tensorflow.org/
[5] https://scikit-learn.org/

4. Research Methodology and Contributions

4.1 Introduction

In chapter 2 the review of literature review with an overview of different methods of modelling traffic flows, travel times, road congestion analysis, and the prediction models positioned around the latest state of the art machine learning algorithms and in chapter 3 the chosen models are discussed in detail. This chapter covers details of the potential datasets, more favourable datasets breakdown their suitability for the research methodology and the proposed mechanism for the chosen machine learning models.

4.2 Study Area

This section explains the approach and the criteria to gather the traffic flow characteristics data. The initial aim of this research is to analyse the data in the Hertfordshire UK area. Due to the type, availability and format of the data it was decided to consider the UK traffic road networks as a case study for this research. This section presents an explanation of the study area and the related datasets. Due to the comprehensive data availability the considered defined area is shown in figure 4.1 a. The broader application of this research outcome will follow the same procedure for the whole of road networks.

4.3 Data Collection

Before getting into the proposed algorithm it is very important to know the procedures adopted to gather the data. After a lot of search some of the dataset with comparatively reliable sources are shortlisted in table 4.1. A list of potential open datasets that are suitable for the proof of concept implementation are shown in table 4.1. We have only discussed the datasets that have the enough information for our model validation and testing. On site sensor loops and radar technology that were used to log the data at discrete points make the collected data a complete set of suitable datasets for our research methodology.

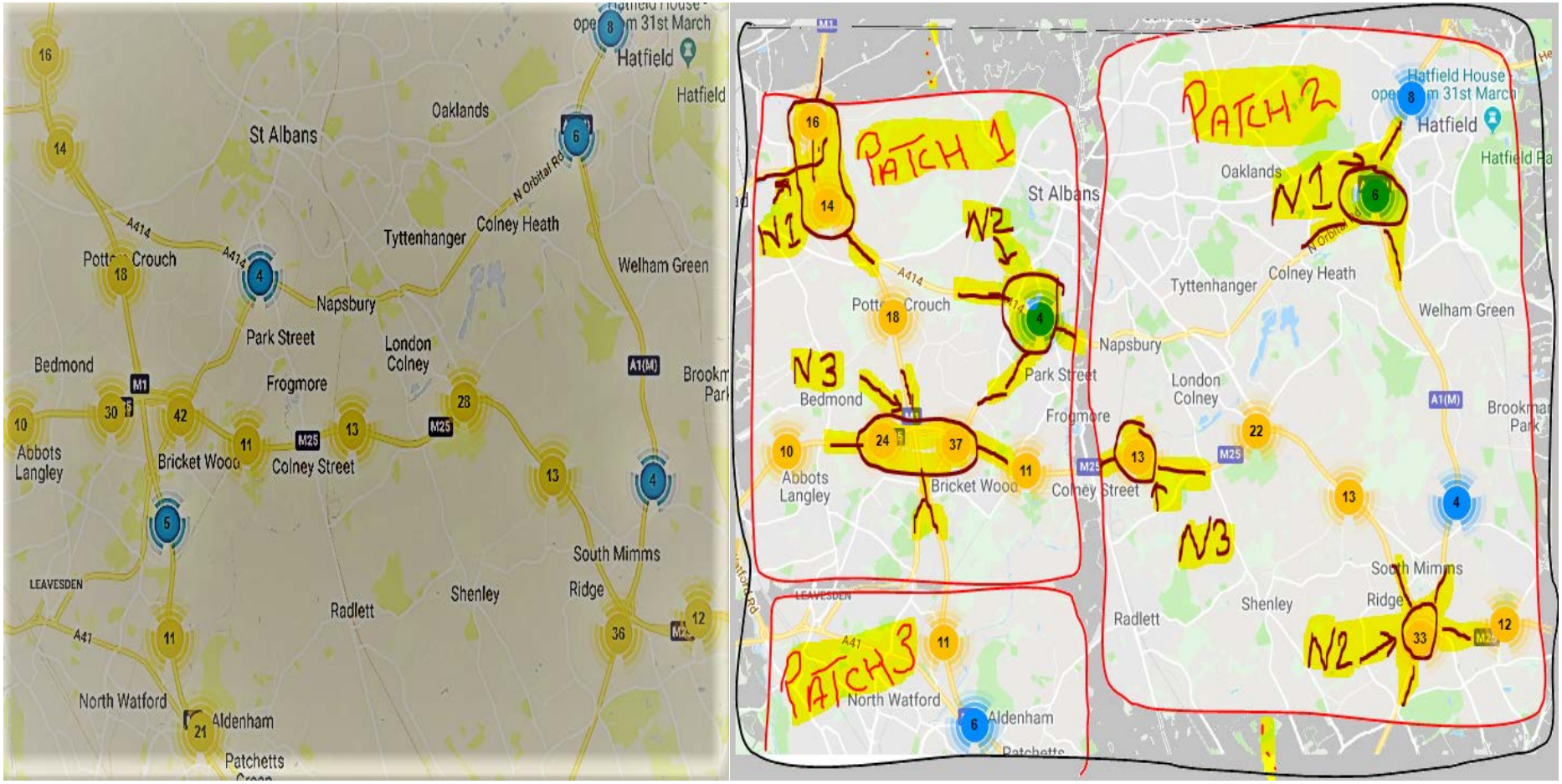
Considering the UK only data, Highways England (refer table 4.1, No. 10) is responsible for most of motorways and major category A roads in England. Highways England has outsourced his National Traffic Information Service (NTIS) for seven years to a joint venture between Mouchel and Thales, called Network Information Services (NIS) ltd. NTIS has installed equipment at regional control centres to interface with the various subsystems of the Highways England Traffic Management Systems (HATMS). This equipment provides access to the Motorway Incident Detection and Automatic Signalling (MIDAS) traffic data and high occupancy alerts with the ability to set variable message signs and to receive variable message sign (VMS) network signal settings via the message sign and signal subsystems of HATMS. Traffic data is also collected from traffic monitoring units and travel data from automatic number plate recognition (ANPR) cameras located at strategic locations on the network. Both categories of data are collected 5-minute intervals once processed the data is accessed by the subscribers. NTIS collects traffic data from various sensors and make them available in two different forms as isolated sensors data or fused sensors data [91]. HE has categorised data collection sites into three types: MIDAS, TAME and TMU. MIDAS sites which are mostly equipped with inductive loops although few sites also being used for the radar technology trials. Some sites collected data for traffic appraisal, modelling and economics (TAME) purposes only using inductive loops. In an analogy to TAME sites, some sites are equipped with traffic monitoring units (TMU) only.

NO.	Dataset	Description	Location	Suitability	Source
1.	Bus Breakdown and delays	The Bus Breakdown and Delay system collects information from school bus vendors operating out in the field in real time. (2015-2017)	New York	Dataset can be exploited to consider the transport operator performance and the delays caused by their vehicle breakdowns impacting the overall road traffic delays.	Kaggle https://www.kaggle.com/zx4724/bus-breakdown-and-delays-analysis/data
2.	Annual Traffic Volume (ATV) (Major by direction and Minor Roads) – DFT Dataset	Traffic figures give the total volume of traffic on the stretch of road for the whole year and are calculated by multiplying the Annual Average daily flow (AADF) by the corresponding length of road and by the number of days in the years. Traffic figures are presented as: Units = thousand vehicle miles	(covers most sites with minor road data estimates) UK	This dataset can be used to create a prediction model for urban road occupancy and can be used as a separate feature with other traffic prediction models as well.	Department for Transport (DFT) https://www.dft.gov.uk/traffic-counts/download.php [92]
3.	Annual Average Daily Flow (AADF) (Major and Minor Roads) – DFT Dataset	AADF figures give the number of vehicles that will drive on that stretch of road on an average day of the year. AADF figures are presented as: Units = vehicles per day	(covers most sites with minor road data estimates) UK	Average daily flow is aggregated into this dataset which is good for predicting flow per days in an urban network considering major and minor roads.	Department for Transport (DFT) http://www.dft.gov.uk/traffic-counts/download.php [92]
4.	RTA Freeway Travel Time Competition Data	The NSW Roads and Traffic Authority has made 2 years' worth of historical data on road use between 2008 and 2010 available.	NSW, Australia	Suitable for freeway travel predictions for intra sensor points. Not enough to be incorporated into network level models.	Kaggle https://www.kaggle.com/c/RTA

5.	Road Traffic Estimates statistics in Great Britain	The National Statistics publications of road traffic estimates for Great Britain are released on an annual and quarterly basis and provide summary statistics at national, regional, and local authority level.	UK	Not much junction on road level information in the data so it can be employed into the junction level prediction models.	Department for Transport (DFT) https://www.gov.uk/government/collections/road-traffic-statistics
6.	1.6 million UK Traffic Accidents	The UK government amassed traffic data from 2000 and 2016, recording over 1.6 million accidents in the process and making this one of the most comprehensive traffic data sets out there. It's a huge picture of a country undergoing change.	UK	It's an incident log so can be used to predict the road incidents on roads according to the weather, road conditions, accident severity and seasonality along with the time of the day. However, the traffic incidents data can be used as an additional feature for the flow prediction model.	Kaggle https://www.kaggle.com/daveianhickey/2000-16-traffic-flow-england-scotland-wales
7.	Road Traffic Accidents	Information on accidents across Leeds. Data includes location, number of people and vehicles involved, road surface, weather conditions and severity of any casualties	Leeds, UK	It's an incident log so can be used to predict the road incidents on roads according to the weather, road conditions, accident severity and seasonality along with the time of the day. However, the traffic incidents data	UK Government Data Website https://data.gov.uk/dataset/road-traffic-accidents

				can be used as an additional feature for the flow prediction model.	
8.	Road Safety Data- Accidents and Causalities 2016	These files provide detailed road safety data about the circumstances of personal injury road accidents in GB from 1979, the types (including Make and Model) of vehicles involved and the consequential casualties.	UK	This dataset can be incorporated into the flow prediction model to predict the breakdown likelihood of specific vehicle.	UK Government Data Website https://data.gov.uk/dataset/road-accidents-safety-data/resource/91789e37-03e5-48cf-9720-2d13639c32b9
9.	University Bus Company (UNO)	Bus Routes transactions, Automated Vehicle GPS Locations (AVL), Automatic Passenger Counts (APC), Scheduled Bus Arrival and Departure vs Real time Data.	UK	Useful to explore the route performances for routes, prediction models development for predicting passenger counts for stops and predicting the bus stop arrival times.	University Bus company Ticketing Systems Logs.
10.	Highways England Network Journey Time and Traffic flow Data – MIDAS/TAME/TMU Dataset	Highway and major road statistics. Contains the logs of the speeds and the average speed and traffic flow.	UK (selected sites)	The most compact and comprehensive dataset found expressing the traffic flows and their average speeds Suitable for traffic flow prediction models.	UK Government Data Website https://data.gov.uk/dataset/highways-england-network-journey-time-and-traffic-flow-data Highway England Portal http://tris.highwaysengland.co.uk/ [93]

Table 4. 1 Potential Dataset Finds.



a) *b)*

Figure 4. 1 a) Original Sample chosen test area with circles (yellow for MIDAS sites and blue for TAME sites. b) showing the sensors installed at the test sites by Highway England authority. b) Square red line boxes indicate the virtually divided network.

4.4 Data Description

Different data sources have different recorded parameters some of them have common parameters i.e. timestamp of the log, vehicle flow etc. Data is recorded through sensors activity at the model sites. The sensors used were loop based data from traffic monitoring units (TMU) and journey time was inferred using ANPR equipment in case of Highway England gathered dataset. The sensor loops in the road surface measured the actual speeds, vehicle flows and occupancy whilst travel times between two points was measured using ANPR camera recognition. If one of the loop was deemed faulty on the site it was reported and the flow value were imputed from previous values and not the vehicle category and speeds [91]. The following two datasets are chosen due to their suitability towards the testing and validity of our proposed network methodology, which are further sketched out in deep details as below.

5.4.1 MIDAS/TAME/TMU Dataset

Highway England provides data for every fifteen-minute period since April 2015 on all the motorways and category 'A' roads managed by highway England, termed as Strategic Road Network in England. Category 'A' major roads are freeway or dual carriageways and motorways. The Motorway Incident Detection and Automatic Signalling (MIDAS) original Gold dataset is logged everyone minute. It contained certain rules based upon which the gathered data at the site was logged and the basic ones are as: publication time, speed (threshold: 240 km/h), vehicle flows (threshold: 120 veh/min), occupancy and headway is reported on a per lane basis. Vehicle flows are divided into five categories depending on each individual vehicle length and are determined by the roadside installed traffic monitoring equipment. These categorised vehicle flows were converted to the volumetric unit of vehicles per minute values for each lanes and were fused together to get readings of the carriageway [91]. Table 4.2 shows the important data fields in the MIDAS traffic flow dataset. The files are generated monthly for each model site. Each file just contains flow, speed and day type logs from the major highways, junctions and motorways since they are all managed by HE. TMU sites that reports every five minutes contains almost similar data fields to the MIDAS and TAME datasets with the only exception that MIDAS dataset contained the data for motorways only. Whereas at TAME and TMU sites, loop only technology was not implemented on motorways but rather on carriageways and normal major roads. The data was logged every five minutes in the case of of TMU sites and it contained fields like speed, flows, occupancy and headways reported on per site basis averaged across all the lanes of carriageway. Figure 4.1 shows the breakdown of the dataset into different categories from Highway England. Traffic flow data related to TAME sites contains some additional fields in addition to MIDAS legacy data fields and they are shown in table 4.3.

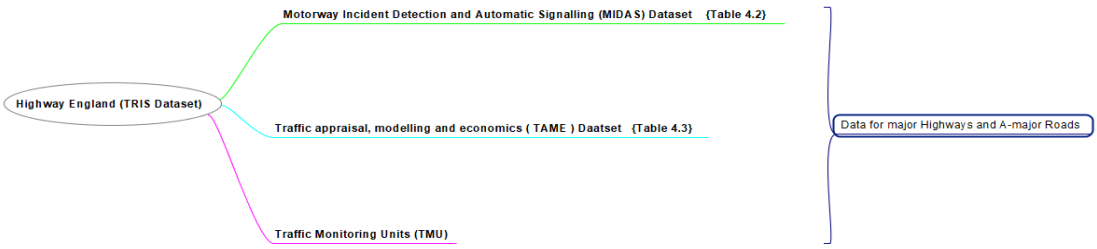


Figure 4. 2 Highway England Dataset Breakdown.

Data Field	Description
Total Carriageway Flow	The number of vehicles detected on any lane within the 15-minute time slice.
Total Flow vehicles less than 5.2m	The number of vehicles less than 5.2m detected on any lane within the 15-minute time slice.
Total Flow vehicles 5.21m - 6.6m	Number of vehicles between 5.21m - 6.6m detected on any lane within the 15-minute time slice.
Total Flow vehicles 6.61m - 11.6m	The number of vehicles between 6.61m - 11.6mn detected on any lane within the 15-minute time slice.
Total Flow vehicles above 11.6m	The Number of vehicles above 11.6m detected on any lane within the 15-minute time slice.
Speed Value	The average speed in km/h. of all vehicles for all lanes measured by the site over the 15-minute period.
Day Type	The following are valid: • 0 - First working day of normal week; • 1 - Normal working Tuesday; • 2 - Normal working Wednesday; • 3 - Normal working Thursday; • 4 - Last working day of normal week; • 5 - Saturday, but excluding days falling within type 14; • 6 - Sunday, but excluding days falling within type 14; • 7 - First day of school holidays; • 9 - Middle of week - school holidays, but excluding days falling within type 12, 13 or 14; • 11 - Last day of week - school holidays, but excluding days falling within type 12,13 or 14; • 12 - Bank Holidays, including Good Friday, but excluding days falling within type 14; • 13 - Christmas period holidays between Christmas day and New Year's Day; • 14 - Christmas Day/New Year's Day.
Quality Index	The Indication of the quality of the data provided. The number of valid one-minute records reported and used to generate the Total Traffic Flow and speed. A quality index of 0 indicates no valid records.
Network Link Id	An identifier unique to the NTIS link.

Data Field	Description
Average Speed in MPH	The average speed of vehicles per NTIS link for the 15-minute time slices.
Category 1 Speed Count	The average count of vehicles detected by the TAME site with a speed less than 10 mph in the 15 minutes time for all lanes.
Category 2 Speed Count	The average count of vehicles detected by the TAME site with a speed between 10 to 15 mph in the 15-minute time interval for all lanes.
Category 3 Speed Count	The average count of vehicles detected by the TAME site with a speed between 15 to 20 mph in the 15-minute time interval for all lanes.
Category 4 Speed Count	The average count of vehicles detected by the TAME site with a speed between 20 to 25 mph in the 15-minute time interval for all lanes.
Category 5 Speed Count	The average count of vehicles detected by the TAME site with a speed between 25 to 30 mph in the 15-minute time interval for all lanes.
Category 6 Speed Count	The average count of vehicles detected by the TAME site with a speed between 30 to 35 mph in the 15-minute time interval for all lanes
Category 7 Speed Count	The average count of vehicles detected by the TAME site with a speed between 35 to 40 mph in the 15-minute time interval for all lanes.
Category 8 Speed Count	The average count of vehicles detected by the TAME site with a speed between 40 to 45 mph in the 15-minute time interval for all lanes.
Category 9 Speed Count	The average count of vehicles detected by the TAME site with a speed between 45 to 50 mph in the 15-minute time interval for all lanes.
Category 10 Speed Count	The average count of vehicles detected by the TAME site with a speed between 50 to 55 mph in the 15-minute time interval for all lanes.
Category 11 Speed Count	The average count of vehicles detected by the TAME site with a speed between 55 to 60 mph in the 15-minute time interval for all lanes.
Category 12 Speed Count	The average count of vehicles detected by the TAME site with a speed between 60 to 70 mph in the 15-minute time interval for all lanes.

Category 13 Speed Count	The average count of vehicles detected by the TAME site with a speed between 70 to 80 mph in the 15-minute time interval for all lanes.
Category 14 Speed Count	The average count of vehicles detected by the TAME site with a speed greater than 80 mph in the 15-minute time interval for all lanes.
Category speed counts included flag	This denotes whether there are speed bin values present. Possible values are: • 0 - Not Present; • 1 - Present.

Table 4. 2 Traffic Flow, Additional field names and description features unique to TAME Dataset [94].

Data Field	Description
AADF Year	AADFs for each year (from 2000 onwards).
CP (count point)	Unique reference for the road link that links the AADFs to the road network.
ONS GOR Name	Former Government Office Region that the CP sits within.
ONS LA Name	Local authority that the CP sits within.
Road	This is the road name (for instance M25 or A3).
R Category	The classification of the road type.
iDir	Direction of travel.
S Ref E	Easting coordinates of the CP location.
S Ref N	Easting coordinates of the CP location.
A-Junction	The road name of the start junction of the link.
B-Junction	The road name of the end junction of the link
LenNet_miles	Total length of the network road link for that CP (in miles).
FdPC	AADF for pedal cycles.
Fd2WMV	AADF for two-wheeled motor vehicles.
FdCar	AADF for Cars and Taxis.
FdBus	AADF for Buses and Coaches

FdLGV	AADF for LGVs.
FdHGVR2	AADF for two-rigid axle HGVs.
FdHGVR3	AADF for three-rigid axle HGVs.
FdHGVR4	AADF for four or more rigid axle HGVs.
FdHGVA3	AADF for three or four-articulated axle HGVs.
FdHGVA5	AADF for five-articulated axle HGVs.
FdHGVA6	AADF for six-articulated axle HGVs.
FdHGV	AADF for all HGVs.
FdAll_MV	AADF for all motor vehicles.

Table 4. 3 AADF dataset common Data field names and description [93].

5.4.2 AADF Dataset

From the available datasets in table 4.1, AADF dataset is managed by DFT authority. It contained the average annual daily flows (AADF) and traffic flow counts which gave the opportunity to analyse the network on a street and minor roads level. Unlike highway England dataset, this dataset covered not only the Motorways, A-class roads but also the minor roads that includes B-class, C-class and certain urban unclassified roads. However, overall the minor roads data is not as comprehensive dataset when compared to the major roads this is due to the fact that minor road estimates were only gathered at some of the sample points [92]. Some of the AADF major roads data fields that made the DFT dataset more favourable for study and made it a better fit for use in the experimentation and testing, are shown in table 5.4. AADF data figures are produced for each junction to junction link on the major roads for every year. AADF stands for average over a full year of number of vehicles passing a point in the road network each day. Figure 5.3 gives the DFT dataset basic network road topology-based breakdown.

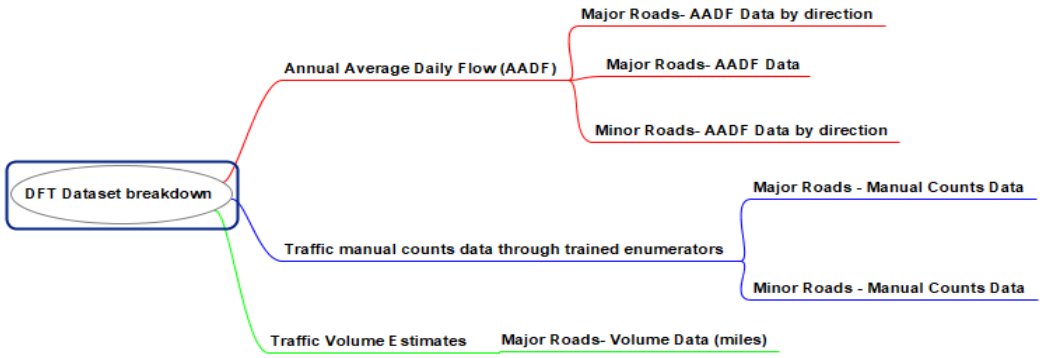
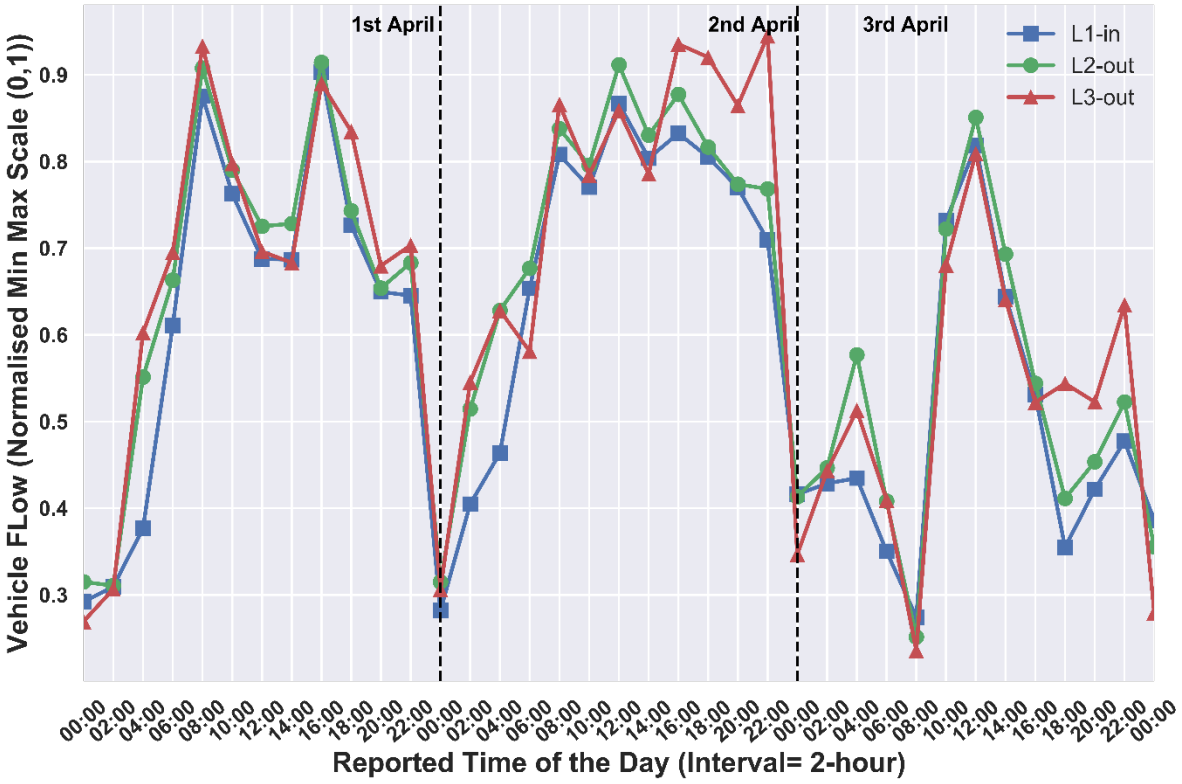


Figure 4. 3 DFT Dataset Breakdown.

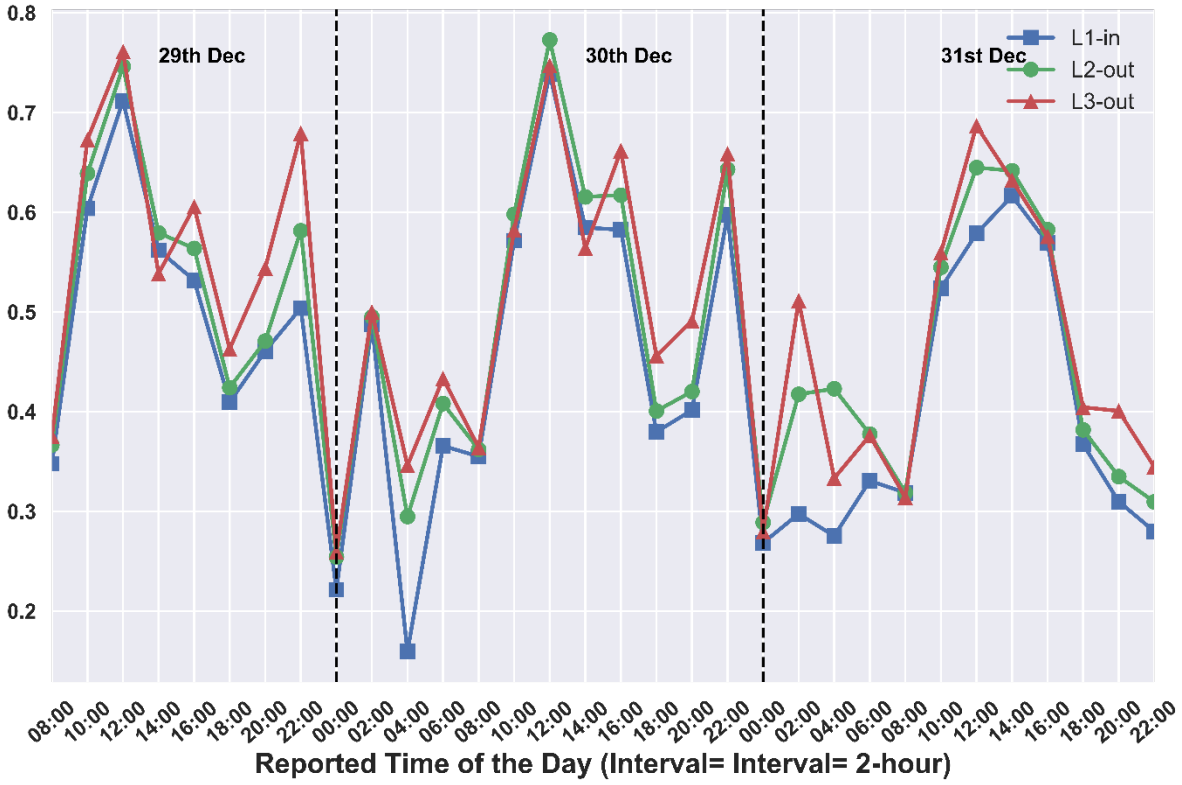
4.5 Data Preparation

The operational process of flow-based predictions is a multi-stage process (refer Figure 5.8). The process starts with a series of live data streams containing the time series data covering all the concerned nodes or junctions of the road network initially chosen in the study area. Like any machine learning algorithms, the incoming real time data is tested on the trained classifier model to predict the prediction variables. In a conventional machine learning model implementation, the validation scores are calculated on the validation set to compare the performance efficiency and prediction accuracies of the tested algorithms.

All the experiments are performed on the traffic flow MIDAS dataset for the Hatfield Hertfordshire UK area junctions as shown in figure 4.1 a & b. The used dataset contained traffic flow information for two-hour timed aggregated intervals from start of 1st April 2015 to the end of 31st Dec 2015 for the highway roads. First three and last three raw dataset plots from patch 1 node 2 links (refer figure 4.1 b.) are shown in figure 5.4. After the data collection process data preparation process involved gathering the relevant data fields for the model development. As mentioned in section 5.4, the data was collected for the number of passing vehicles using the loop detectors technology installed on both the ends of the selected highway links. The data pre-processing is carried out using the steps as below:



a)



b)

Figure 4. 4 a) First and b) last three days of pre-processed data from Patch 1, Node 2 associated Links.

4.5.1 Data Cleaning

The raw link dataset had approximately fifteen percent of values that were missing. Due to the ongoing trends comprising of seasonality and other environmental factors it is very important to retain the inherit trends in the traffic data. So, the missing values are imputed using the backward fill approach. In backward fill approach the flow value is imputed using the next interval original recorded value. This imputation process was continued until all the missing values were imputes. Although the data inconsistency was resolved but this technique can danger the inherit data properties if the missing value rates is too high.

4.5.2 Data Integration

A total of 3252 data samples are used for each considered link. They are reshaped according to the equation 4.1, to form an array of [3252, number of flows considered on the junction]. The dimensionality of 3252x4 is considered in the current experiment considering the link predictions associated with the patch 1 node 2 as a test case. This reshaping is performed in the case only when c more than one feature is considered for the predictions on a node.

4.5.3 Data Normalisation

After the data aggregation and reshaping is done it is further generalized and normalized by scaling for the minimum and maximum values among each data column. i.e. intra flow links normalization. Further the reshaped dataset is lagged by one-time interval, two-time interval and three-time intervals to make it suitable for the supervised training in case of short, medium- and long-term forecasts which is further discussed in the experiments section.

4.5.4 Data Reduction

With the aim to generate the training and validation sets to train and validate the ML 30% of the original dataset is considered as the validation set. K fold cross validation is performed thrice for every chosen model with the validation and testing data. Since it's a time series consecutive interval data the order of training and validation ensemble is very important. Therefore, the tail end 30% series values are considered for the validation of trained models after each training iteration.

4.5.5 Data Discretisation

For the multi feature prediction model scenarios, among the originally reported dataset there are twelve intervals in a twenty-four-hour time window. Only the twelve intervals are considered which are two hours apart each to make the ML models training not only fast but a more generalized representation of the sequential data throughout the day.

4.5.6 Dependent and Independent Data Variables

Literature review have shown that many different types of variables have been used for the analysis of traffic flow prediction problems. Some of the common variables that have been used in previous studies, includes spatial variables (traffic flow for a link road), temporal variables (time of the sampled data) and seasonal variables (weather conditions). A broad understanding of the AADF and DFT datasets field variables revealed that it can be classified into two main categories: independent variables and dependent variables. The selection of dependent and independent variables from the datasets are subjected to the suitability of our proposed aims and model development along with keeping the performance measures inline from previous studies. The data preparation which involved fetching the data from the data sources and light pre-processing with a bit of data cleaning is done. Data is gathered for the designated test area as defined in the initial study area. The final filtered dataset at the end of data preparation contains the total vehicle flow for all the links on a junction for

the network area, compiled for different time interval i.e. 15,30,45,60 minutes. An initial insight into the data is to be developed along with the trends filtering for different independent variables like type of the day. Figure 4.4 a & b shows the results post data preparation as outlined in sub sections 4.4.1-4.4.4 for the sample area considered for experimentation as illustrated in figure 4.2 a & b.

4.6 Preliminary Analysis

Preliminary data analysis is second phase of the methodology process (refer figure 5.8). Preliminary phase involves the network sampling by dividing it in several patches and subsequently each patch into nodes according to the proposed strategy.

4.6.1 How Network Patch and Nodes Are Defined?

The road network is divided into a series of successive virtual geographical additive patches. Each patch contains different nodes defined by the road junction and links intersections (refer figure 4.1 b). Due to the comprehensive data availability for up to the type-A network roads only highway and category 'A' major roads are considered as the local and uncategorised local roads data is estimated and recorded buy the local county councils in United Kingdom. Figure 4.1 a show the original chosen area comprising of motorways and highway roads. The yellow and blue circles represent the sensors installed at the sites by Highway England authorities for the flow and journey data sampling (refer table 4.2 and 4.3). The yellow circles represent the motorway sites for the MIDAS datasets whereas blue circles are the representation of the TAME sites for general highway roads.

In our study the focus is majorly on the traffic flow, the causes of bottlenecks and the effects on the overall traffic travel times. The rules that we defined in our study, to declare a possible virtual network patches are as follows:

- First and for most the defined patch is considered an enclosed virtual geo-fence boundary defined system, to study the effects of the dependent variables (i.e. total flows for the links, the resulting journey and the inflicted travel times) in that patch.
- A patch must consider the minimum of one node in it. The node is defined by the sensor site with the aggregate of the current data availability for all the intersecting road links.
- A patch (n) acts like an independent system with its own inputs (traffic flows) source origins from subsequent patch (n-1) and likewise the output to be dumped into another successive path (n+1) as shown in figure 5.5.
- Order of exploiting the patches data in our ML model is of extreme importance. Order is important such that the traffic flows output of one system is the input of the other system in line.

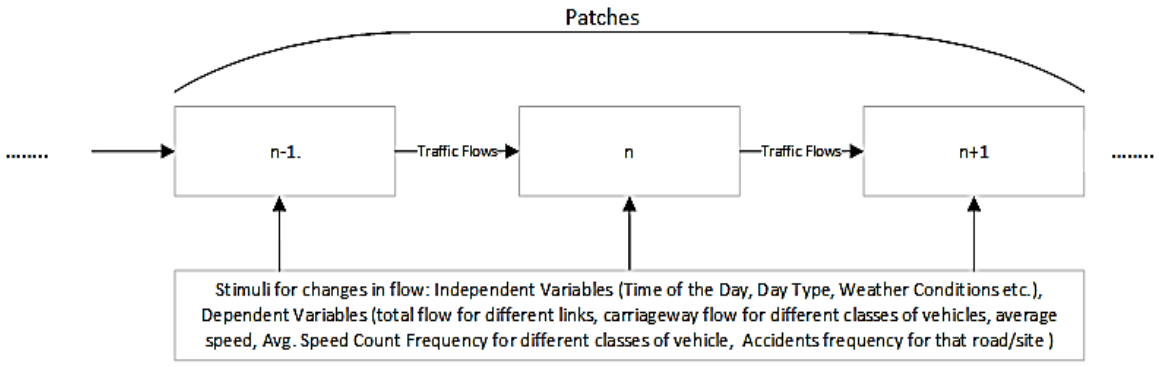
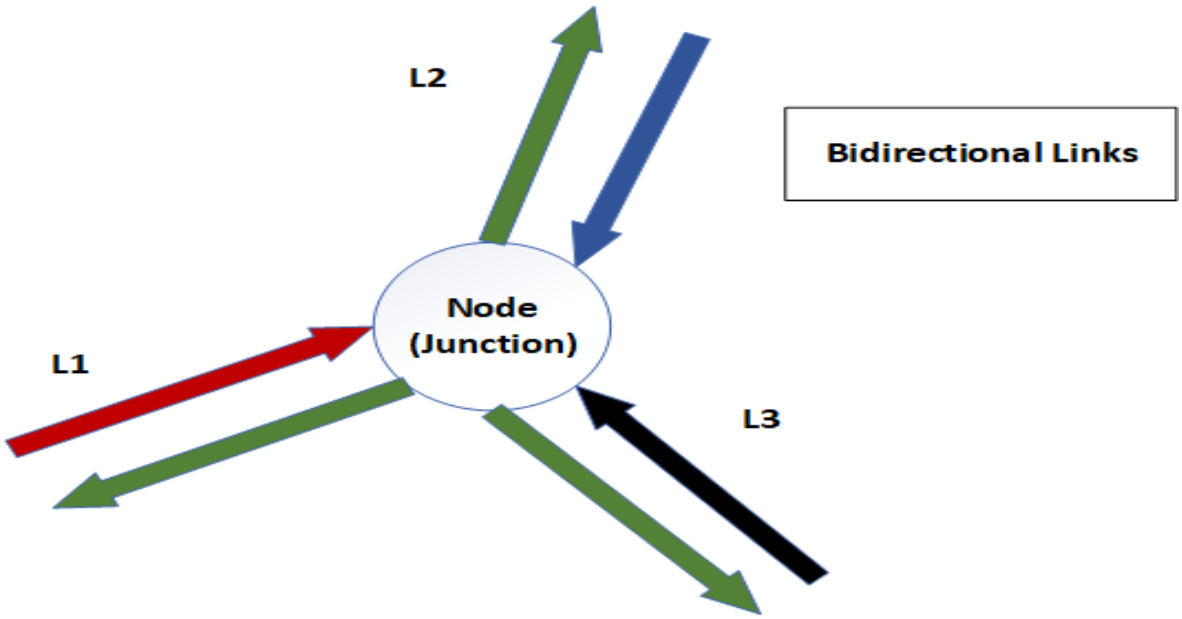


Figure 4. 5 Systems as Network of Patches.

4.6.2 Preparing the Dataset Subset for Each Node of a System Patch

After nodes identification in a patch the next step is to gather and prepare the dataset subset for the patch and this is achieved by compiling the dataset for all the nodes and their associated bidirectional links in that node. The subset dataset for each patch includes the total traffic flows for each link on all the nodes. Initially It is thought to keep the methodology and experimental simple by just considering one junction as shown in figure 5.6 a.

a)



b)

Figure 4. 6 a) P_1-N_2, Highway junction under consideration (Google Maps, 2018). b) Node illustration retaining junction original topology.

Figure 4.6 a) shows the original patch one, node junction two along with the associated bi-directional links. Figure 4.6 a) shows the node N_2 marked along with the road links forming the node. This node comprises of four road links (L_1, L_2, L_3, and L_4). All the road links in figure 4.6 a) are two-way links which signifies that they bear not only the burden of the incoming flow of traffic but also the outgoing traffic flow. All the links considered in this case belongs to the motorway category of highways more specifically, L_2, L_3 belongs to the A1(M) whereas L_1 is part of North Orbital road. Links are numbered according to clockwise rule with the first link being the one that falls in the zero to ninety-degree range and the later follow in a sequence. Table 4.5 shows the final subset dataset for node N_3 given only the links division with field header variables for the whole of the patch P_1 (refer figure 4.1 b). Patch P_1 filtered dataset header fields are shown in table 4.5 to convey the concept of links flow divisions, because it's an example of diverse node which contains varying number of links N_i i.e. three, four and four for nodes N_2, N_1 and N_3 respectively.

$\boldsymbol{N_1}$	$\boldsymbol{N_2}$	$\boldsymbol{N_3}$
$L1_{in}$ & $L1_{out}$	$L1_{in}$ & $L1_{out}$	$L1_{in}$ & $L1_{out}$
$L2_{in}$ & $L2_{out}$	$L1_{in}$ & $L1_{out}$	$L2_{in}$ & $L2_{out}$
$L3_{in}$ & $L3_{out}$	$L1_{in}$ & $L1_{out}$	$L3_{in}$ & $L3_{out}$
$L4_{in}$ & $L4_{out}$	----------	$L4_{in}$ & $L4_{out}$

Table 4. 4 Links Divisions for Patch P_1(refer figure 4.1 b).

4.7 Methodology

In this section, the proposed traffic model representation is presented. In view of the proposed methodology the traffic model is considered as a set of nodes with corresponding inputs and output links. The traffic flow for a set of links will have an influence on the traffic flows of the output links. The traffic model is considered as a block box interpreting and modulating the system inputs. As system is governed by a set of rules associated with the fixed and dynamic states which are mapped to the outputs. This is shown in graphical form with the mathematical expressions as in figure 4.7. Such as each individual road links for a node can be modelled as an objective function consisting of variable parameters as shown in figure 4.7.

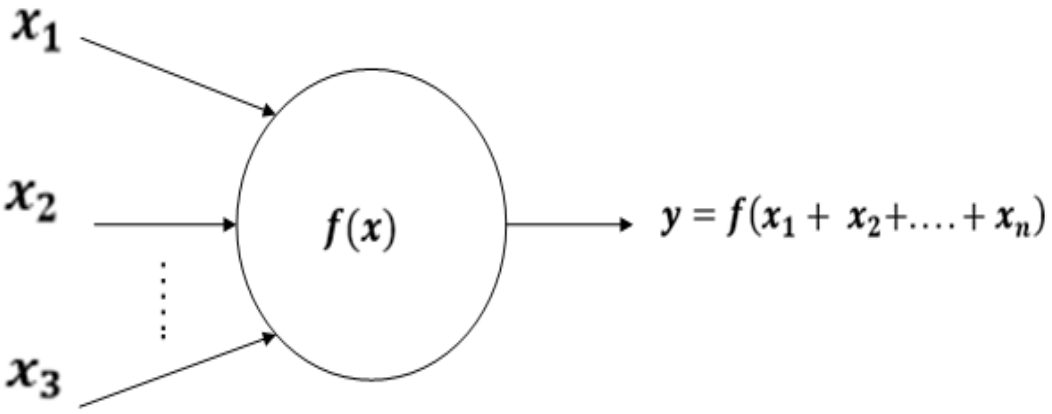


Figure 4. 7 General Network Node Link Dependencies Written in An Analogy with The General Function Definition.

.

4.7.1 Traffic Network Representation on a Junction Level

Each spatially located junction with its inflows and outflows is an independent system. Each network junction is designated as a node denoted by N_x , where x gives the node number in a patch to which the node belongs. As the highway links are bidirectional, the link, represented by $Lx_{in\ or\ out}$ can be an inflow (in) and out flow (out), where x is the number of links associated with a node under consideration. As an example, for the experimentation and consideration of the proposed mechanism, a simple node in figure 5.6 a) is considered and it's equivalent representation using the nodes and the links configuration is illustrated in figure 4.6 b). Further, the bidirectional arrows indicate the bidirectional traffic flows of the nodes. Here outflow implies traffic flow moving away from the node and inflows to those moving into the node.

4.7.2 Formulation of Network Flow Estimation Function

To predict the outflow of traffic for each individual link on a single node, all the incoming link flows are to be considered for the output flow forecast objective function. They key here is to retain the spatial topology of the original link with in the forecasting objective function. The outflow of a node's link is determined by summation of inflows of individual links of the node. Figure 4.6 b) shows that the output flow associated with ta link is dependent on the inflows of every other link in the same node [94, p. 7160]. The estimated traffic outflow for the link $L1_{out}$ is given by equation 4.1, showing the dependency of the flow objective function on the inflows associated with the rest of the links of the same node. Equation 4.2 is a more general objective function mathematical representation which describes the conservation of flow with a node system where x is the link for which the flow is being calculated and n is the total number of links in the same node N_2. This makes the objective function retain the correlations in the flow characteristics for each individual node link when the single node is considered as a basic unit level in the traffic network. Equation 4.3 gives the aggregated sum for a patch with the summation of involves nodes respective inflow or outflows. Starting from a micro link level to the macro network patch representations while keeping the network spatial topology intact.

$$L1_{out} = f\ (\ L2_{in}\ +\ L3_{in}\ +\ L1_{in}\ \ )\qquad \{\ L1,\ L2,\ L3\ \in\ (\ N_2)\ \} \tag{4.1}$$

$$L(x)_{out}\ = f\ (\ L\ (\ n-x)_{in}\ )\qquad \begin{cases} x,\ n, & \in\ same\ N \\ & x < n \end{cases} \tag{4.2}$$

$$(\ P_N)_i = \sum_{n=1}^{N}(\ N_n\ )_i\qquad \{\ \forall\ i\ \in\ [in, out]\ \} \tag{4.3}$$

4.7.3 Node Level Traffic Flow Mathematical Representation

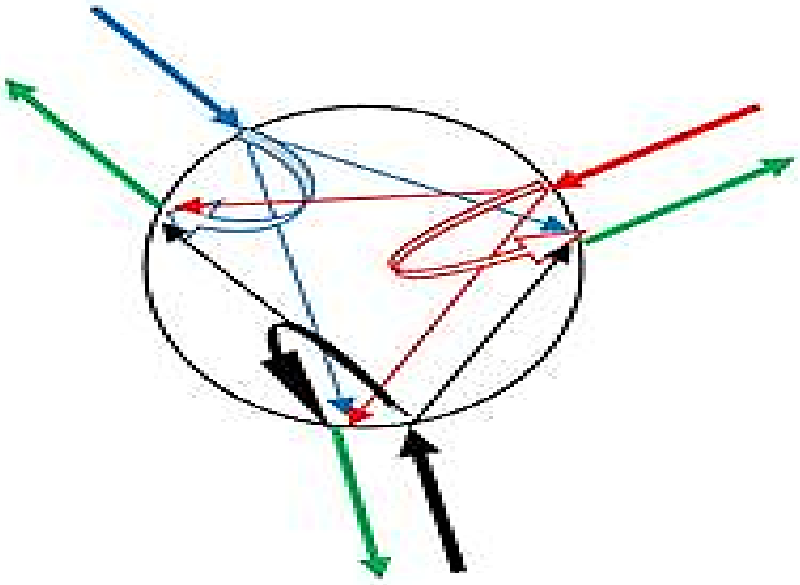
With reference to the figure 4.6 b), let the node i consisting of a set of attached links $L(j)$ that are associated with bi-directional traffic flows F_i . Each link $L(j)$ at node i is associated with traffic inflow $F_i\ (\ L(j)\ )_{in}$ and a corresponding outflow indicated by $F_i\ (\ L(j)\ )_{out}$. The function for the traffic flow of links $L(j)$ considered the fact that the traffic inflow of every link contributes partially and to a certain a degree to the outflow of each of the other links at the same node. In other words, the traffic outflow of a link is a function of the traffic inflow of all the other links including its own at the node. This is given in the mathematical representation as in equation 4.4.

$$F_i\ (\ L(\ j\ )_{out}) = \sum_{j=1}^{n} \frac{F'\ (\ L\ (\ j\ )_{in})}{F_i\ (\ L\ (\ j\ )_{in})} \tag{4.4}$$

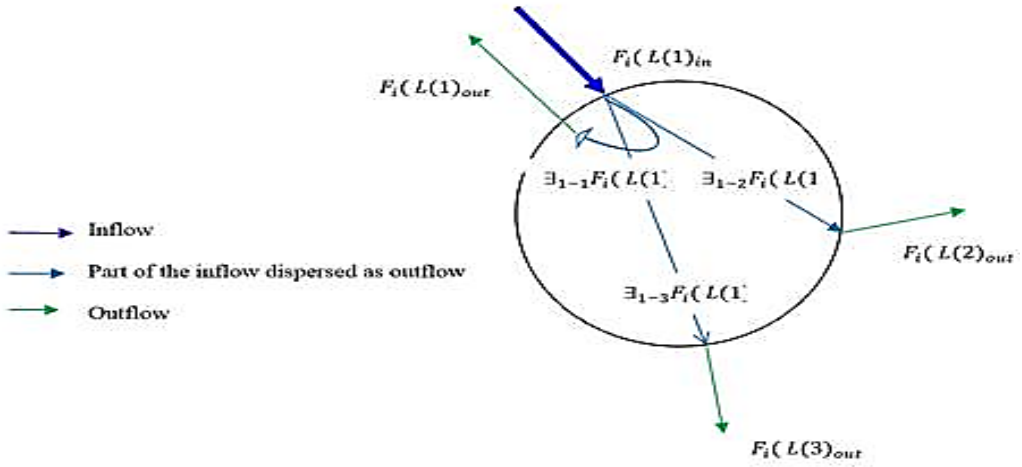
Where in equation 4.4, $F'\ (\ L\ (\ j\ )_{in})\ /\ F_i\ (\ L\ (\ j\ )_{in}\ )$, represents a fraction of the traffic inflow that is contributed to an outflow of a specific link.

This above mathematical representation is further illustrated graphically in figure 4.8 a & b. In figure 4.8 a) the circle represents a node i with three links. The thick blue arrow indicates the traffic inflow

of link $L\ (1)_{in}$ that gets dispersed into the node and flows through the rest of the links. This show that the flow contributes to the outflow of the rest of links including itself. This dispersion indicated by thin blue arrows in figure 4.8 a). The outflow of each of the links in figure 4.8 b) is shown in green arrows. The symbol $\exists_{1-j}$ indicates that part of the inflow of link $F(\ L(1)_{in}\ )$ contributes to the outflow of the links $F(L(1)_{out}\ )$. The sum of the traffic flow of $F(\ L(1)_{in}\ )$ inside the node represented by thin blue arrows is equal to the traffic inflow of $L(1)$ represented by a thick blue arrow, at a time instant as shown in figure 4.8 a). This applies to the traffic inflow of all other links at the node as shown in figure 4.8 b).



a)



b)

Figure 4. 8 a) Extension of traffic network at node i showing three links and their associated inflows and outflows. b) A simple traffic network at a node i with 3 links. It shows the distribution of incoming traffic dispersed as outgoing traffic at the node.

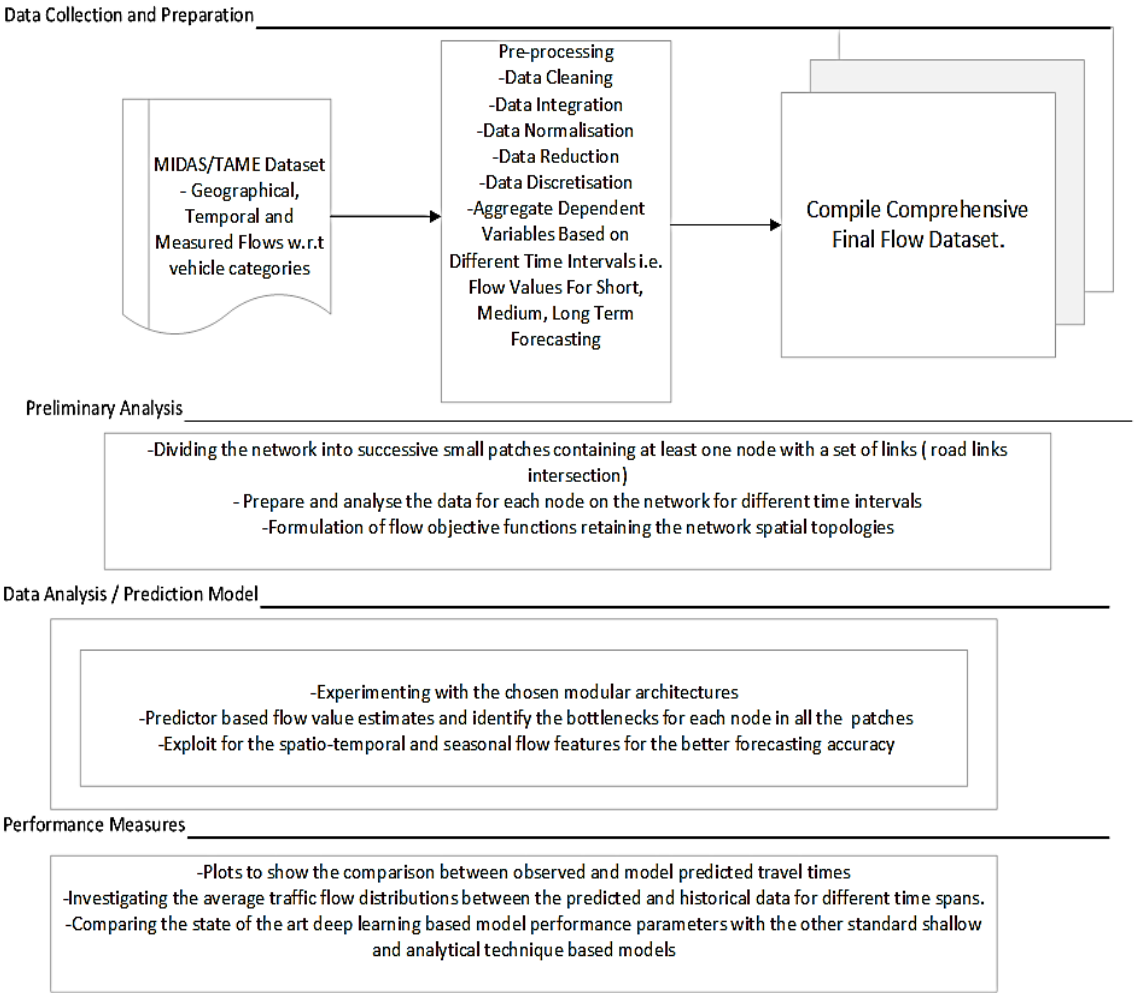


Figure 4. 9 Implementation Steps for The Proposed Methodology.

4.8 Summary

In this chapter, possible gathered datasets are discussed in detail along with their suitability to aim of this research and the chosen study area for the data collection. After a thorough data description, MIDAS dataset as the chosen dataset, is passed through a set of data preparation steps which involve cleaning, integration, normalisation, data reduction and discretisation techniques. Possibility of dependent and independent variables in the dataset are also explored. Network division into patches and further into nodes with attached traffic road links is presented. The topology based proposed network methodology to be employed for ML models is discussed in detail at the end of this chapter.

5. Experiments and Results: Evaluation of The Proposed Frameworks

In this chapter, the preliminary analysis is done to decide upon the best analytical methods. Then the proposed research methodology to predict the highway traffic flow predictions in Hatfield area is discussed. The main aim of this chapter rests upon presenting the findings with an in-depth analysis of traffic flow prediction using hybrid DNN techniques. This chapter introduces the structure of how the experiments are performed based on the proposed methodology, chosen ML model techniques along with the reported results. Section 5.1 describes the experiment settings for experimental scenarios, which are given in section 5.2. Data correlation study is carried out in section 5.3. Experimental setup is described in 5.4 and section 5.5 lists the actual experimental results in detail.

5.1 Experimental Settings

In this section the performance metrics used to report the best performing individual models and evaluation methods for comparing different models are introduced. The chosen dataset is analysed further for correlation analysis along with the training and testing of the proposed models. Further the merits and demerits of the proposed methods, fusion of different modular architectures for traffic flow prediction is carried out.

5.1.1 Performance Metrics

Model performances are compared based on the evaluation done by the two prediction error estimates: The mean relative error (MAE) and root mean square error (RMSE) are used for the measurement of the different model accuracies. MAE measures the given model average magnitude of errors in a set of predictions. MAE is the average of the sample's absolute differences between prediction and actual value where all the individual differences are treated the same weight wise. RMSE is used for the comparison of accuracies among different models. The MRE and RMSE are expressed mathematically as given in Eq. (5.1) & (5.2), respectively.

$$MAE\ (y, y') = \frac{1}{T}\sum_{t=1}^{T} |y_t - y'_t| \tag{5.1}$$

$$RMSE\ (y, y') = \left\{\frac{1}{T}\sum_{t=1}^{T}(|y_t - y'_t|)^2\right\}^{1/2} \tag{5.2}$$

Where y_t and y'_t are the actual and predicted traffic flows at time t respectively. These performance indices do the job of measuring the linear score that averages the prediction error with the same weight as with RMSE and MRE it allows the relative residual error measurement by assigning larger weights to larger errors. It is also important to see how different models perform for different node links among different junctions. This is done by the empirical distribution function plots. Model accuracies are also analysed during rush peak and normal non-peak hours.

5.1.2 Evaluation Settings

The chosen error measures explained in section 5.1.1 gives the error over each directional link of the considered nodes. To determine the correct accuracies between models the Empirical distribution function (EDF) and k-fold validation is applied with the testing data. Firstly, the procedure of how this is done is introduced in this section. Finally, we also include the explanation of how the error estimates are compared for the result with data being filtered based on different criteria.

5.1.3 Empirical Error Distributions

Different models generate the multivariate level predictions for traffic flows. The factor that differentiates different models based on their performance is how well they predict for every link on the considered nodes. Error measurements among different models are highlighted using the cumulative distribution function (CDF) which is in the form of EDF. Let us assume for example $x \in X$, where X is the performance measure of the model in the form of the calculated RMSE or MRE for the prediction results for each node link. The distribution tells us how much of x is distributed, from the value of 0 up to the maximum sample distribution level of 1 in the sample space.

5.1.4 Error Distribution Comparisons

The dataset considered in section 4.5, does have many intrinsic characteristics involved in it based on the time of the day, day of the week, holiday period, normal working day, flow and speeds based on the different vehicle categories and some extrinsic characteristics including events and other natural weather factors involved as well. It is very important to analyse the dataset for these factors. Also, different model performances for these factors are also compared separately by limiting the data filter criteria for these factors before the performance measures are measured. Different model performances maybe better or worst based on the traffic flow for different times so the thresholding of the traffic flow volume is also done in some scenarios.

5.2 Experiments

This section sheds a light on experiments performed and the logical reason behind for doing them. **Firstly**, the experiments with model performances for different prediction time horizons are discussed, based on the chosen model architecture and what input data lag gives the better prediction results. In the **second** scenario, the effects on deep model results by including more variables beside the flow data are reported. It's the combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) variants i.e. LSTM and GRU that make up the deep learning models.

The existing models are to be outperformed if the newly proposed model is to be considered a better alternative. Thus, the chosen considered benchmarks including: The Auto Regressive Moving Average (ARIMA), Historical Average (HA), Random Walk (RW) or Random Forest Regressor (RFR), Support Vector Regressor (SVR), Feed Forward Backpropagation Neural Network (FFBNNs), Deep Belief Neural Network (DBNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTMs), Backpropagation-Long Short Term Memory -Neural Network (B-LSTM-ANN) and Deep Convolutional Neural Network – Long Short Term Memory (DCNNs-LSTM) (refer chapter 3.), are compared. HA and RFR constitute the bare bone simplest models considered here. As both the models signify two different abilities that they inherit. Firstly, any methodology to be considered as a better model must have the performance show of better than RFR as it suggests that the compared model has got the ability of meaningful data learning whereas HA is taken as a baseline performing model for trendiness. Any worst model performance than HA would suggest the inability and slackness of the data learning abilities.

Case 1: Prediction Interval

The interval lag is the time lag from the last observed values to the time step in future the model is trying to forecast. In this case different time steps or prediction intervals are considered for the better understanding of the model behaviour and usefulness. Thus, short interval or one-time step (fifteen - minutes interval), medium interval (thirty minutes) and long interval (sixty minutes) are defined and experimented with in three different cases of input data. The MIDAS dataset samples used in this research, are recorded at fifteen-minute intervals by default. This defines one interval index equals

fifteen minutes or one-time interval. But as mentioned in data pre-processing, the time intervals with two hours apart are considered for this experiment. So essentially next time step means the flow recorded in the last fifteen minutes and the preceding timestep will be two hours apart from the last recorded sample. To have a better understanding of the traffic flows, these intervals help understand the short (fifteen-minute, one-time step), medium (thirty minutes, two-time steps) and long-term (forty minutes or three-time steps ahead) reliability of the prediction models.

Case 2: Inclusion of Related Variable

In the second experiment case feature vectors can be formed including other related MIDAS dataset variables. The idea of including weather related variables, to improve upon the flow predictions is already considered in [95][96]. But in this research, we restrict the experiments with just flow data, but the approach can be adopted using the additional traffic information related variables from MIDAS dataset. All The available variables given by the MIDAS dataset are discussed in table 4.2. The choice of the considered variables in this experiment case are given as:

- Vehicle speed value (The average speed in km/h. of all vehicles for all lanes measured by the site over the 15-minute period)
- Total carriageway flow (The number of vehicles detected on any lane within the 15-minute time slice.)

So, 15-minute is the time index by which the data is recorded. Some of the potential recorded variables, over each time interval (15-minutes) from MIDAS dataset that can also be used in conjunction with flow and speed features as more meaningful feature vector are the basis of further problem solution, as listed below:

- Day type (Day of the week, normal week working days, first, middle and last day of the week school holidays and bank holidays, day of the year)
- Time of the day (the interval index of 15-minutes or one-time step). Times of the days are used as a further multi-feature multi feature deep end model training keeping the proposed objective function intact.
- Flow of different vehicle categories (Total flow of vehicles in range less than 5.2m, Total flow of vehicles in range 5.21m - 6.6m, Total flow of vehicles in range 6.61m - 11.6m, Total flow of vehicles in range above 11.6m).

5.3 Correlation Analysis

In this section we analyse the dataset for studying the relevancy in the feature's selection (auto-correlation) and the relevancy of the main selected features to other considered features (cross-correlation). The primary feature is the total carriageway flow as the main selected feature variable and secondary considered features are the time lagged versions of link flows.

5.3.1 Auto-Correlation

To check the dependence of different time intervals for the carriageway flow values, auto-correlation test is performed using the timed lag version of the flow data. This analysis tells if the number of previous time intervals (n-steps) that are relevant and have an effective correlator effect on future values corresponding to ahead time intervals (n-steps), so as the optimal interval steps can be considered in the prediction model. Figure 6.1 shows the auto correlation graph for the original traffic flow features data for the incoming link $L1_{in}$. Each lag is for the 15-minute interval time step. The blue shaded area represents the 95% of the confidence interval for the correlation coefficient. Any correlation coefficient past the confidence interval shows the existence of significant autocorrelation

between traffic flow at time (t) and (t-interval step). Traffic flow values exhibit a significant correlation for even up to forty lags in the past as can be seen from figure 5.1. Although after about twenty lags the correlation becomes periodic depicting the trendiness in the flow values, so the lags past the 20 lags can be discarded. The similar autocorrelation coefficient behaviour is observed in outgoing and incoming traffic of the other connected links as well. Next, we analyse the cross-correlation of different linked road links.

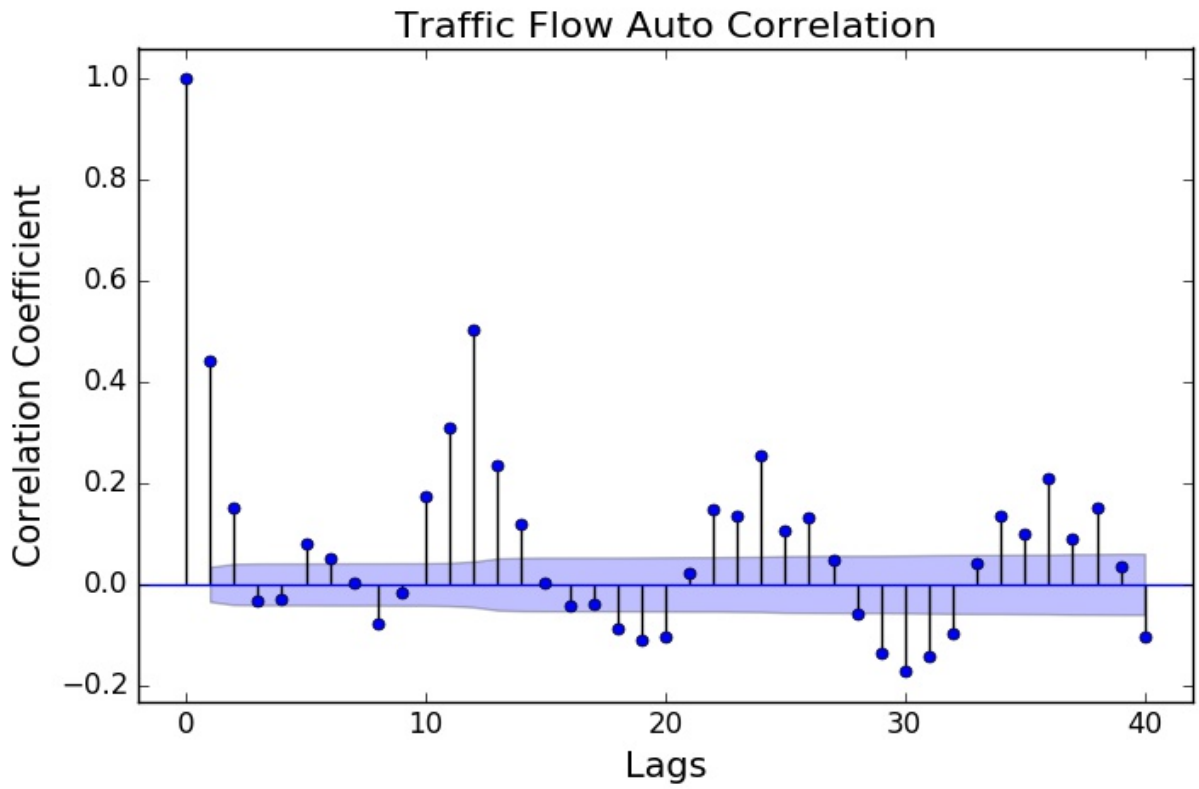


Figure 5. 1 Original Flow features auto-correlation for the incoming link $L1_{in}$.

5.3.2 Cross-Correlation

To cross check the dependence of different junction connected links traffic flows at different time steps, cross-correlation is performed. The cross-correlation results are shown below in figure 5.2. To properly understand the traffic flow parameters that determine the shape of traffic flow profiles, it is necessary to investigate the cross-correlation of connected traffic links. In this case we analyse the cross correlation for the past six-time intervals for the $\mathrm{L1_{in}}$. Cross correlation results are shown in figure 5.2. The labels $L1_{in}$, $L1_{out}$, $L2_{out}$, $L3_{out}$, $L1_{in\_1}$, $L1_{out\_1}$, $L2_{out\_1}$, $L3_{out\_1}$, in figure 5.2 refer to traffic flows related to link one (inflow), link one (outflow), link two (outflow), link three (outflow), link one (inflow with one-time interval lagged), link one (outflow with one-time interval lagged), link two (outflow with one-time interval lagged), link three (outflow with one-time interval lagged) etc.

respectively. The cross correlation between each pair of links are given by real numbers along with the colour map plot which represents the pair relevant cross correlation. The final plotted links lag set is shown in figure 5.3. In order to get the sense of the lagged linked pairs cross and auto-correlation Pearson coefficient was considered [97]. There exists a higher auto-correlation of the links with their own time lagged versions (Pearson coefficient > +0.5) but the cross-correlation fades away towards no correlation (Pearson coefficient = 0) and becomes non-linear (Pearson coefficient <-0.5), if this link lag increases. As in figure 5.3, the link $L1_{in}$ vs $L1_{in\_1}$, $L1_{in\_2}$, $L1_{in\_3}$, $L1_{in\_4}$, $L1_{in\_5}$, $L1_{in\_6}$ exhibits no linear or zero cross-correlation. And same is true for the same linked lagged pairs, where auto correlation is maximum but cross correlation is zero. But as more further time lags are considered the correlations become either non-correlated or non-linear. It is interesting to know that at later lag periods the cross correlation becomes nonlinear i.e. less than zero. For example, $L3_{out\_5}$, $L3_{out\_6}$ each individually have insignificant correlation with other lag pairs except when compared with other links fifth and sixth lags (Lx_{in_5}, Lx_{in_6}), where it exhibits very high cross-correlations resulting in a linear or close to linear correlations i.e. greater than zero Pearson coefficient.

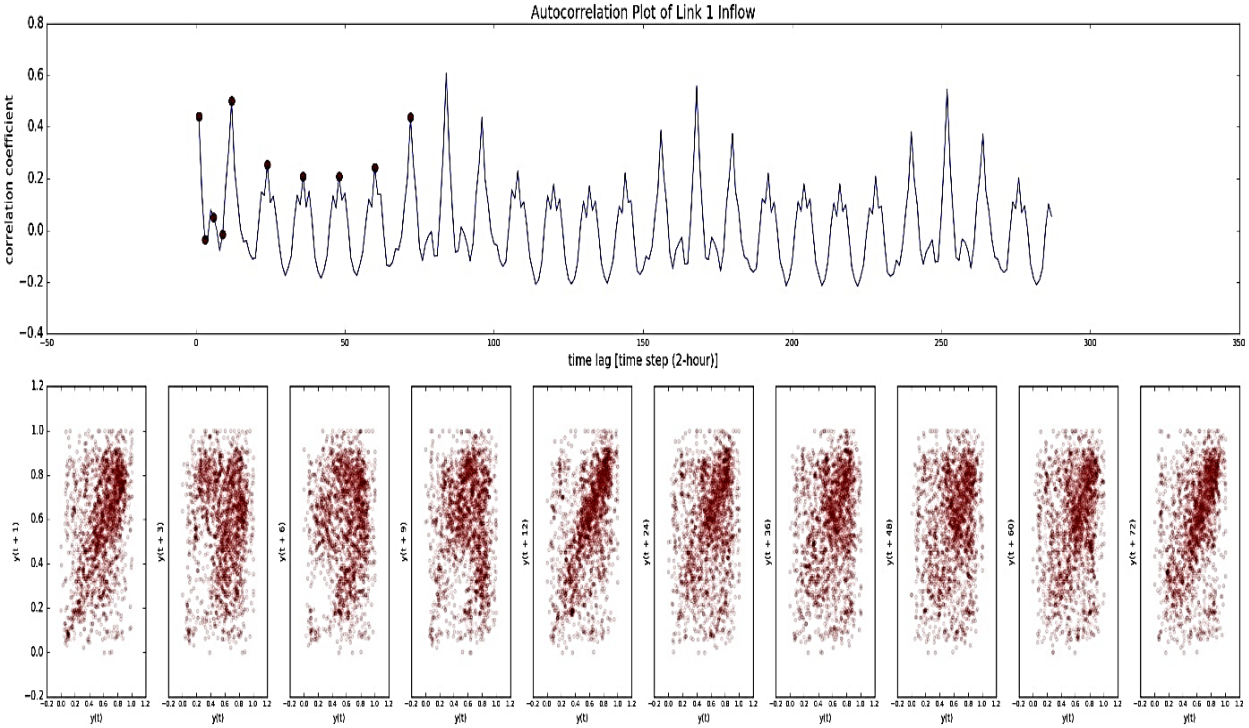


Figure 5. 2 Cross Correlation of Link $L1_{in}$ with it's Time Lagged Versions.

From the plot in figure 5.3 the following corollary can be drawn:

- The cross correlation of the links with their own lagged versions is almost zero at any interval whereas their auto correlation depends on the time lags as shown in figure 5.1.
- The lagged pairs have more linear correlation to the links nearest lagged versions suggesting the that the trend is flowing through to the next time lag and fading away gradually in the subsequent lags.
- There is a significant trendiness across links of the relating time lag hence they are more cross correlation.
- At any one-time lag consideration, suggests that the traffic flow distributes not evenly to the joined links but follows the flow conservation principle so much that the linked road link can have a nonlinear relation to the master inflow link regardless of other linked links on the same

junction. This fact supports our intuition for the proposed methodology mentioned in section 4.7.

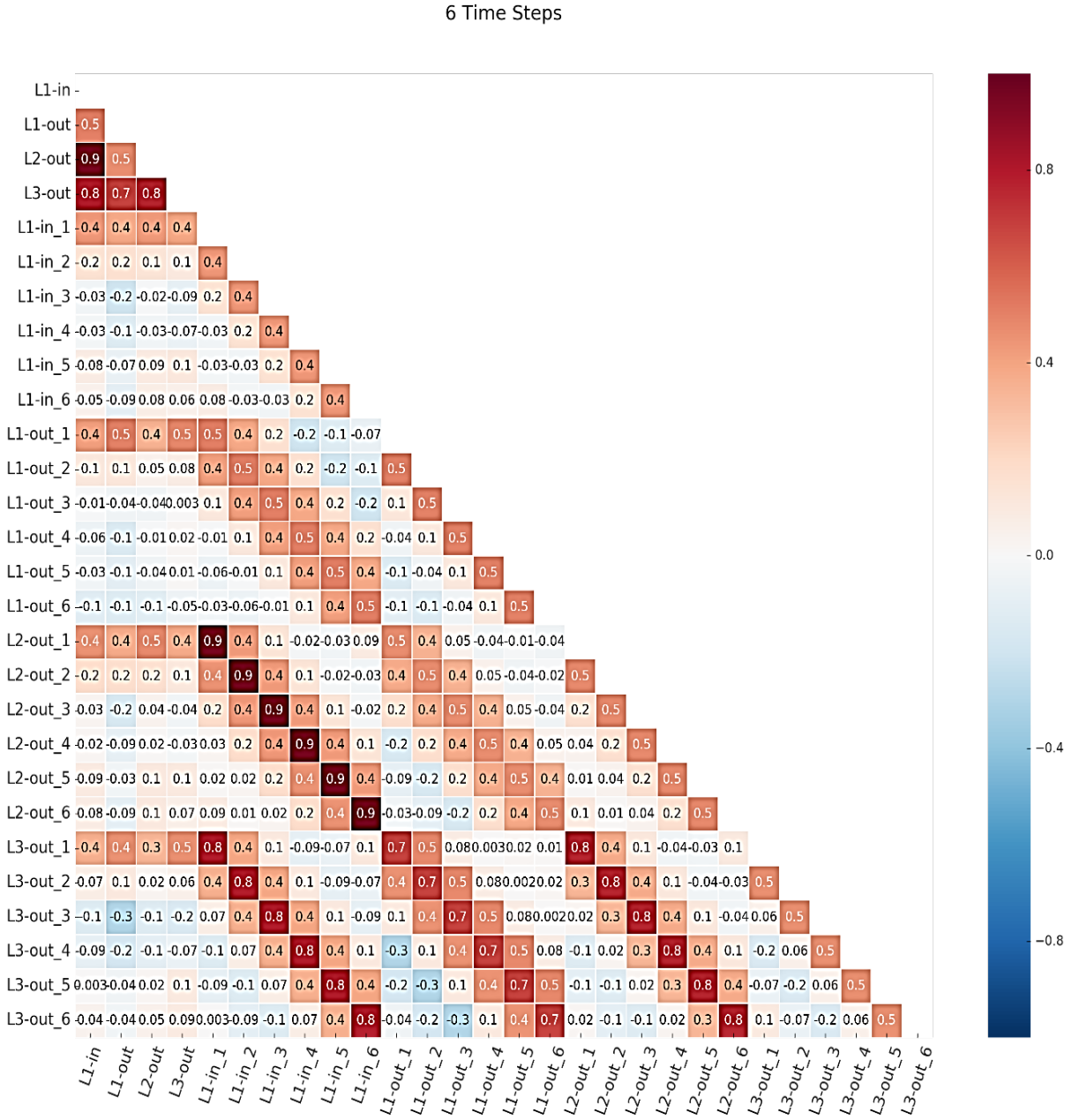


Figure 5. 3 Cross-Correlation of Connected Links for The Past Six-Time Steps.

5.3.3 Relation Between Traffic Flow Profiles and Times of the Day

Further, to understand the flow profiles of the connected links with respect to the time of the day. Figure 5.5 consists of two figures, a) being the links correlation pair plots with no time lags and b) is the plot with flow from six-time steps ago. It is very clear from figure 5.3 that the time of the day have a significant effect on the link flows. Although the incoming flow from $L1_{in}$ is distributed into $L1_{out}$, $L2_{out}$, $L3_{out}$ but the distribution of this division is time dependent. As can be seen in figure 5.5 b) $L1_{in\_6}$ exhibits a high-density flow distribution at 4th, 6th and 22nd hour of the day. Compared to

$L1_{out\_6}$, traffic flows at the 6th, 8th and 20th hour are mostly dominant. Which is because these hours fall in to the category of peak hours. On the contrary, $L2_{out\_6}$ get its share of flow maximum at the 6th, 10th and 16th hour of the day. And most of the flow out of $L1_{in\_6}$ taken by $L3_{out\_6}$, which is further apparent by the likewise flow density distribution profiles of both links. Comparison of other link pairs suggests the fact that peak flow in one linked is not distributed evenly to the connected flow receiving links. But the most crucial think to note is the time of the day. Figure 5.4 exhibits the relation $L1_{in}$ of flow profile for with respect to the times of the days. Based on these results traffic flow profile can be divided into three time slots: 1) Morning Peak Hours (04:00-10:00), 2) Normal Hours (10:00-16:00), 3) Evening Peak Hours (16:00-20:00), 4) Late-Night Off-Peak Hours (20:00-04:00). The average correlation for the evening peak hours between the link pairs falls below 0.5 this is because the traffic flow is reversing in the opposite direction and some of the original morning flow won't go through the actual incoming link channel and may take some other route. Whereas for the late-night off-peak hours the flow becomes minimum at first due to minimalistic traffic on road but then it starts to keep pace during its last hours to constitute the peak morning hours.

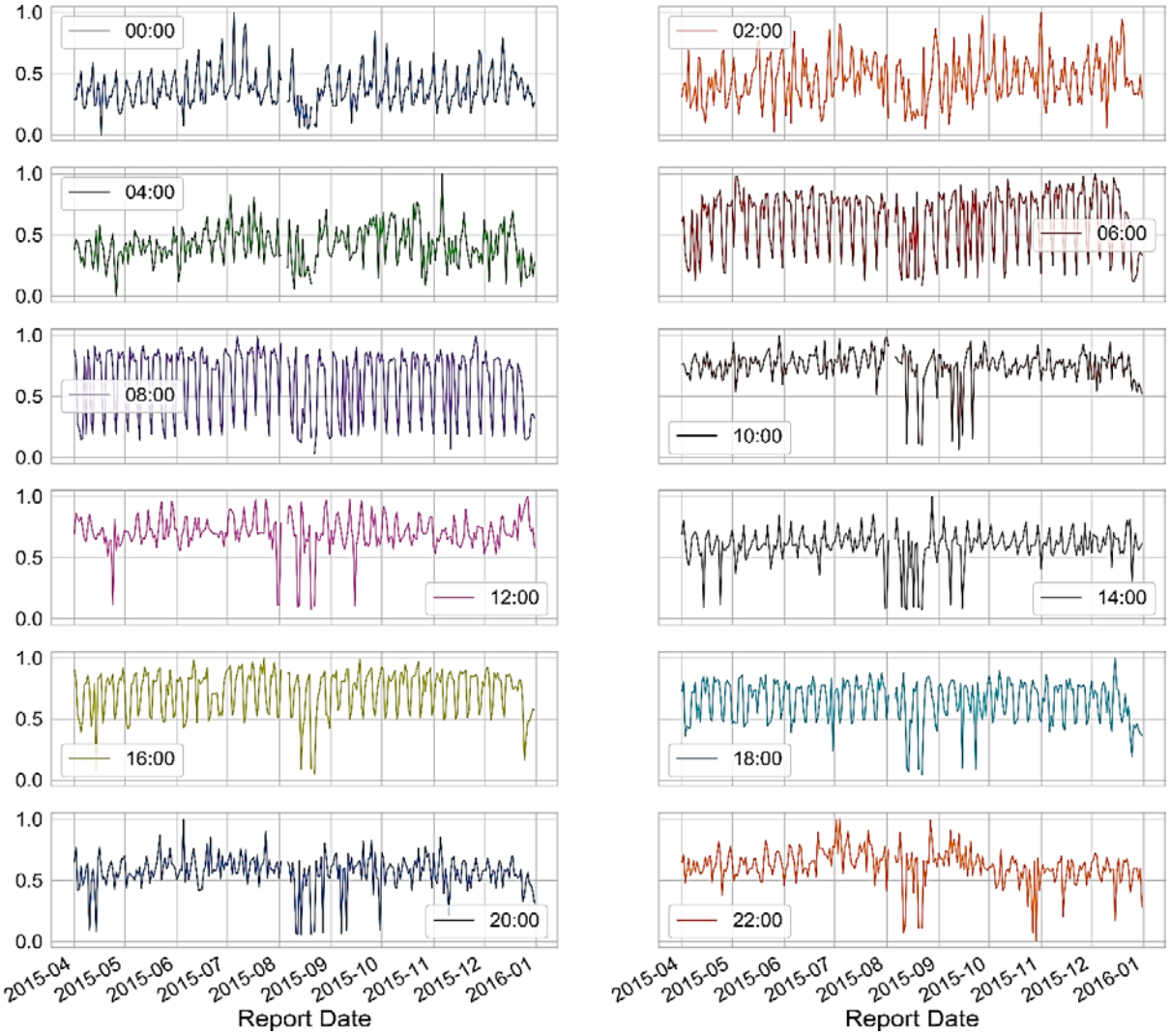
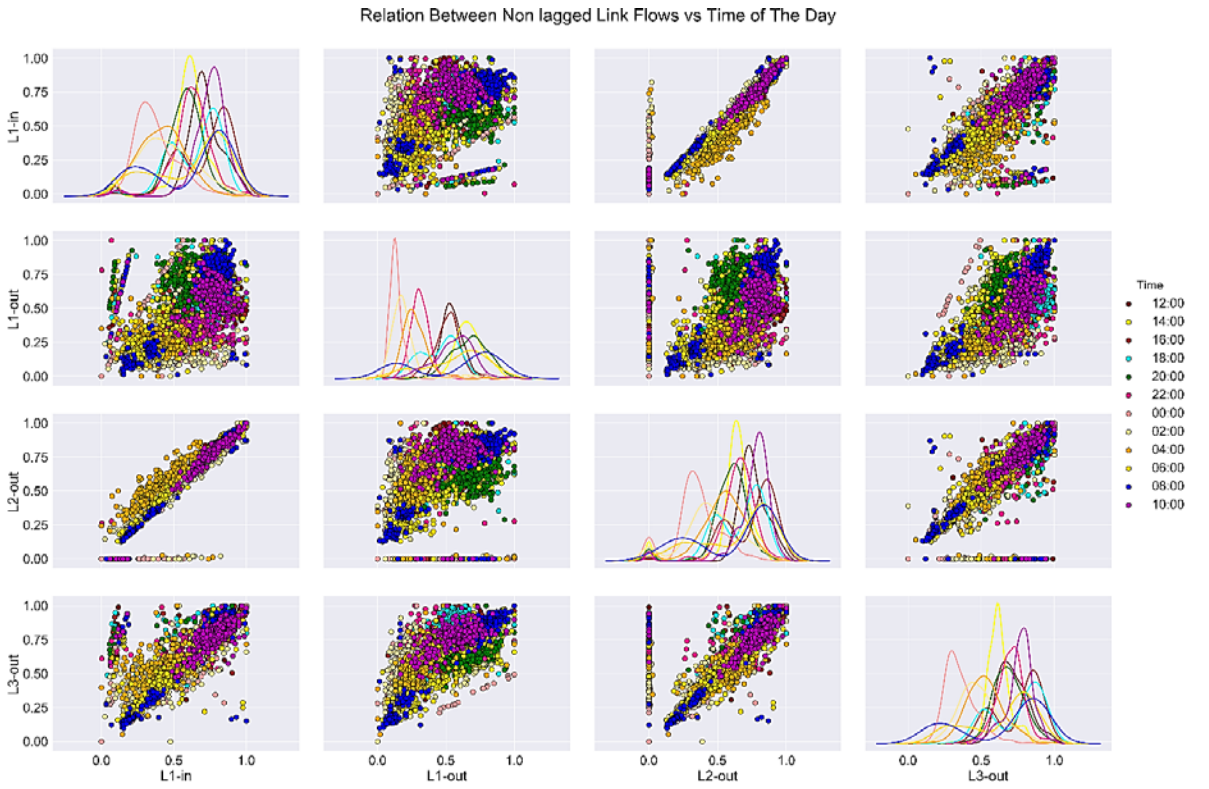
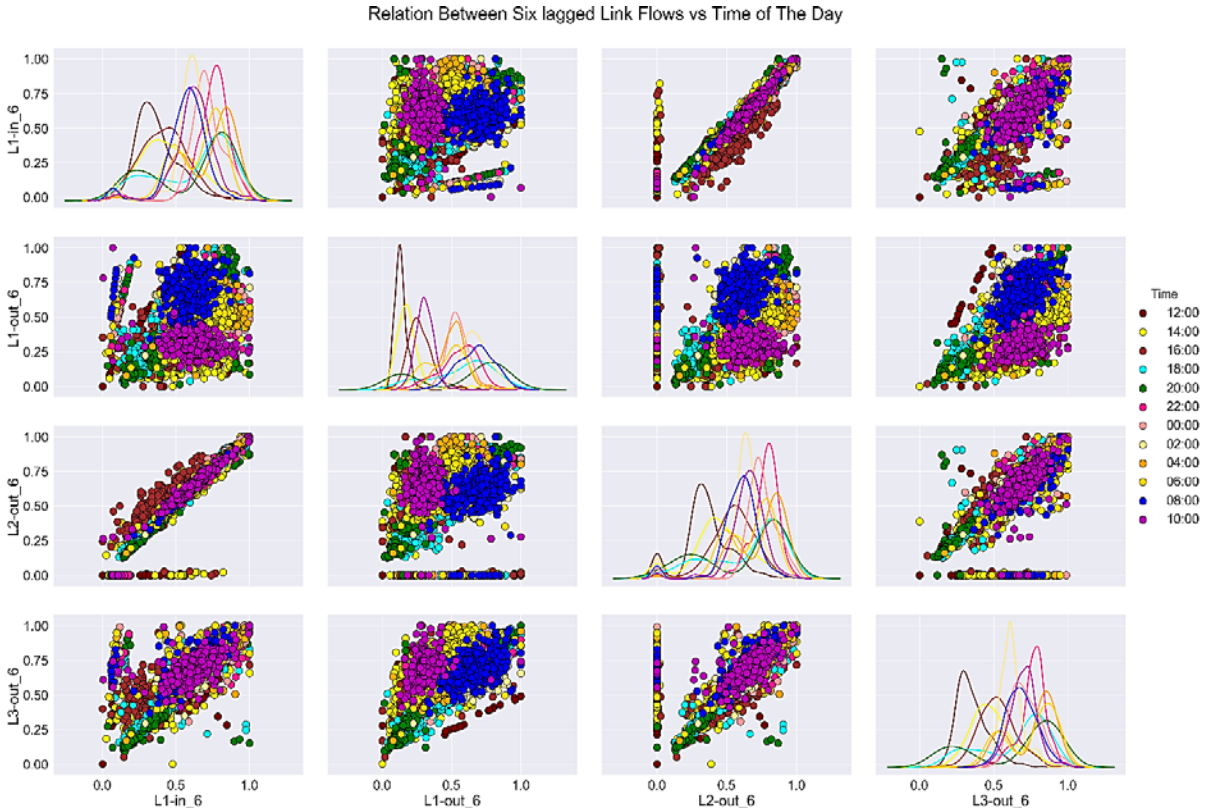


Figure 5. 4 Link $L1_{in}$ Normalised Flow Profiles with Respect to The Times of The Days.



a)



b)

Figure 5. 5 a) Correlation Between Non-Lagged Interconnected Link Pair Normalised Flows vs Time of the Day. b) Correlation Between Non-Lagged Interconnected Link Pairs Normalised Flows vs Time of the Day.

5.3.1 Seasonality and Trends in Traffic Flows

The final link flows are plotted in figure 5.5 a & b. and as illustrated in figure 5.4, the link flow pairs have a positive correlation for most of the times of the day. The obvious thing is that traffic volume increases with peak hours and decreases with off peak hours when the day light is getting dimmed. Furthermore, it is significant that traffic flow is inversely correlated to the speeds of the vehicles. During adverse weather and low visibility conditions average speed is lower than normal resulting in decreased traffic flows. These changing factors have a major effect on traffic flows especially traffic congestions. Traffic congestion may also be the function of seasonality and trends. From figure 5.5, in both the plot pairs it is seen to have some level of flow at a certain time of the day that is highly correlated, but not so correlated in the lagged versions of the pair plots. This explains that traffic flow profiles are highly seasonal dependent. Also, because in winter season conditions with low visibility the difference between the peak and normal hours gap is shortened. This difference in flow behaviour is more discernible by the density distribution curve shift of link pairs for the morning peak hours and normal hours of the day as can be seen in figure 6.6 in the trend plots. Traffic flow trend may remain constant for most of the year as it's the function of regular road users, but it is the seasonality that makes the flow exhibit major variations due to the density of the road being used at any time as shown in figure 6.6. Figure 6.6 shows the seasonality breakdown of the four links previously being discussed. Since the original observed data is gathered at two-hour intervals, the additive decomposition at a frequency rate of two months (sixty days) allows to see the periodicity very well in the trends.

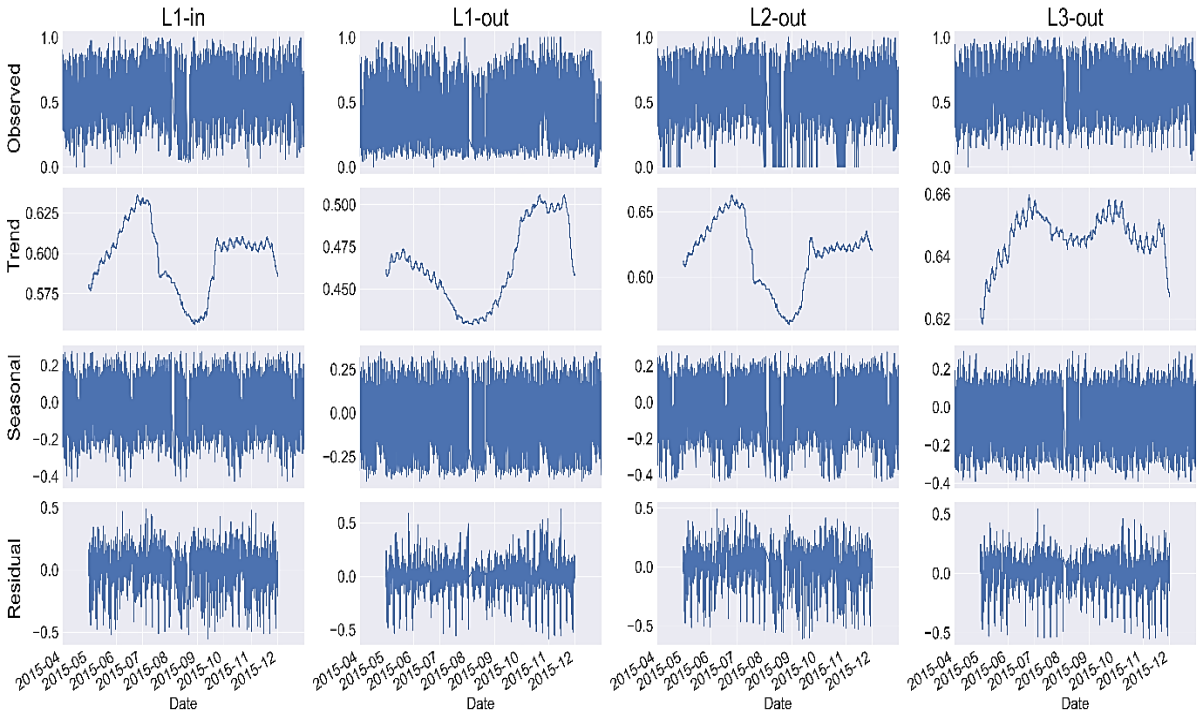


Figure 5. 6 Links Seasonality Breakdown

From the results of Figure 5.6 it can be inferred that there is a significant amount of seasonal component involved in the traffic flows. Eliminating the seasonal component and trendiness in the flow profiles gives the unaffected residual traffic flows that remains pretty much the same for any link. As expected, there is a clear seasonal reverse shift in the seasonal plots which is captured in the trend plots as well. Indicating that the summer traffic volume does gets changed when the winter season starts. The trend plots on the other hand do reflect that traffic flows or traffic volume does dip for a

while when the days are shorter and the winter time changes but then it starts to get higher when the days are back to their nominal length at the end of the December. Likewise, the flow densities may differ for different days of the week, but the overall flow profiles almost resemble the general flow profiles. The example of this behaviour is shown in figure 5.7 for $L1_{in}$.

The conclusions made in these sub sections authenticates that above discussed facts that interconnected links does influence the intra link traffic flows for a specific time of the day and this effect is a seasonal one based on the localisation of the links. Therefore, these hidden features are further sought to be explored in our proposed flow prediction architectures.

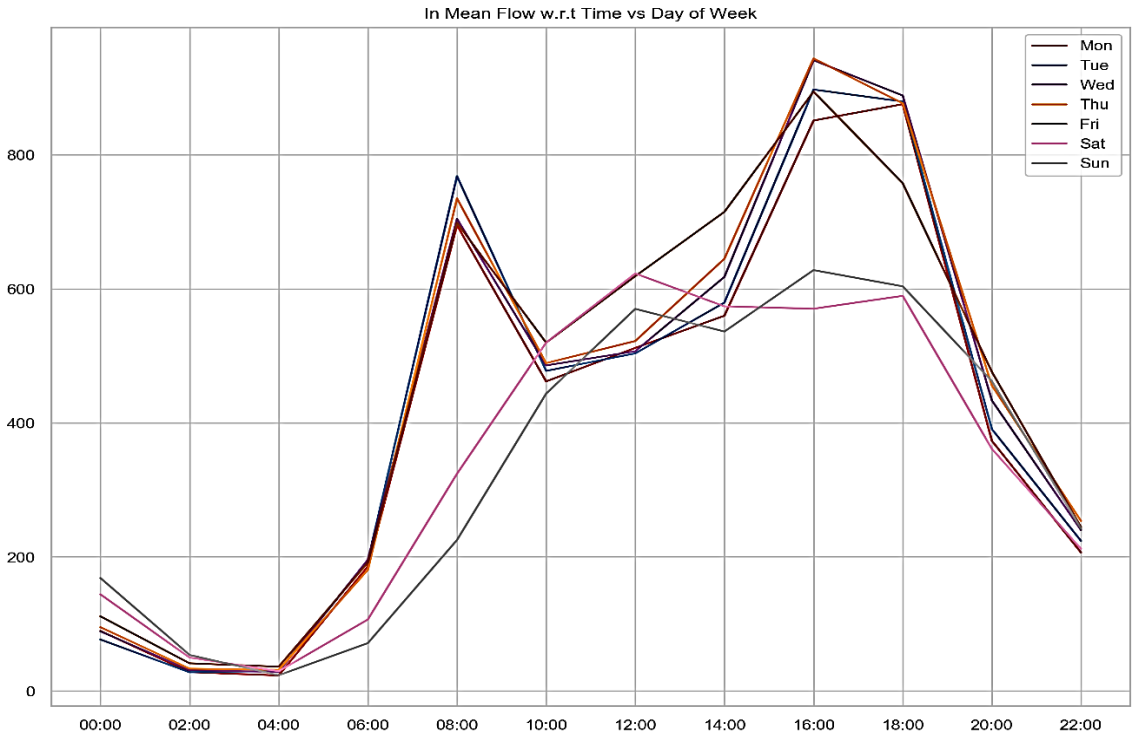


Figure 5. 7 Link $L1_{in}$ Flow Profiles with Respect to The Times of The Day Along with the Days of the Week Breakdown.

5.3.2 Seasonality and Trends in Traffic Flows

The temporal dependency in traffic flow time series explains the inherent dimensions of data that are time dependent. This temporal order suggests that that time dependent consistencies need to be handled in a special manner. In terms of statistical modelling, these observations are ideally considered consistent. In the time series analysis, it's referred to as the time series being stationary. Since the traffic flow time series in our case has, he seasonality and trend characteristics attached to it. So, the stationarity of time series needs to be checked for the effectiveness of our applied forecasting algorithms. A visual plot might be the obvious stationary test for the time series. Figure 6.8 shows the monthly averaged original flow observations plot for all the four links. The plots for all the four links look stationary just from the look of it. The observations in stationary time series are not dependent on time. A stationary time series is easier to model as statistical methods require the series to be stationary for them to be effective for forecasting it. To further confirm the stationarity of the flow series Augmented Dickey Fuller (ADF) Statistical test [98] is performed.

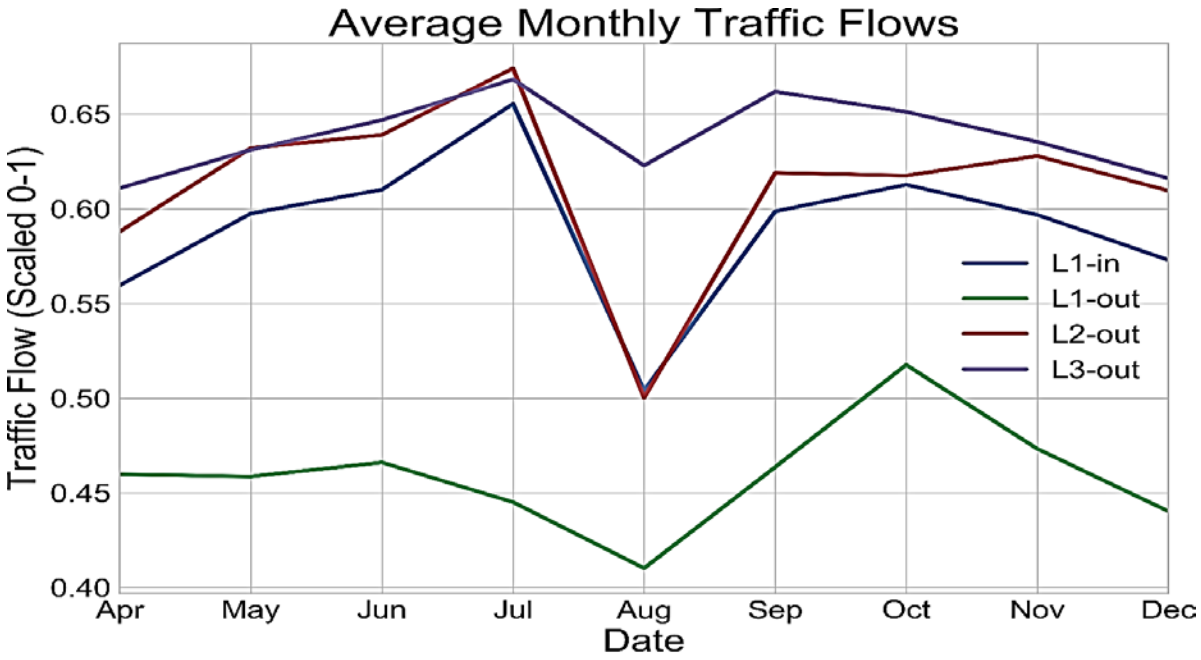


Figure 5. 8 Averaged Monthly Traffic Flows.

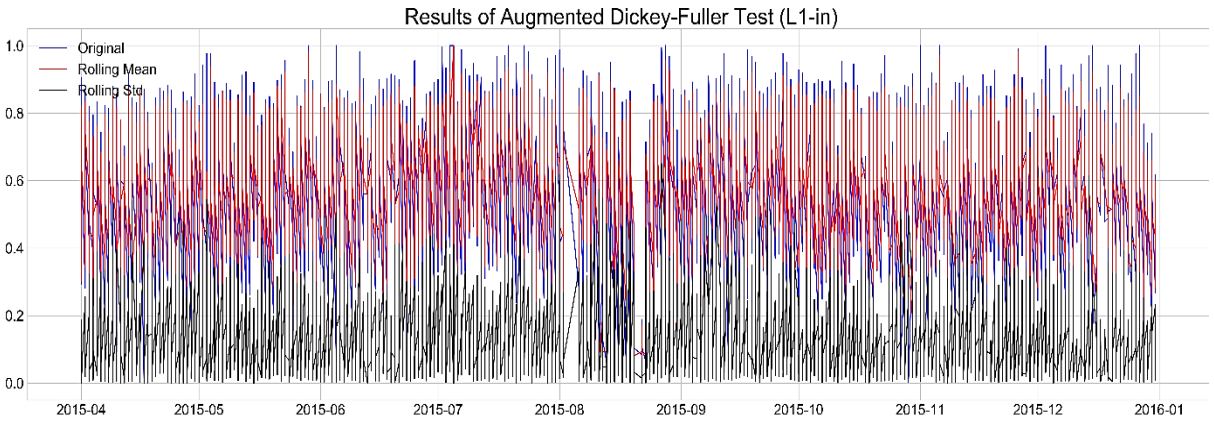
Figure 5.9 shows the result of the ADF statistics test on the on $L1_{in}$ flow series. The ADF test is also called unit root test. It tells how much a trendiness in the time series is there. ADF uses an auto regressive model. The null hypothesis set is that the time series can be represented by a unit root i.e. the series in non-stationary. The alternate hypothesis being that the series is stationary. By comparing the ADF test statistic to critical values we can accept or reject the hypothesis by comparing how much both the result values differ each other. Figure 5.9 shows that the test statistic value of -8.62 is much less than all the three critical thresholder values of the test suggesting that the flow series have no unit root, rejecting the null hypothesis and that the time series is stationary and does not have time-dependent structure but instead have a seasonality component so it can be classified as seasonal stationary and not a strict stationary flow series. The same test was performed for all the link flows and they all rejected the null hypothesis and exhibited a strong seasonal stationarity behaviour. Also, to double check the p value is less than 0.05, which affirms our intuition that the series is in fact stationary. And so no further steps are needed to perform to make the flow data series strict stationary.

5.1 Experimental Environment

Experimental setup has been deployed on a single personal running laptop which made it easier for off campus working with continued development as certain top end deep machine learning algorithms took longer the expected for the best parameter estimation and for the ML models to be trained with them.

Personal Machine Specifications:

Laptop from Hewlett-Packard machine running Windows 10 Education 64 bit. With 8 cores Intel(R) Core (TM) i7-3630QM CPU @ 2.40GHz with a total of 8 processor threads, 8 GB RAM and Intel(R) HD Graphics 4000.



#Lags Used	ADF Test Statistic	Critical Value (1%)	Critical Value (10%)	Critical Value (5%)	Number of Observations Use	p-value
28.0	-8.623164748844072	-3.4323805665026175	-2.5672475896829807	-2.8624371819849372	3223.0	6.028169014699766e-14

Figure 5. 9 Stationary Test: Augmented Dickey Fuller Test Results.

5.2 Experimental Results

In this section the experimental results are presented. The experiment results in this section comprise of the comparison of mean absolute error (MAE) and root mean square error (RMSE) results. The MAE and RMSE are both calculated for the training and the test data respectively and overall for the links data. Further we discuss different predictions cases as defined in section 5.2. More detailed explanation of the performance measures mentioned in both the cases is presented in the conclusions section.

5.2.1 Case 1: Experiment with Different Prediction Intervals

In this case prediction performances of ML models for three different prediction horizons are compared. Table 5.1 shows the short-term forecasting horizon results exhibited by the models. In table 5.2 the medium-term forecasting results are presented and lastly long-term forecasting horizon results are found in table 5.3. All the three tables are coloured based on colour gradient for the MAE and RMSE as the error measure for the models, further the minimum values measured are highlighted with the decreasing colours. The final MAE and RMSE results are aggregated and averaged over each time step in flow feature only scenario.

Method	HA	SARIMA	RFR	SVR	ANN	DBN	CNN	LSTM	LSTM-ANN	DCNN-LSTM
MAE	0.177935	0.19216	0.137087	0.140051	0.137642	0.16688	0.138813	0.138013	0.139437	0.153756
RMSE	0.227545	0.230162	0.178845	0.179993	0.180963	0.207613	0.180102	0.180266	0.182798	0.192985

Table 5. 1 MAE and RMSE Results for The Short-Term Prediction Horizon.

Method	HA	SARIMA	RFR	SVR	ANN	DBN	CNN	LSTM	LSTM-ANN	DCNN-LSTM
MAE	0.185212	0.192189	0.137378	0.149899	0.137187	0.166353	0.138336	0.139925	0.138566	0.146414
RMSE	0.233728	0.230181	0.179056	0.195172	0.180811	0.206379	0.181268	0.181731	0.182525	0.193073

Table 5. 2 MAE and RMSE Results for The Medium-Term Prediction Horizon.

Method	HA	SARIMA	RFR	SVR	ANN	DBN	CNN	LSTM	LSTM-ANN	DCNN-LSTM
MAE	0.186344	0.192229	0.128377	0.131684	0.135098	0.167058	0.135628	0.139026	0.130534	0.140997
RMSE	0.233046	0.230215	0.172138	0.174138	0.178084	0.207976	0.183387	0.18011	0.174358	0.187741

Table 5. 3 MAE and RMSE Results for The Long-Term Prediction Horizon.

5.2.2 Case 2: Experiment with Inclusion of the Related Variables

As discussed in section 5.2, time of the day is one of the potential variables that could be used as an additional feature variable along with other flow variables in a multi variate machine learning models. In section 5.3.3, traffic flow versus time of the day relation is analysed, strengthening the popular belief that times of the day, day of the week does influence the traffic flows and traffic volumes. To better understand how well the deep learning top end models, perform with the added extra features, the results of these selected models (LSTM-ANN, DCNN-LSTM) along with their results from no additional features used are compared in tables 5.4 & 5.5 & 5.6. Model names highlighted with * are using the proposed objective function including extra features. The same gradient colour scheme follows in all the table, as already used in the previous performance tables. In this case the dataset is prepared following our proposed mythology for multi-link, and time dependent flow optimisation using deep learning techniques as discussed in section 4.7. The Results of case two are discussed further in section 6.2

Method	LSTM-ANN	LSTM-ANN*	DCNN-LSTM	DCNN-LSTM*
MAE	0.139437	0.113907	0.153756	0.132825
RMSE	0.182798	0.169279	0.192985	0.185119

Table 5. 4 MAE and RMSE aggregated Results of The Short-Term Prediction Horizon for The Multi Feature Inclusion.

Method	LSTM-ANN	LSTM-ANN*	DCNN-LSTM	DCNN-LSTM*
MAE	0.138566	0.133835	0.146414	0.151087
RMSE	0.182525	0.198325	0.193073	0.20392

Table 5. 5 MAE and RMSE aggregated Results of The Medium-Term Prediction Horizon for The Multi Feature Inclusion.

Method	LSTM-ANN	LSTM-ANN*	DCNN-LSTM	DCNN-LSTM*
MAE	0.130534	0.177887	0.140997	0.15457
RMSE	0.174358	0.234275	0.187741	0.205643

Table 5. 6 MAE and RMSE aggregated Results of The Long-Term Prediction Horizon for The Multi Feature Inclusion.

5.3 Summary

In this chapter, the experiments done to carry out the simulation of the models are discussed. With the initial correlation analysis of the traffic flow data and the detailed dataset breakdown it was discussed in detail. In the end the experimental results for different scenarios are presented. These results are further discussed in the evaluation and conclusion chapter 7.

6. Evaluation and Conclusion

This chapter presents the evaluation of the experimental results performed in section 5.2. Further the performance results are evaluated in section 6.1 with further discussions on results is presented in section 6.2. At the end, conclusion of this study is presented in section 6.2.

6.1 Evaluation

In this section, the forecasting results of the performed experiments are evaluated. Each case of the experiments is evaluated. Section 6.1.1 explains the performance measure for the case of just considering the flow variables for three different prediction horizons: short, medium- and long-term predictions. Whereas in section 6.1.2 prediction performances from the extended link flow variables based on time dependent proposed flow optimisation function are discussed in detail.

6.1.1 Case 1: Evaluation of Experiment Results with Different Prediction Intervals

Different prediction horizons were considered during experimentation. The results are compared for each using the ECDF plots as mentioned in the section 5.1.3 and from the result tables 5.1, 5.2 & 5.3. fivefold validations ECDF plots are plotted on the test data for the comparison purpose. This way of comparison generates more useful insights from the single domain performance results data giving a clear indication of the overall model performance. Each presented table is presented with colour intensity changes, heatmap like technique as is used in figure 5.3 which reflects the value change across the performance tables and makes it easier to differentiate the performances visually.

6.1.1.1 Short Term or Fifteen Minute Prediction Horizon

The short-term prediction performance measures are given in table 5.1. It is apparent that most of the neural based learning techniques have relatively comparable performance results for both MAE and RMSE. Although MAE is usually lower then it's RMSE counterpart in each case. This is because MAE just captures the absolute error rates and averages them across the ensemble results space whereas RMSE is the relative error measure that incorporates the relative error deviation between the results ensemble space. When compared to the baseline Historical Average (HA) model as in [99], it is apparent that all the models perform well enough to learn the data. The degree to which they generalise the data is different. As given in [100], RFR the second to baseline model also exhibits the performance comparable to deep learning models, this can be related to the sparsity in the data that made it easy for the RFR to predict for the label values. SARIMA on the other hand, due to the non-linear nature of the data is not able to generalise it well [45] , although the seasonal component was dealt well. LSTMs did exhibit a much better performance as expected due to the recurrent neural nets. LSTM-ANN and DCNN-LSTM although combining the powers of deep learning, LSTMs and ANNs did not make much of a difference in this single feature-based prediction because the correlation learning between multiple feature will be the real test for these models. DBN on the other hand in this comparison exhibited a medium to lower high-class performance.

6.1.1.2 Medium Term or Thirty Minute Prediction Horizon

The medium-term prediction results showed some mixed variations in comparison to the short-term prediction results as given in table 5.2. But clearly RFR, HA, SARIMA and SVR performances dropped quickly. While there was a slight performance improvement seen in the high-end deep learning far right models. LSTM based models showed a bit decrease in prediction performance this is because LSTM recurrent neural networks had to learn more temporal features, which in this case are more the closely related results are reported in [42]. Whereas ANN and DBN purely feed forward neural net-based networks have converged more towards the real values more than in the case of medium prediction horizon due to more spatial data. This can be attributed to the increase in the feature

engineering that the models had to do as a result of data pre-processing that had to be done to make it more suitable for the medium horizon predictions.

6.1.1.3 Long Term or Forty Minute Prediction Horizon

In the case of long prediction horizon, table 5.3 showed that the HA, SARIMA and DBN models performance results got worse than in the case of medium horizon. Whereas as all other models performances have gone up. The can be attributed to the more feature engineering done by the subsequent deep models that made them able to predict effectively at long intervals. This behaviour is apparent from the lighter colours exhibited by deep end models in results table 5.3 whereas the relative brighter colour represents larger error or more deviated performance measures.

6.1.2 Case 2: Evaluation of Experimental Results with Inclusion of the Related Variables

In this section the experimental results from section 5.2.2 with the inclusion of related time-based flow link variables utilising the proposed objective function are evaluated. Deep learning models including LSTM-ANN and DCNN-LSTM were only considered for this experiment case. The preliminary analysis in section 5.3.2 showed that traffic flows are highly correlated and the their exists a strong correlation with rest to the time of the day. Thus LSTM-ANN and DCNN-LSTM explored the spatial-temporal features for better prediction accuracy under adverse circumstances when other shallow models failed. The ECDF score plots for the case scenarios for the short, medium and long term are given in figures 6.1-6.3, respectively. Whereas, the ECDF plots with multi feature learning are given in figures 6.4-6.6.

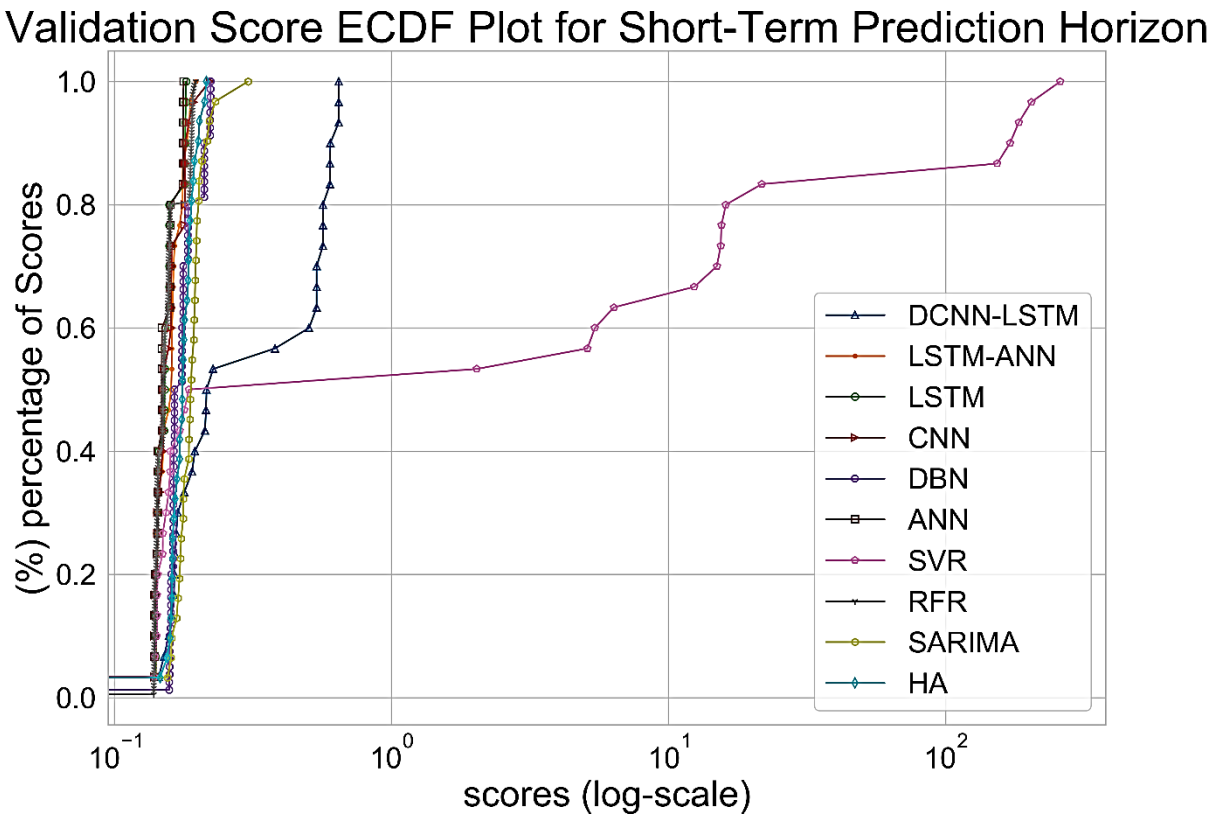


Figure 6. 1 Empirical CDF Plot of Absolute Mean Square Error Score on the Short-Term Prediction Results.

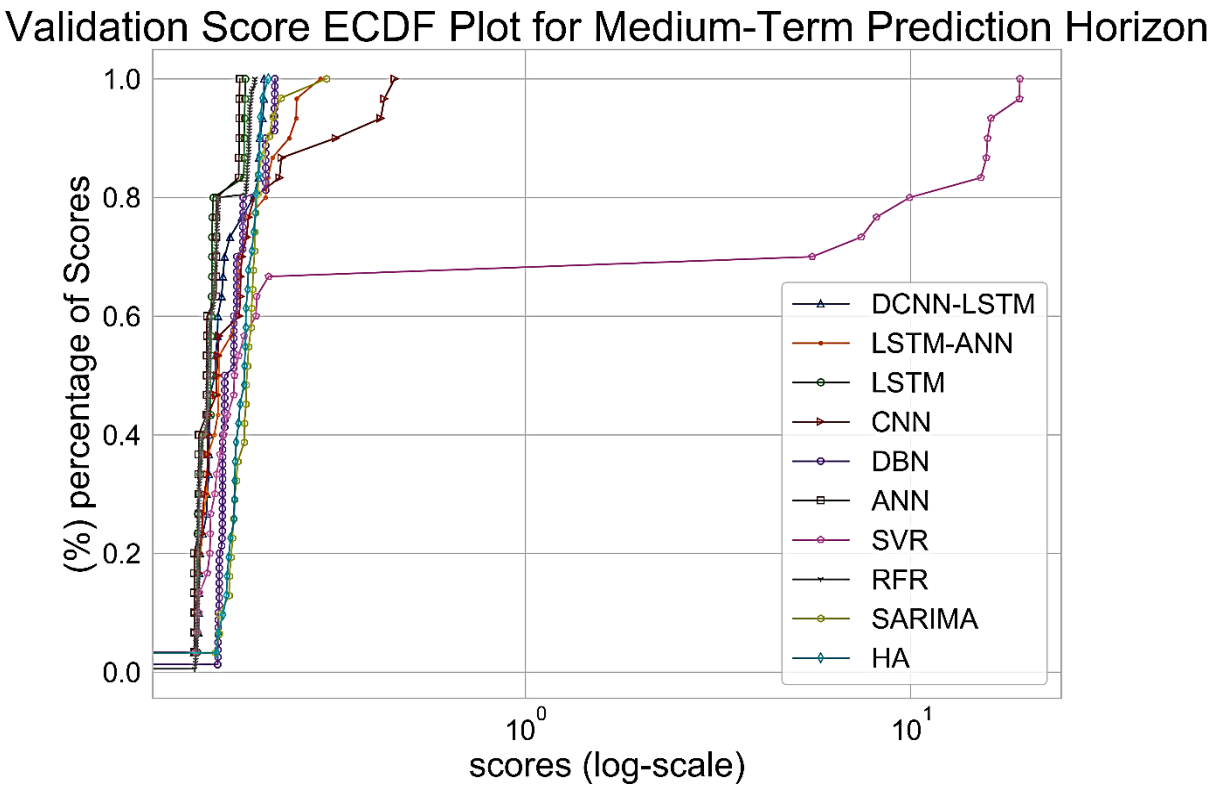


Figure 6. 2 Empirical CDF Plot of Absolute Mean Square Error Score on the Medium-Term Prediction Results.

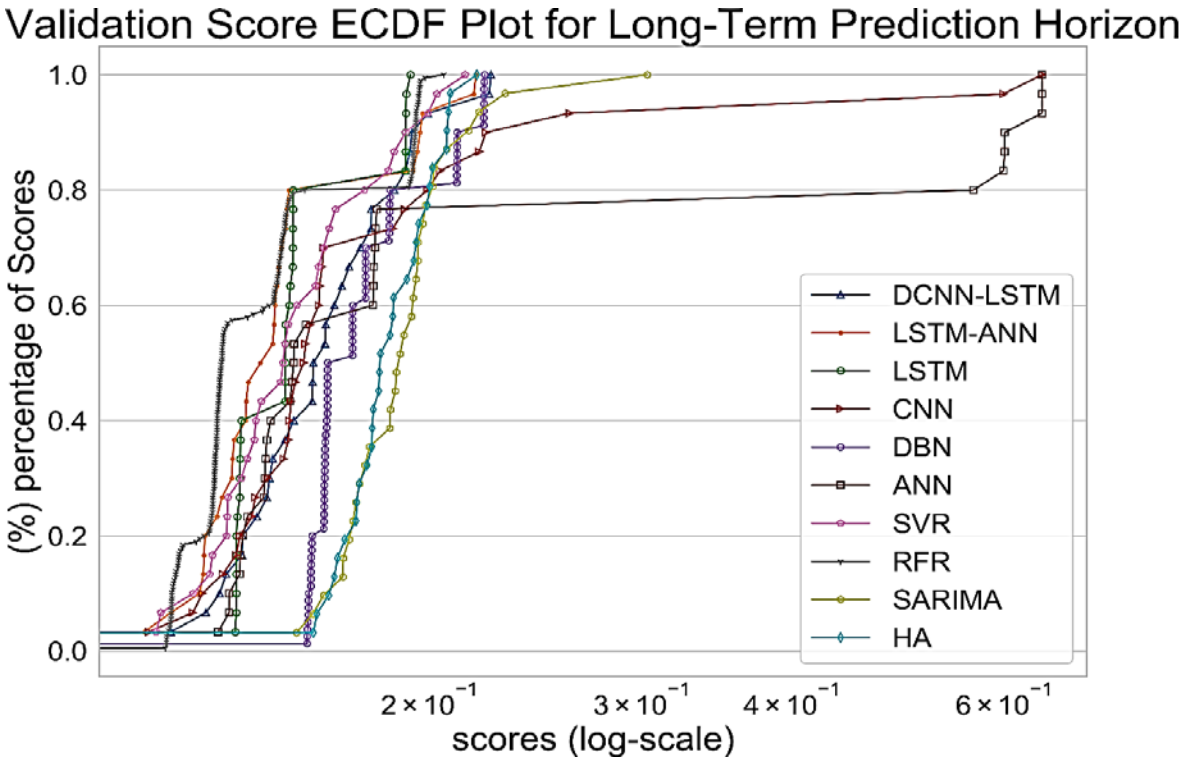


Figure 6. 3 Empirical CDF Plot of Absolute Mean Square Error Score on the Long-Term Prediction Results.

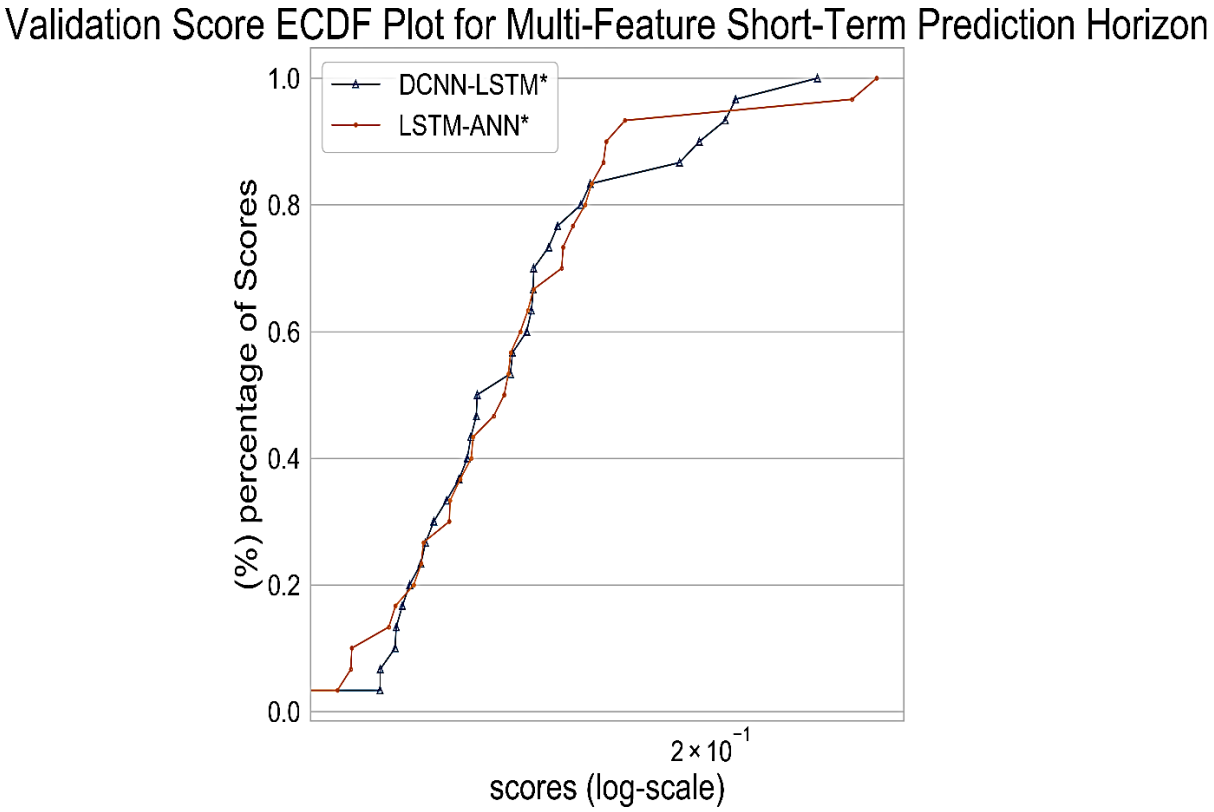


Figure 6. 4 Empirical CDF Plot of Absolute Mean Square Error Score on the Short-Term Prediction Results with Multi Link Proposed Flow Learning.

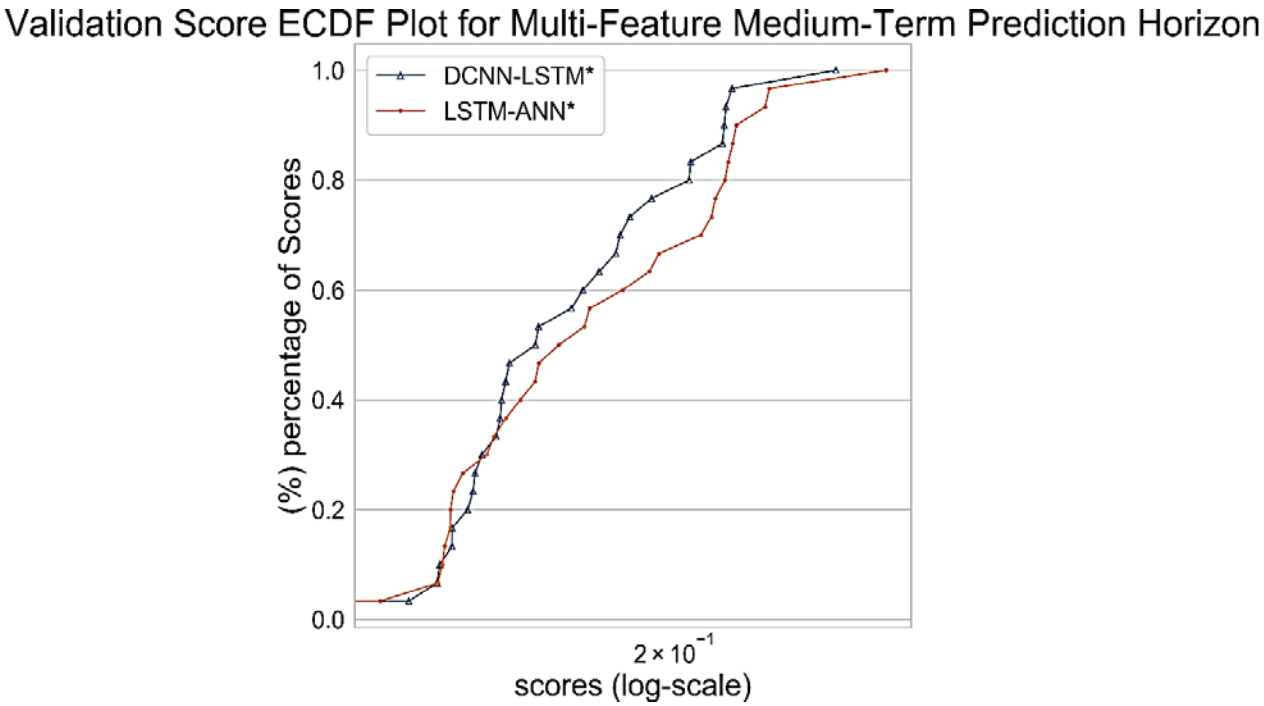


Figure 6. 5 Empirical CDF Plot of Absolute Mean Square Error Score on the Medium-Term Prediction Results with Multi Link Proposed Flow Learning.

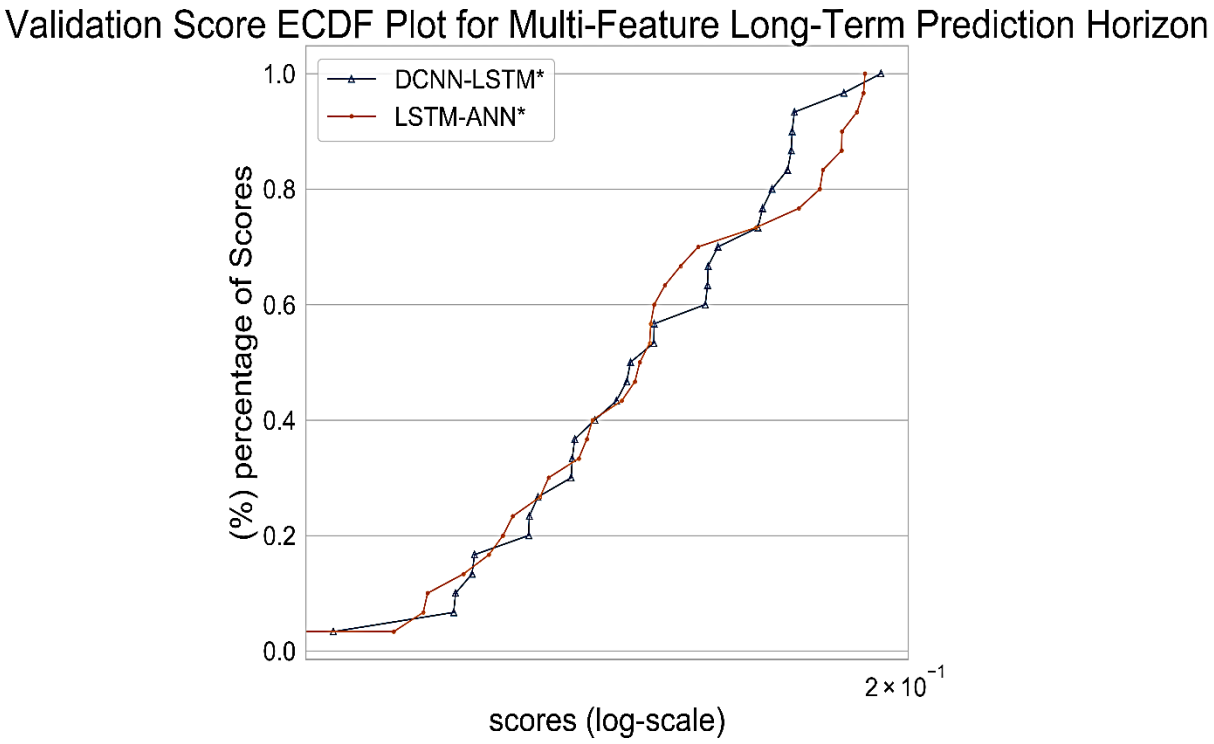


Figure 6. 6 Empirical CDF Plot of Absolute Mean Square Error Score on the Long-Term Prediction Results with Multi Link Proposed Flow Learning.

6.2 Discussion

Consistently from figures 6.1 & 6.2 it was found that SVR gave best results in both short- and medium-term predictions. The ECDF was calculated on the k-fold validation results scores where the error, considered was the mean absolute error. The ECDF was plotted to get the better understanding of the ECDF on the individual model performances. Whereas further detailed model mean scores are given in appendix A for each individual model with respect to their hyperparameter grid search. The reason for SVR exhibiting best performance in these two cases is due to the regression mechanism which leaded to classify the regressed nodes easily when they're no other related features being considered as been also reported in [47]. Which is apparent as in later experiments i.e. figure 6.3-6.6, when the feature data was increased then SVR struggled to keep up with the deep learning models.

From the ECDFs, it was clear that models that deep learning cannot just learn the time series data but also predict with relative ease and less error than the statistical techniques were the neural based models and the specific RNN based LSTM exhibited its superior performance than simple feed forward neural network.

6.2.1 Different Traffic Flow Profiles or Conditions

Although a bit similar validation performance on test data was revealed by most of the models. But when the flow rate increases or there is a sudden anomaly in the traffic, LSTM based techniques performed better in multi-feature case. The plots in figure 6.1-6.3 did show error gap between the considered baseline HA and advanced models (CNN, LSTM etc) reflecting the behavioural learning of advanced models. Although the flow data was normalised, but the flow was pretty much normal. But the models performance can be differentiated more with different flow rate profiles or conditioning

the learning and testing of advanced models to further rule out the redundant models. So different morning, evening or day condition can be considered specifically for training or class balancing in terms for the data pre-processing.

By extending the feature vector with the proposed link objective function, the advanced deep models performances showed an improvement as can be seen in figure 6.4-6.6. This is because with each horizon increase as part of lag, during supervised pre-processing one more lagged feature was considered which brought about the stability and more learning capability into these models. So, the LSTM-ANN* model performances are improved while the DCNN-LSTM* performance accuracy went down for long term predictions, but the models can be further compared by considering other learned variables before having a final say. This can be attributed to the fact that the CNNs are unable to preserve the pattern integrity in the original data in a bid to generate feature vectors through convolutional layers but LSTMs on the other hand did a good job by forgetting what is not likely the possible outcome using their forget and output gates. Such mechanism is missing in DCNNs. Almost similar experimental observations are reported in [36].

6.2.1 Limitations

6.2.1.1 Common Pre-Processing Assumptions

It's worth mentioning some of the limits and limitations of this thesis. Firstly, the original collected raw dataset timeline spanned over for eight months. To prepare the raw dataset multiple complex individual links data had to be manipulated using pre-assumptions made to compile the data for further pre-processing. Some of the assumptions made; included the parallel highway road links flow data aggregations and anti-parallel flow link subtraction while applying the basic concepts of fluid flow.

6.2.1.2 Lack of Availability of Common ITS Data Across Literature

Secondly, the training and test data did not have the seasonal changes for the whole one year. The trained model seen mostly spring, summer and pre-winter time data. Which didn't generalise the overall annual behaviour of traffic. The dataset can be made more generalised as part of the future works. Unfortunately, the models presented in this research are not tested for any literature research mentioned dataset as the details mentioned were missing to reproduce the dataset or the experimental environment. A common dataset for comparative research elements is the need of the current ITS.

6.2.1.3 Hyperparameter Tuning Mechanism for Each Individual Model

Although the search for the best hyper parameter had been carried out pre-training. But due to the time constraints a much broader search can be carried out by increasing the domains range for the hyper parameters search inputs. The best performing hyper parameters for each model are given in appendix A. The final model was trained using hyper parameters searched based on the models best scores.

6.3 Conclusions

The research questions raised in section 1.2 are addressed in this section:

RQ1:

What are the potential hindering challenges for the practical implementation of the road traffic parameter forecasting systems?

RQ2:

What are the state-of-the-art traffic prediction machine learning architectures for traffic flow forecasting and what effect does the proposed methodology have on the chosen model performances?

Research questions one and two (RQ1 & RQ2) are answered by the detailed literature review in section 2.3 and 2.4. The main hinderance found through the literature review was that different prediction models considered different traffic datasets and there were no common datasets across the literature, which was one of the key issues. Ideally the model performance merits must be judged utilising the common datasets. As it becomes difficult to address which one is the state-of-the-art model due to the dynamic nature of the gathered traffic networks data. Through the recent advancement in deep learning algorithms they have defined the new limits for the state of the art, which indeed in true in the scope of this thesis as well and has been proved by the experimental results in this thesis.

Deep Learning Networks that evolved from different neural network-based forecasting models have been extensively studied in the literature [44]. They have been integrated into deep belief networks (DBNs) and later into Convolutional networks (CNNs) with much success. But currently the focus of researchers is mostly on the deep learning and hybrid data riven models. i.e. CNN-LSTMs DCNN-LSTMs.

RQ3:

What are the state-of-the-art traffic prediction machine learning architectures for traffic flow forecasting?

From the latest literature review the state-of-the-art deep machine learning techniques are now being freshly considered in the field of ITS. The freshly proposed techniques in this thesis utilise a modular approach on a road junction level which employs good features of a model to tackle the dynamic nature of the data. This gives rise to the researchers being proposing an amalgam of hybrid data driven techniques that mostly centre around the RNNs and ANNs.

RQ4:

What deep machine learning approaches have to offer when compared to conventional or shallow machine learning techniques considering the traffic flow data?

Deep learning techniques have the added advantage of adaptability and continuous model training which makes them a favourable candidate for the big data problems. Where shallow machine learning techniques like SVR and RFR limits themselves as in this thesis, deep learning models takes the charge. In ITS researchers are mostly focussing on spatial-temporal transport data. Which for deep learning models is handed by different parts i.e. LSTMs handles the temporal data learning and ANNs or CNNs handles the spatial based data.

The bi-directional flow function of individual roads is reported considering the net inflows and outflows by a topological breakdown of the highway network. Further, the proposed objective function is optimised and compared for constraints involved using statistical and neural based machine learning models considering different loss functions and training optimisation strategies. Finally, we report the best fitting machine learning model parameters for the proposed flow objective function for better prediction accuracy. The deep learning models are also tested in a separate experiment case for the features that are time dependent in the experiments. Although every flow time series is time dependent but the combination of how the input data is fed to the models with respect to the time does matter because the models exploits for the features that they see, which for

the proposed methodology in this thesis, was partly incorporated as part of the data pre-processing phase.

The driving force of deep modular learning models is the hyperparameter tuning of each individual model which took a lot of the authors time for the experimentation as well. But this is the key for the making the best bets out of deep ML model. Without this the shallow ML models might perform better than the deep learning techniques.

Conclusively, the results from experiments exhibit that shallow machine learning techniques can be used if the data is sparse enough to be categorically predicted like in the case of SVR and RFR and if not then the patterns in data needs to be learned properly using FFBNN and LSTM based deep learning techniques, since the later performed better in highly correlated sparse data conditions . Also, the proposed network breakdown for machine learning implementation does influence the performance of final model which in our experiments improved than those of with no objective function to consider traffic network flow links [44].

6.4 Contributions

There are two main contributions that are made in this thesis.

6.4.1 Thorough ITS Literature Study on Prediction Models

The literature review is presented in a very structured form starting right from discussing in details the popular statistical forecasting approaches to shallow machine learning techniques and hybrid data driven deep learning models. This work has compared the FFBN, RFR, SVR, RNN based models, CNN and CNN-RNN hybrid models for the ITS traffic flow predictions utilising the real data in UK.

6.4.2 Junction level Proposed Flow Prediction Objective Function

The second contribution is the proposed topological junction based modular road traffic networks break down for spatial-temporal data exploitation by the models and to predict the flow in and out of the road links. To the authors knowledge, this is the first time that such comprehensive study for ITS especially for traffic flow forecasting have been carried out.

6.5 Future Works

This thesis explains that the deep learning-based methods can be applied to the traffic flow data from the Highway England (HE). As the MIDAS dataset (refer section 4.4.1) have variously gathered traffic parameters which can be used in conjunction with the flow features. This incorporation of new feature vectors (i.e. average lane speeds fused, local weather conditions etc) can greatly lift the performance of the deep learning models and can be deployed as further reliable Realtime traffic predictions systems for the public. This would further modify the objective functions and would make them more elaborative a complete representation of the network path (refer Appendix C), which would lead to congestion emerging bottleneck points identification and their effect on individual links flow forecasting. The trained model performances can be further subjected to different flow conditions which will lead to more insights as to how the models will perform under varying traffic conditions. These forecasting techniques would help the public and transport providers to adopt the safety measures before the event is about to happen. Further detailed future works that could be adopted are given in appendix C.

References

[1] P. Domingos, “A few useful things to know about machine learning,” *Commun. ACM*, vol. 55, no. 10, p. 78, 2012.

[2] R. Gupta and C. Pathak, “A machine learning framework for predicting purchase by online customers based on dynamic pricing,” *Procedia Comput. Sci.*, vol. 36, no. C, pp. 599–605, 2014.

[3] R. C. Staudemeyer and C. W. Omlin, “Extracting salient features for network intrusion detection using machine learning methods,” *South African Comput. J.*, vol. 52, no. July, pp. 82–96, 2014.

[4] M. Rabbani, R. Khoshkangini, H. S. Nagendraswamy, and M. Conti, “Hand Drawn Optical Circuit Recognition,” *Procedia Comput. Sci.*, vol. 84, pp. 41–48, 2016.

[5] B. van Riessen, R. R. Negenborn, and R. Dekker, “Real-time container transport planning with decision trees based on offline obtained optimal solutions,” *Decis. Support Syst.*, vol. 89, pp. 1–16, 2016.

[6] A. Verikas, A. Gelzinis, and M. Bacauskiene, “Mining data with random forests: A survey and results of new tests,” *Pattern Recognit.*, vol. 44, no. 2, pp. 330–349, 2011.

[7] M. Schuh, J. Sheppard, S. Strasser, R. Angryk, and C. Izurieta, “An IEEE standards-based visualization tool for knowledge discovery in maintenance event sequences,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 7, pp. 30–39, 2013.

[8] A. S. Ahmad *et al.*, “A review on applications of ANN and SVM for building electrical energy consumption forecasting,” *Renew. Sustain. Energy Rev.*, vol. 33, pp. 102–109, 2014.

[9] A. Anwar, T. Nagel, and C. Ratti, “Traffic origins: A simple visualization technique to support traffic incident analysis,” *IEEE Pacific Vis. Symp.*, pp. 316–319, Mar. 2014.

[10] J. W. C. van Lint, “Reliable Travel Time Prediction for Freeways,” te Delft, 2004.

[11] A. Abadi, T. Rajabioun, and P. A. Ioannou, “Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, 2015.

[12] C. Hsu and F. Lian, “A Case Study on Highway Flow Model Using 2-D Gaussian Mixture Modeling,” in *Proceedings of the 2007 IEEE Intelligent Transportaion Systems Conference*, 2007, pp. 790–794.

[13] S. Oh, Y. J. Byon, K. Jang, and H. Yeo, “Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach,” *Transp. Rev.*, vol. 35, no. 1, pp. 4–32, 2015.

[14] C. Goves, R. North, R. Johnston, and G. Fletcher, “Short Term Traffic Prediction on the UK Motorway Network Using Neural Networks,” *Transp. Res. Procedia*, vol. 13, pp. 184–195, 2016.

[15] K. Kumar, M. Parida, and V. K. Katiyar, “Short term traffic flow prediction in heterogeneous condition using artificial neural network,” *Transport*, vol. 30, no. 4, pp. 397–405, 2015.

[16] Z. Abdelhafid, F. Harrou, and Y. Sun, “An Efficient Statistical-based Approach for Road Traffic Congestion Monitoring,” in *5th Int. Conf. Electr. Eng. - Boumerdes*, 2017, vol. 2017–Janua, pp. 1–5.

[17] R. Li and G. Rose, “Incorporating uncertainty into short-term travel time predictions,” *Transp. Res. Part C Emerg. Technol.*, vol. 19, no. 6, pp. 1006–1018, 2011.

[18] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Short-term traffic forecasting : Where we are and where we ’ re going,” *Transp. Res. Part C*, vol. 43, pp. 3–19, 2014.

[19] C. Siripanpornchana, S. Panichpapiboon, and P. Chaovalit, “Effective variables for urban traffic incident detection,” *IEEE Veh. Netw. Conf. VNC*, vol. 2016–Janua, pp. 190–195, Dec. 2016.

[20] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, “A compressive sensing approach to urban traffic estimation with probe vehicles,” *IEEE Trans. Mob. Comput.*, vol. 12, no. 11, pp. 2289–2302, 2013.

[21] Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bajgain, “Improved Deep Hybrid Networks for Urban Traffic Flow Prediction Using Trajectory Data,” *IEEE Access*, vol. 6, pp. 31820–31827, 2018.

[22] G. Fusco, C. Colombaroni, and N. Isaenko, “Short-term speed predictions exploiting big data on large urban road networks,” *Transp. Res. Part C Emerg. Technol.*, vol. 73, pp. 183–201, 2016.

[23] F. Schimbinschi, L. Moreira-Matias, V. X. Nguyen, and J. Bailey, “Topology-regularized universal vector autoregression for traffic forecasting in large urban areas,” *Expert Syst. Appl.*, vol. 82, pp. 301–316, Oct. 2017.

[24] F. Su, H. Dong, L. Jia, Y. Qin, and Z. Tian, “Long-term forecasting oriented to urban expressway traffic situation,” *Adv. Mech. Eng.*, vol. 8, no. 1, pp. 1–16, 2016.

[25] S. Oh, Y. Kim, and J. Hong, “Urban Traffic Flow Prediction System Using a Multifactor Pattern Recognition Model,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2744–2755, 2015.

[26] Z. Yuan and C. Tu, “Short-term Traffic Flow Forecasting Based on Feature Selection with Mutual Information,” in *Materials Science, Energy Technology, and Power Engineering I AIP Conf. Proc.*, 2017, vol. 020179, no. 1, pp. 1–9.

[27] A. Zeroual, N. Messai, S. Kechida, and F. Hamdi, “A piecewise switched linear approach for traffic flow modeling,” *Int. J. Autom. Comput.*, vol. 14, no. 6, pp. 729–741, 2017.

[28] Q. Li, S. Li, and Y. Wang, “Traffic incident data analysis and performance measures development,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. 086, pp. 65–69, 2007.

[29] J. Wang, X. Li, S. S. Liao, and Z. Hua, “A Hybrid Approach for Automatic Incident Detection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1176–1185, 2013.

[30] R. Kalsoom and Z. Halim, “Clustering The Driving Features Based On Data Streams,” *IEEE*, pp. 89–94, Dec. 2013.

[31] H. Nguyen, C. Cai, and F. Chen, “Automatic classification of traffic incident’s severity using machine learning approaches,” *IET Intell. Transp. Syst.*, vol. 11, no. 10, pp. 615–623, Dec. 2017.

[32] C. E. L. Hatri and J. Boumhidi, “Fuzzy deep learning based urban traffic incident detection,” *2017 Intell. Syst. Comput. Vis.*, pp. 1–6, Apr. 2017.

[33] J. Guo, Z. Liu, W. Huang, Y. Wei, and J. Cao, “Short-term traffic flow prediction using fuzzy information granulation approach under different time intervals,” *IET Intell. Transp. Syst.*, vol. 12, no. 2, pp. 143–150, 2018.

[34] M. M. Rahman, S. C. Wirasinghe, and L. Kattan, “Analysis of bus travel time distributions for varying horizons and real-time applications,” *Transp. Res. Part C Emerg. Technol.*, vol. 86, no. December 2017, pp. 453–466, 2018.

[35] R. Fernandez and R. Planzer, “On the capacity of bus transit systems,” *Transp. Rev.*, vol. 22, no. 3, pp. 267–293, 2002.

[36] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, “Traffic Flow Prediction with Big Data: A Deep Learning Approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2015.

[37] X. Ma, H. Yu, Y. Wang, and Y. Wang, “Large-scale transportation network congestion evolution prediction using deep learning theory,” *PLoS One*, vol. 10, no. 3, 2015.

[38] J. Y. Ahn, E. Ko, and E. Kim, “Predicting Spatiotemporal Traffic Flow Based on Support Vector Regression and Bayesian Classifier,” *2015 IEEE Fifth Int. Conf. Big Data Cloud Comput.*, pp. 125–130, 2015.

[39] X. Ma, Z. Dai, Z. He, J. Ma, Y. Y. Wang, and Y. Y. Wang, “Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction,” *Sensors (Switzerland)*, vol. 17, no. 4, p. 818, Apr. 2017.

[40] R. Al Mallah, A. Quintero, and B. Farooq, “Distributed Classification of Urban Congestion Using VANET,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2435–2442, Sep. 2017.

[41] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, “Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3204–3217, 2017.

[42] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” *Proc. - 2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2016*, pp. 324–328, 2017.

[43] G. Yang, Y. Wang, H. Yu, Y. Ren, and J. Xie, “Short-term traffic state prediction based on the spatiotemporal features of critical road sections,” *Sensors (Switzerland)*, vol. 18, no. 7, 2018.

[44] X. Cheng, R. Zhang, J. Zhou, and W. Xu, “DeepTransport: Learning Spatial-Temporal Dependency for Traffic Condition Forecasting,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018–July, pp. 1–8, 2018.

[45] S. V. Kumar and L. Vanajakshi, “Short-term traffic flow prediction using seasonal ARIMA model with limited input data,” *Eur. Transp. Res. Rev.*, vol. 7, no. 3, pp. 1–9, 2015.

[46] J. Guo, W. Huang, and B. M. Williams, “Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 50–64, 2014.

[47] M. T. Asif *et al.*, “Spatiotemporal patterns in large-scale traffic speed prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, 2014.

[48] J. Xin and S. Chen, “Bus Dwell Time Prediction Based on KNN,” *Procedia Eng.*, vol. 137, pp. 283–288, 2016.

[49] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, “A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting,” *Transp. Res. Part C Emerg. Technol.*, vol. 62, pp. 21–34, 2016.

[50] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, “A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting,” *Neurocomputing*, vol. 179, pp. 246–26, 2016.

[51] J. Amita, S. S. Jain, and P. K. Garg, “Prediction of Bus Travel Time Using ANN: A Case Study in Delhi,” *Transp. Res. Procedia*, vol. 17, no. December 2014, pp. 263–272, 2016.

[52] X. Ma, Z. Tao, Y. Y. Wang, H. Yu, and Y. Y. Wang, “Long short-term memory neural network for traffic speed prediction using remote microwave sensor data,” *Transp. Res. Part C Emerg. Technol.*, vol. 54, pp. 187–197, 2015.

[53] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” *Sensors (Switzerland)*, vol. 17, no. 7, pp. 1–16, 2017.

[54] C.-M. Hsu, F.-L. Lian, and C.-M. Huang, “A Systematic Spatiotemporal Modeling Framework for Characterizing Traffic Dynamics Using Hierarchical Gaussian Mixture Modeling and Entropy Analysis,” *IEEE Syst. J.*, vol. 8, no. 4, pp. 1126–1135, 2014.

[55] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, “Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting,” *Proc. 2017 SIAM Int. Conf. Data Min.*, pp. 777–785, 2017.

[56] W. Fan and R. B. Machemehl, *Characterizing Bus Transit Passenger Waiting Times*, vol. SWUTC/99/1, no. 1. 1999.

[57] R. Fernández, "Modelling public transport stops by microscopic simulation," *Transp. Res. Part C Emerg. Technol.*, vol. 18, no. 6, pp. 856–868, 2010.

[58] National Research Council (U.S.) *et al.*, "Guidelines for the design and location of Bus Stops," *Transit Coop. Res. Progr.*, 1994.

[59] J. B. D.B.Hess, "Waitingfor the bus," *J. Public Transp.*, vol. 7, no. 4, pp. 67–84, 2004.

[60] P. G. Furth and T. H. J. Muller, "Service Reliability and Hidden Waiting Time: Insights from Automatic Vehicle Location Data," *Transp. Res. Board*, vol. 1955, 2006.

[61] F. McLeod, "Estimating bus passenger waiting times from incomplete bus arrivals data," *J. Oper. Res. Soc.*, vol. 58, no. 11, pp. 1518–1525, 2007.

[62] N. E. Myridis, "Probability, Random Processes, and Statistical Analysis, by H. Kobayashi, B.L. Mark and W. Turin," *Contemp. Phys.*, vol. 53, no. 6, pp. 533–534, Nov. 2012.

[63] O. C. Ibe, O. A. Isijola, O. A. Isijola-Adakeja, and O. C. Ibe, "M/M/1 multiple vacation queueing systems with differentiated vacations and vacation interruptions," *IEEE Access*, vol. 2, pp. 1384–1395, 2014.

[64] G. Xin and W. Wang, "Model Passengers' Travel Time for Conventional Bus Stop," *J. Appl. Math.*, vol. 2014, pp. 1–9, Apr. 2014.

[65] D. A. Wu and H. Takagi, "M/G/1 queue with multiple working vacations," *Perform. Eval.*, vol. 63, no. 7, pp. 654–681, Jul. 2006.

[66] H. Yu, Z. Wu, D. Chen, and X. Ma, "Probabilistic Prediction of Bus Headway Using Relevance Vector Machine Regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1772–1781, Jul. 2017.

[67] Z. Yu, J. S. Wood, and V. V. Gayah, "Using survival models to estimate bus travel times and associated uncertainties," *Transp. Res. Part C Emerg. Technol.*, vol. 74, pp. 366–382, 2017.

[68] H. Yu, D. Chen, Z. Wu, X. Ma, and Y. Wang, "Headway-based bus bunching prediction using transit smart card data," *Transp. Res. Part C Emerg. Technol.*, vol. 72, pp. 45–59, 2016.

[69] B. A. Kumar, L. Vanajakshi, and S. C. Subramanian, "Bus travel time prediction using a time-space discretization approach," *Transp. Res. Part C Emerg. Technol.*, vol. 79, pp. 308–332, 2017.

[70] M. Meng, A. Rau, and H. Mahardhika, "Public transport travel time perception: Effects of socioeconomic characteristics, trip characteristics and facility usage," *Transp. Res. Part A Policy Pract.*, no. xxxx, pp. 0–1, 2018.

[71] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich, "Traveling time prediction in scheduled transportation with journey segments," *Inf. Syst.*, vol. 64, pp. 266–280, 2017.

[72] A. Comi, A. Nuzzolo, S. Brinchi, and R. Verghini, "Bus travel time variability: Some experimental evidences," *Transp. Res. Procedia*, vol. 27, pp. 101–108, 2017.

[73] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '14*, no. 5, pp. 25–34, 2014.

[74] B. Yang, C. Guo, and C. S. Jensen, "Travel cost inference from sparse, spatio-temporally correlated time series using markov models," *Proc. VLDB Endow.*, vol. 6, no. 9, pp. 769–780, 2013.

[75] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik, "Dynamic route planning with real-time traffic predictions," *Inf. Syst.*, vol. 64, pp. 258–265, 2017.

[76] L. Gasparini, E. Bouillet, F. Calabrese, O. Verscheure, B. O'Brien, and M. O'Donnell, "System and

analytics for continuously assessing transport systems from sparse and noisy observations: Case study in Dublin," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. April 2015, pp. 1827–1832, 2011.

[77] B. Sun *et al.*, "An improved k-nearest neighbours method for traffic time series imputation," *©IEEE CAC 2017*, vol. 10, no. October, pp. 7346–7351, 2017.

[78] M. Moniruzzaman, H. Maoh, and W. Anderson, "Short-term prediction of border crossing time and traffic volume for commercial trucks: A case study for the Ambassador Bridge," *Transp. Res. Part C Emerg. Technol.*, vol. 63, pp. 182–194, 2016.

[79] Y. Duan *et al.*, "An efficient realization of deep learning for traffic data imputation," *Transp. Res. Part C Emerg. Technol.*, vol. 72, no. 10, pp. 168–181, 2016.

[80] O. D. Cardozo, J. C. García-Palomares, and J. Gutiérrez, "Application of geographically weighted regression to the direct forecasting of transit ridership at station-level," *Appl. Geogr.*, vol. 34, no. 4, pp. 548–558, 2012.

[81] Q. V. Le I.lSutskever, OV.inyals, "Sequence to Sequence Learning with Neural Networks," in *Neural Information Processing Systems Conference*, 2016, pp. 1–9.

[82] L. Deng and N. Jaitly, "Deep Discriminative and Generative Models for Pattern Recognition," pp. 1–26, 2015.

[83] G. B. Zhou, J. Wu, C. L. Zhang, and Z. H. Zhou, "Minimal gated unit for recurrent neural networks," *Int. J. Autom. Comput.*, vol. 13, no. 3, pp. 226–234, 2016.

[84] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[85] K. Yin, W. Wang, X. Bruce Wang, and T. M. Adams, "Link travel time inference using entry/exit information of trips on a network," *Transp. Res. Part B Methodol.*, vol. 80, pp. 303–321, 2015.

[86] F. N. Savas, "Forecast Comparison of Models Based on SARIMA and the Kalman Filter for In ation," 2013.

[87] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting , Second Edition Springer Texts in Statistics*. 2003.

[88] V. N. Vapnik, *Statistical learning theory*. 1998.

[89] Geoffrey E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," London, 2002.

[90] S. Hochreiter and J urgen Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[91] T. M. Units, "National Traffic Information Service DATEX II Service," 2018.

[92] DfT, "Road traffic statistics," pp. 1–13, 2014.

[93] Highways England, "Highways England – Data.gov.uk – Journey Time and Traffic Flow Data April 2015 onwards – User Guide," no. April, pp. 1–14, 2015.

[94] A. Rahi and S. Ramalingam, "Empirical Formulation of Highway Traffic Flow Prediction Objective Function Based on Network Topology," *Int. J. Adv. Res. Sci. Eng. Technol.*, vol. 5, no. November, 2018.

[95] D. Zhang and M. R. Kabuka, "Combining Weather Condition Data to Predict Traffic Flow: A GRU Based Deep Learning Approach," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data*

Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), 2017, pp. 1216–1219.

[96] Y. Jia, J. Wu, and M. Xu, "Traffic flow prediction with rainfall impact using a deep learning method," *J. Adv. Transp.*, vol. 2017, 2017.

[97] M. Shardlow, "An Analysis of Feature Selection Techniques," *Studentnet.Cs.Manchester.Ac.Uk*, pp. 1–7, 2007.

[98] D. A. Dickey, *Stationarity Issues in Time Series Models*. .

[99] W. Fan and Z. Gurmu, "Dynamic Travel Time Prediction Models for Buses Using Only GPS Data," *Int. J. Transp. Sci. Technol.*, vol. 4, no. 4, pp. 353–366, 2015.

[100] Y. Liu and H. Wu, "Prediction of Road Traffic Congestion Based on Random Forest," *2017 10th Int. Symp. Comput. Intell. Des.*, pp. 361–364, 2017.

[101] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *Intell. Transp. Syst. IEEE Trans.*, vol. 7, no. 1, pp. 124–132, 2006.

[102] A. Pascale and M. Nicoli, "Adaptive Bayesian network for traffic flow prediction," *2011 IEEE Stat. Signal Process. Work.*, pp. 177–180, 2011.

[103] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification (2nd edition)," in *John Wiley & Sons, Inc*, no. 2nd ed., 2000.

[104] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 3–19, 2014.

[105] W. Feng, Wei Feng, W. Feng, and Wei Feng, "PDXScholar Analyses of Bus Travel Time Reliability and Transit Signal Priority at the Stop-To-Stop Segment Level," 2014.

[106] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," 2014.

Appendix A : Hyperparameters Tuning Results

In this appendix the search results for the hyperparameters among each individual model are presented with different forecasting horizon. Except the historical Average (HA) model every model had gone through validation curve routines and grid search for hyper parameter. The best scored model hyperparameters were used in the final training. The grid searches were performed for each prediction horizons.

A.1 Experiment Case1: Best Search Hyperparameters Used for Multi Prediction Horizons

The model best performing hyper parameter search results are presented in this appendix.

Historical Moving Average (HA):

Due to the simple working of HA technique no additional parameter was tuned except the running window size which was chosen as given in table A.1

	Window Size	**Prediction Horizon**
1.	Two (2)	Short Term
2.	Three (3)	Medium-term
3.	Four (4)	Long Term

Table A.1 Chosen Window Size for the Prediction Intervals.

Seasonal Arima (SARIMA):

Unlike other models only one grid search for SARIMA was performed (shown in figure A.1) as new observations were later added in to the model without going through the separate fitting routine for forecasting for different horizons.

```
Fit ARIMA: order=(1, 0, 1) seasonal_order=(0, 0, 0, 12); AIC=-1489.555, BIC=-1465.212, Fit time=1.997 seconds
Fit ARIMA: order=(0, 0, 0) seasonal_order=(0, 0, 0, 12); AIC=-788.802, BIC=-776.631, Fit time=0.479 seconds
Fit ARIMA: order=(1, 0, 0) seasonal_order=(1, 0, 0, 12); AIC=-2213.248, BIC=-2188.905, Fit time=4.615 seconds
Fit ARIMA: order=(0, 0, 1) seasonal_order=(0, 0, 1, 12); AIC=-2031.871, BIC=-2007.528, Fit time=2.439 seconds
Fit ARIMA: order=(1, 0, 0) seasonal_order=(0, 0, 0, 12); AIC=-1485.301, BIC=-1467.044, Fit time=0.266 seconds
Fit ARIMA: order=(1, 0, 0) seasonal_order=(2, 0, 0, 12); AIC=-2219.858, BIC=-2189.429, Fit time=23.613 seconds
Fit ARIMA: order=(1, 0, 0) seasonal_order=(2, 0, 1, 12); AIC=-2656.788, BIC=-2620.273, Fit time=32.484 seconds
Fit ARIMA: order=(1, 0, 0) seasonal_order=(3, 0, 2, 12); AIC=-2671.805, BIC=-2623.118, Fit time=124.236 seconds
Fit ARIMA: order=(0, 0, 0) seasonal_order=(3, 0, 2, 12); AIC=-2139.540, BIC=-2096.940, Fit time=103.951 seconds
Fit ARIMA: order=(2, 0, 0) seasonal_order=(3, 0, 2, 12); AIC=-2718.923, BIC=-2664.151, Fit time=117.389 seconds
Fit ARIMA: order=(2, 0, 1) seasonal_order=(3, 0, 2, 12); AIC=-2694.945, BIC=-2634.087, Fit time=142.287 seconds
Fit ARIMA: order=(3, 0, 1) seasonal_order=(3, 0, 2, 12); AIC=-2716.727, BIC=-2649.783, Fit time=184.872 seconds
Fit ARIMA: order=(2, 0, 0) seasonal_order=(2, 0, 2, 12); AIC=-2705.785, BIC=-2657.099, Fit time=59.001 seconds
Fit ARIMA: order=(2, 0, 0) seasonal_order=(3, 0, 1, 12); AIC=-2693.797, BIC=-2645.111, Fit time=137.541 seconds
Fit ARIMA: order=(2, 0, 0) seasonal_order=(3, 0, 3, 12); AIC=nan, BIC=nan, Fit time=nan seconds
Fit ARIMA: order=(2, 0, 0) seasonal_order=(2, 0, 1, 12); AIC=-2634.110, BIC=-2591.509, Fit time=51.667 seconds
Fit ARIMA: order=(3, 0, 0) seasonal_order=(3, 0, 2, 12); AIC=-2712.130, BIC=-2651.272, Fit time=193.465 seconds
Total fit time: 1180.442 seconds
```

Figure A.1 ARIMA Hyperparameter Grid Search for Short, Medium- and Long-Term Prediction Horizon.

And the model was trained with the least AIC exhibiting hyperparameters values from the grid search as given in figure A.2.

Statespace Model Results

Dep. Variable:	y	No. Observations:	3248
Model:	SARIMAX(2, 0, 0)x(3, 0, 2, 12)	Log Likelihood	1368.462
Date:	Fri, 05 Apr 2019	AIC	-2718.923
Time:	14:58:18	BIC	-2664.151
Sample:	0	HQIC	-2699.301
	- 3248		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.0036	0.001	3.153	0.002	0.001	0.006
ar.L1	0.4428	0.014	30.666	0.000	0.414	0.471
ar.L2	-0.1231	0.015	-8.318	0.000	-0.152	-0.094
ar.S.L12	0.6061	0.144	4.210	0.000	0.324	0.888
ar.S.L24	0.4356	0.178	2.440	0.015	0.086	0.785
ar.S.L36	-0.0510	0.039	-1.311	0.190	-0.127	0.025
ma.S.L12	-0.3585	0.141	-2.535	0.011	-0.636	-0.081
ma.S.L24	-0.5630	0.137	-4.119	0.000	-0.831	-0.295
sigma2	0.0251	0.001	48.250	0.000	0.024	0.026

Ljung-Box (Q):	478.43	Jarque-Bera (JB):	440.82
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.97	Skew:	-0.45
Prob(H) (two-sided):	0.59	Kurtosis:	4.56

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Table A.2 SARIMA Model Fit Final Best-Chosen Parameters.

Random Forest Regressor (RFR):

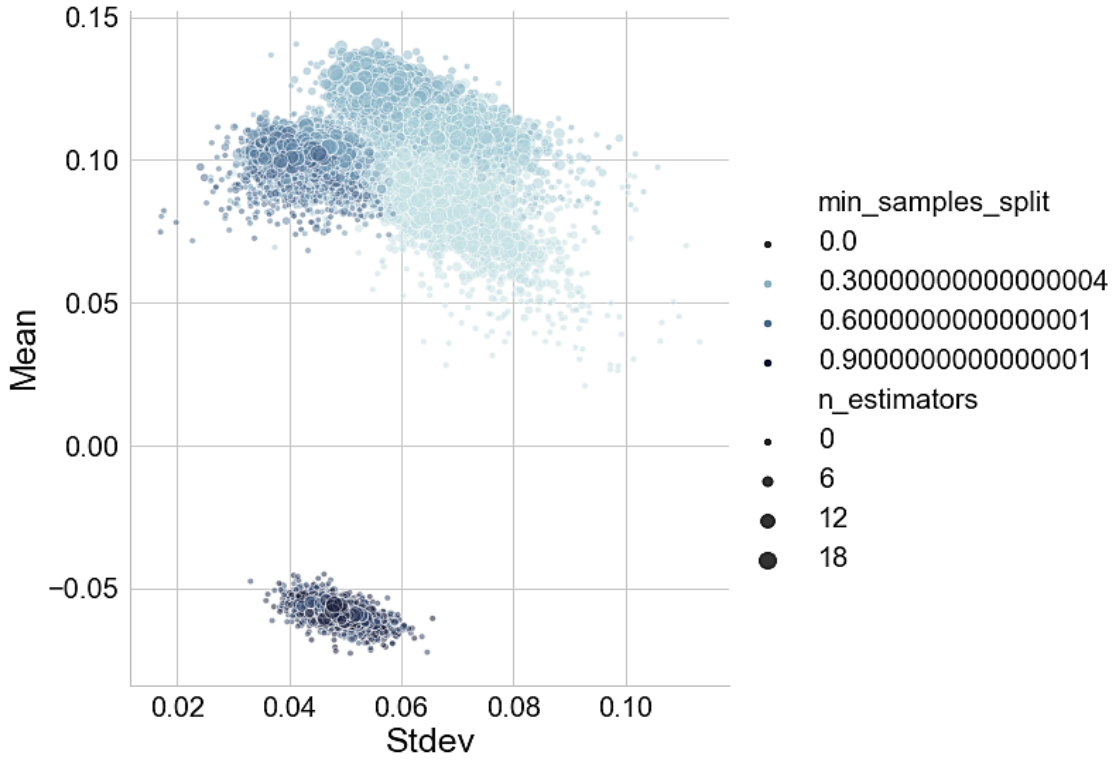


Figure A.2 RFR Hyperparameter Grid Search for Short Term Prediction Horizon.

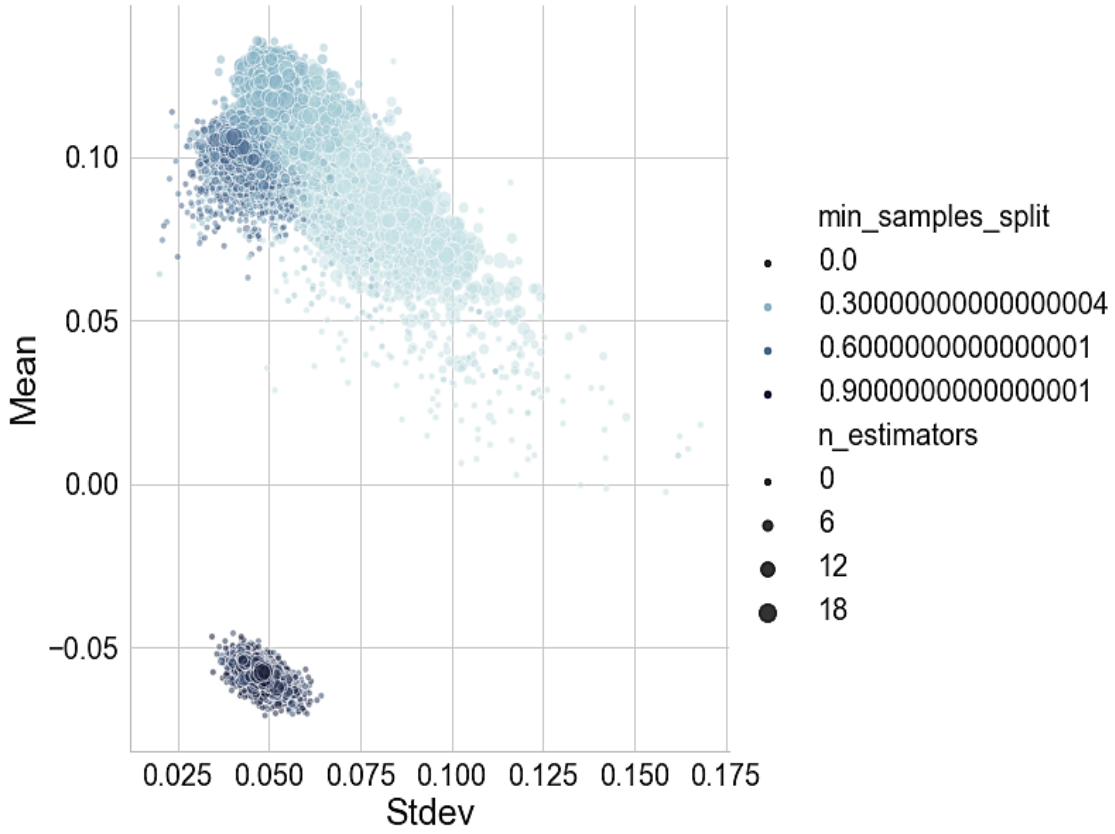


Figure A.3 RFR Hyperparameter Grid Search for Medium Term Prediction Horizon.

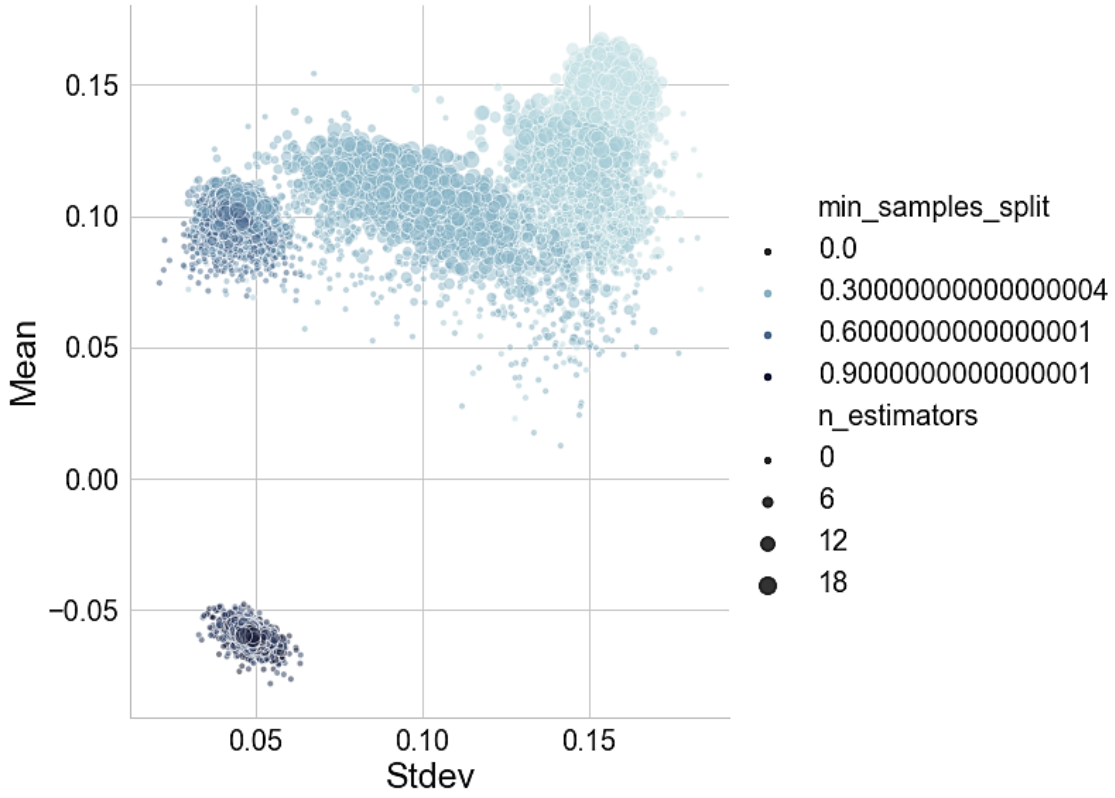


Figure A.4 RFR Hyperparameter Grid Search for Long Term Prediction Horizon.

	Prediction Horizon	Score	rfr__max_depth	rfr__min_samples_leaf	rfr__min_samples_split	rfr__n_estimators
1	Short-Term	0.140803	12	1	0.3	8
2	Medium-Term	0.135331	14	6	0.3	6
3	Long-Term	0.166954	10	4	0.1	16

Table A.3 RFR Model Fit Final Best-Chosen Parameters.

Support Vector Regressor (SVR):

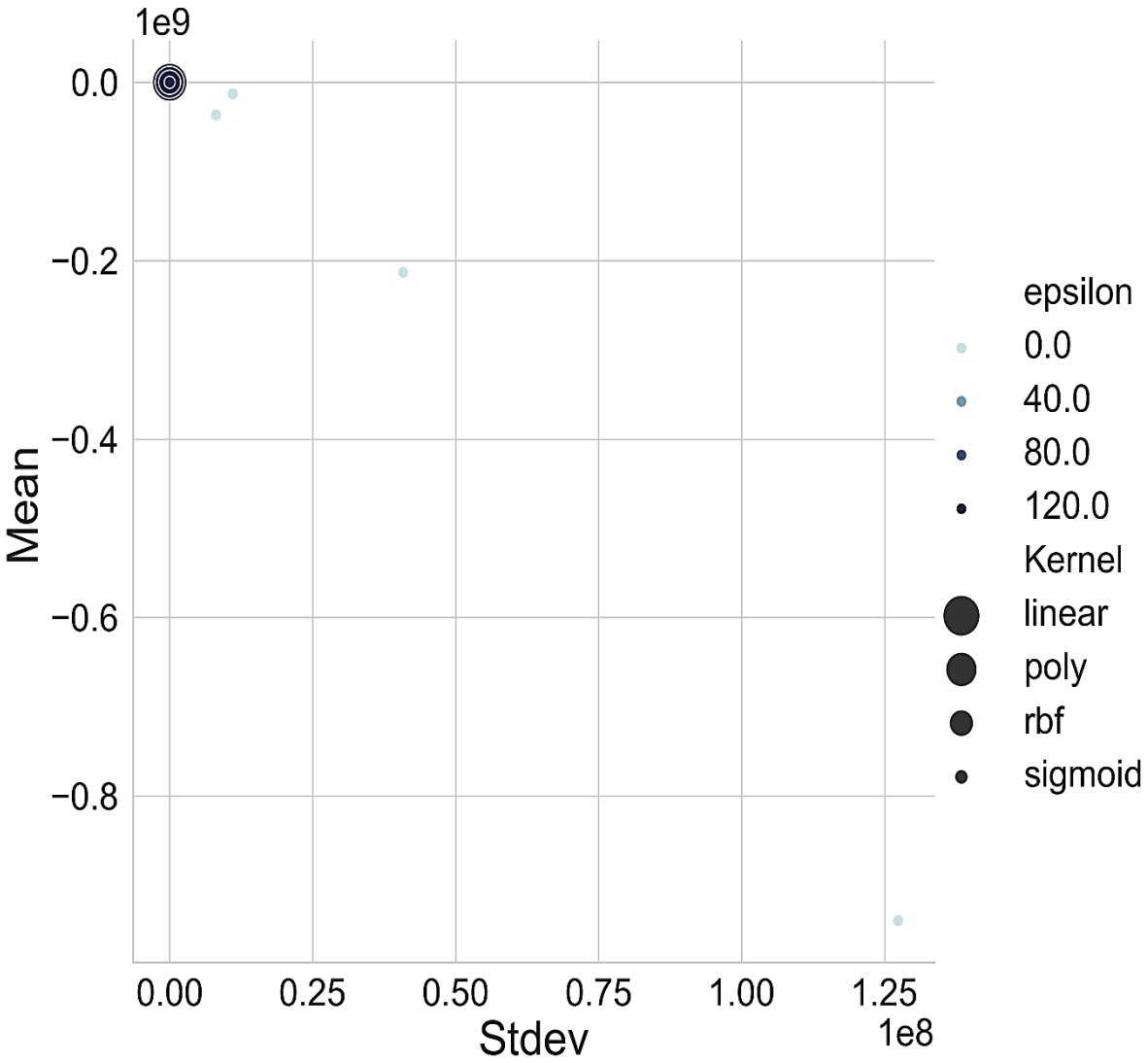


Figure A.5 SVR Hyperparameter Grid Search for Short Term Prediction Horizon.

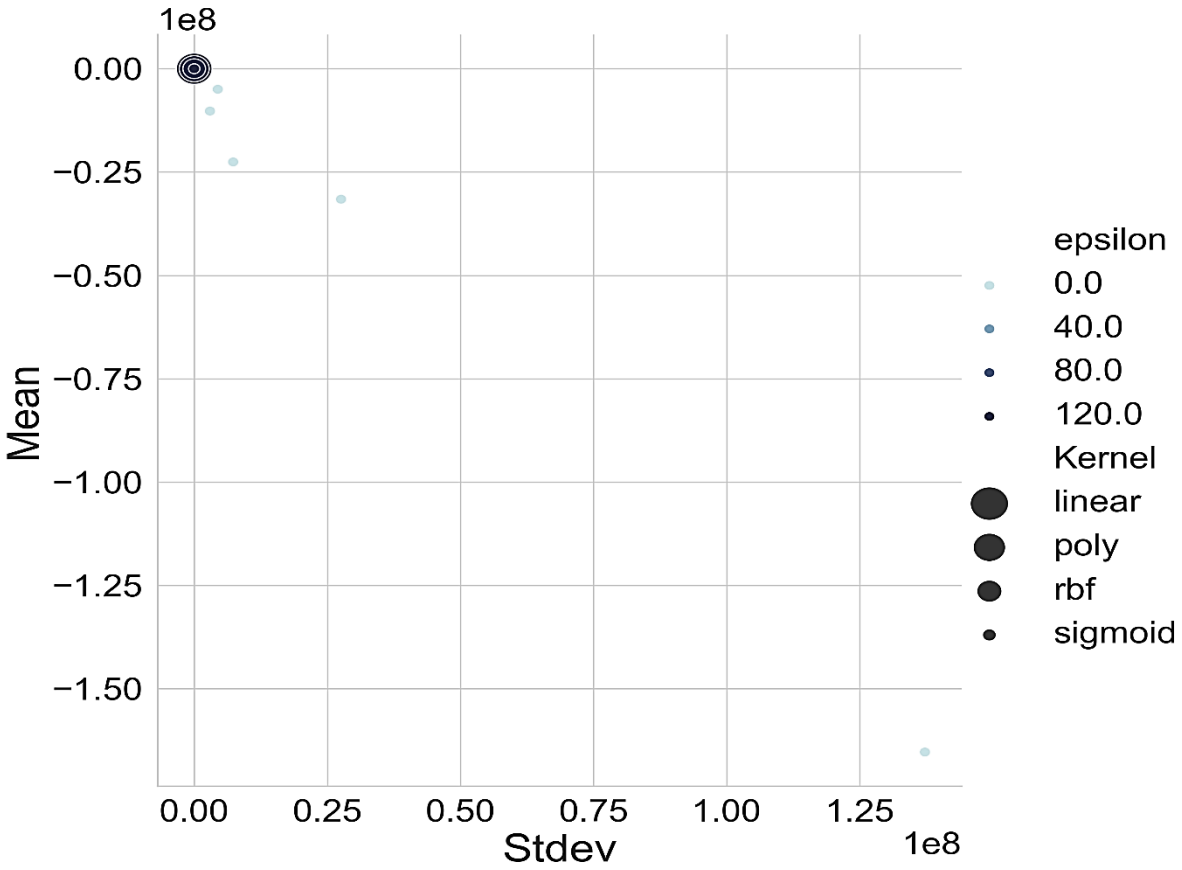


Figure A.6 SVR Hyperparameter Grid Search for Medium Term Prediction Horizon.

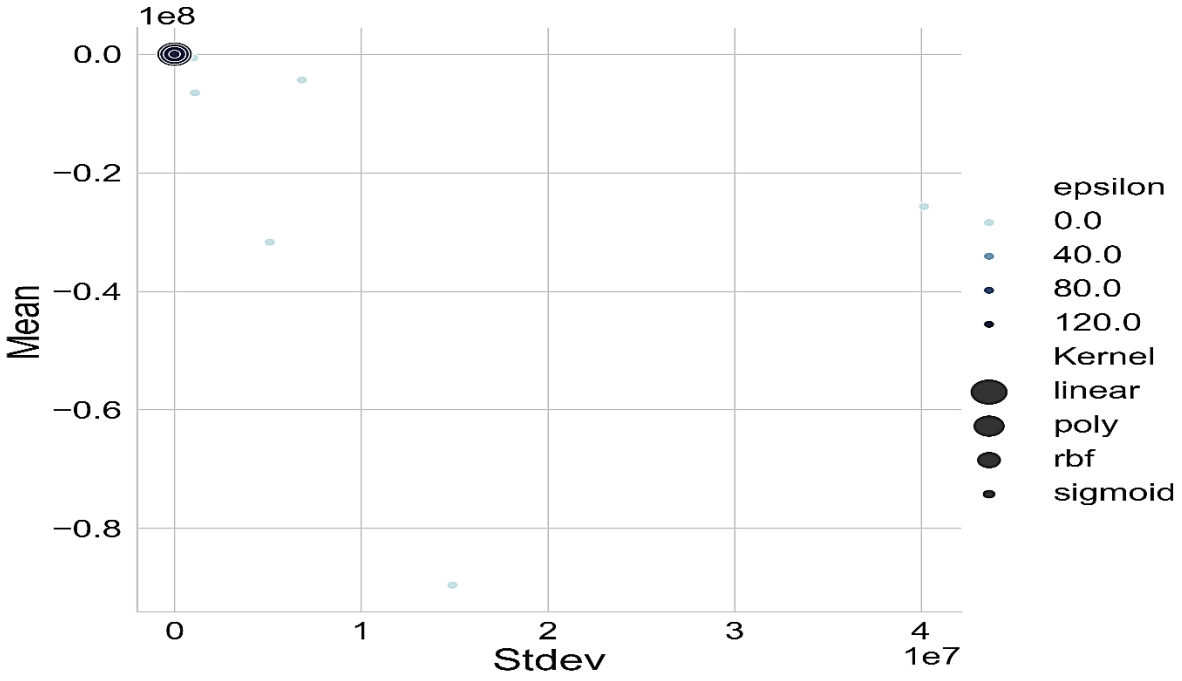


Figure A.7 SVR Hyperparameter Grid Search for Long Term Prediction Horizon.

	Prediction Horizon	Score	svr_C	svr_coef0	svr_epsilon	svr_gamma	svr_kernel
1	Short-Term	0.144922	10	0.5	0.01	0.001	sigmoid
2	Medium-Term	0.148376	100	2.5	0.1	0.1	sigmoid
3	Long-Term	0.212718	0.1	0	0.1	100	rbf

Table A.4 SVR Model Fit Final Best-Chosen Parameters.

Feed Forward Backpropagation Neural Network (FFBNN):

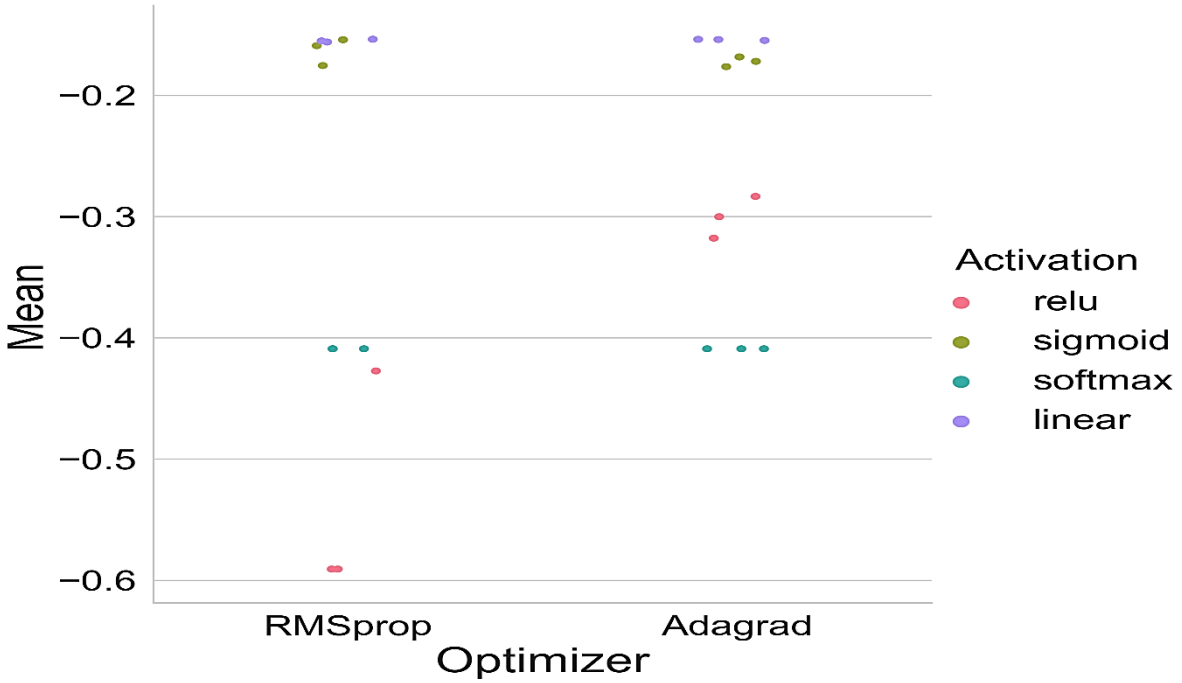


Figure A.8 FFBNN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect to Optimizers and Activation Functions.

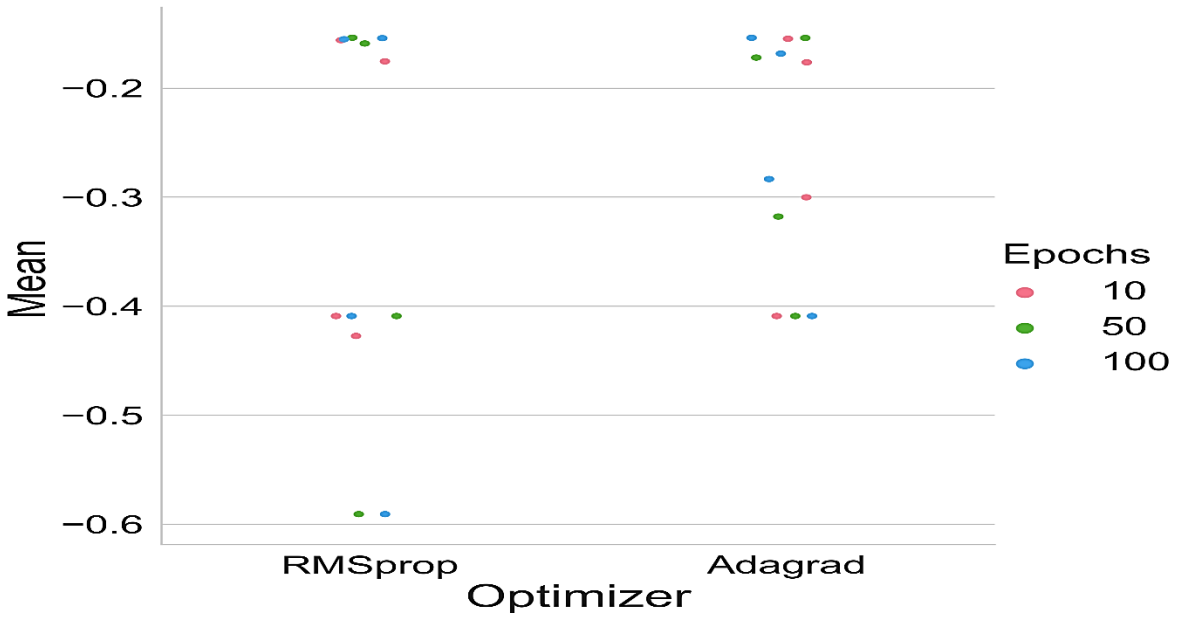


Figure A.9 FFBNN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect to Optimizers and Number of Epochs.

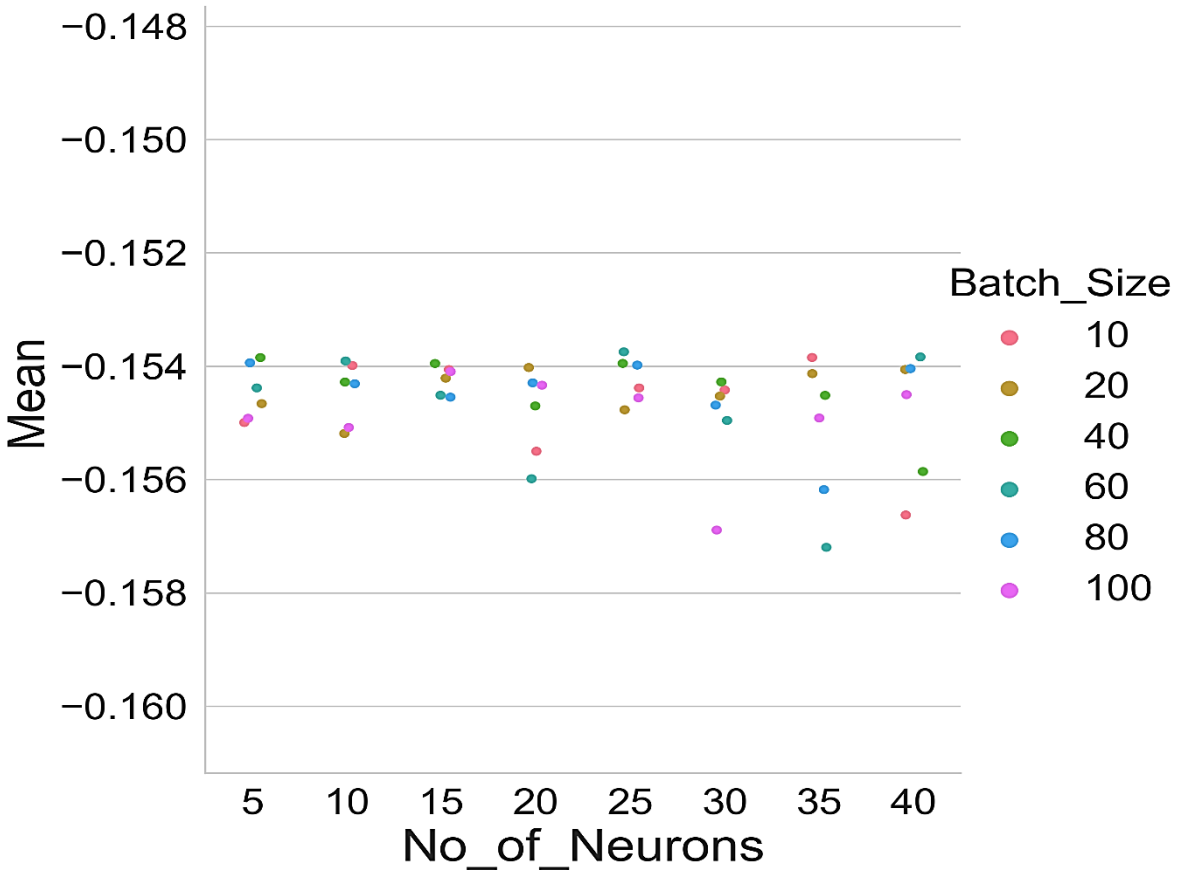


Figure A.10 FFBNN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect to No of Neurons and Batch Sizes.

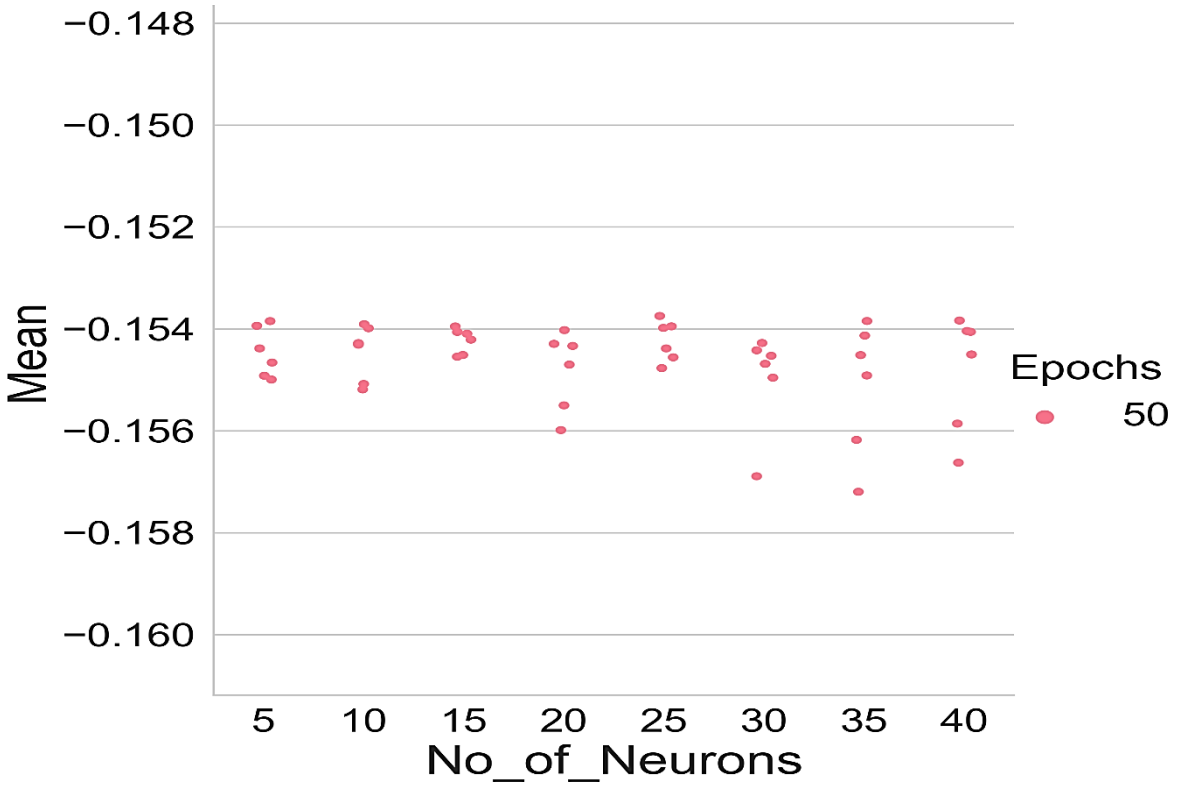


Figure A.11 FFBNN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect to No of Neurons and Epochs.

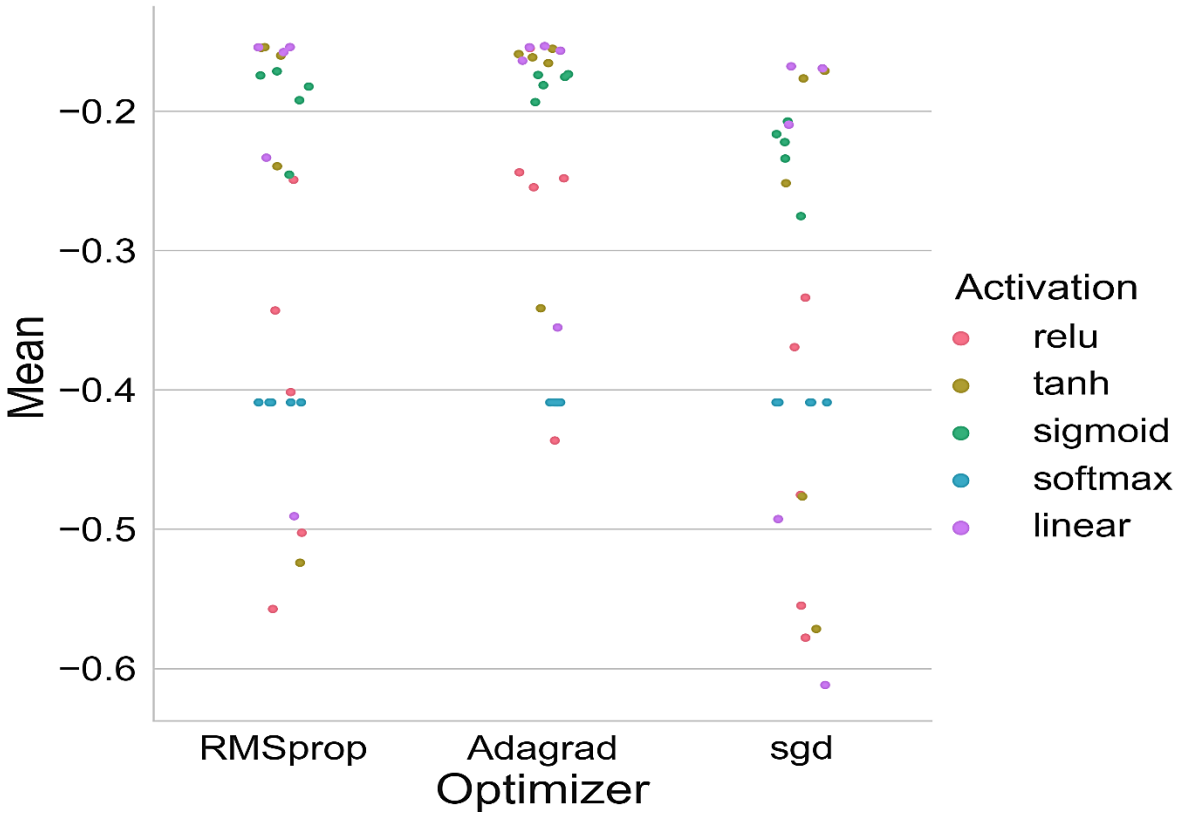


Figure A.12 FFBNN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect to Optimizers and Activation Functions.

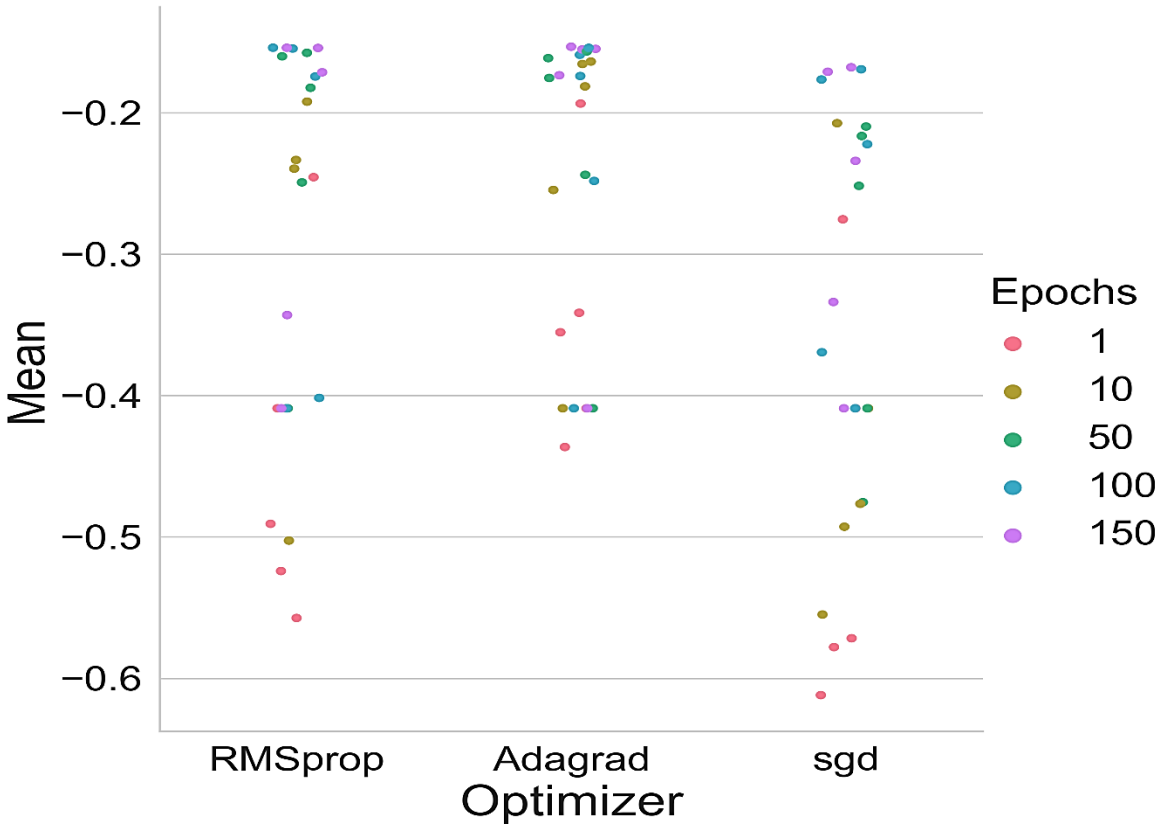


Figure A.13 FFBNN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect to Optimizers and Number of Epochs.

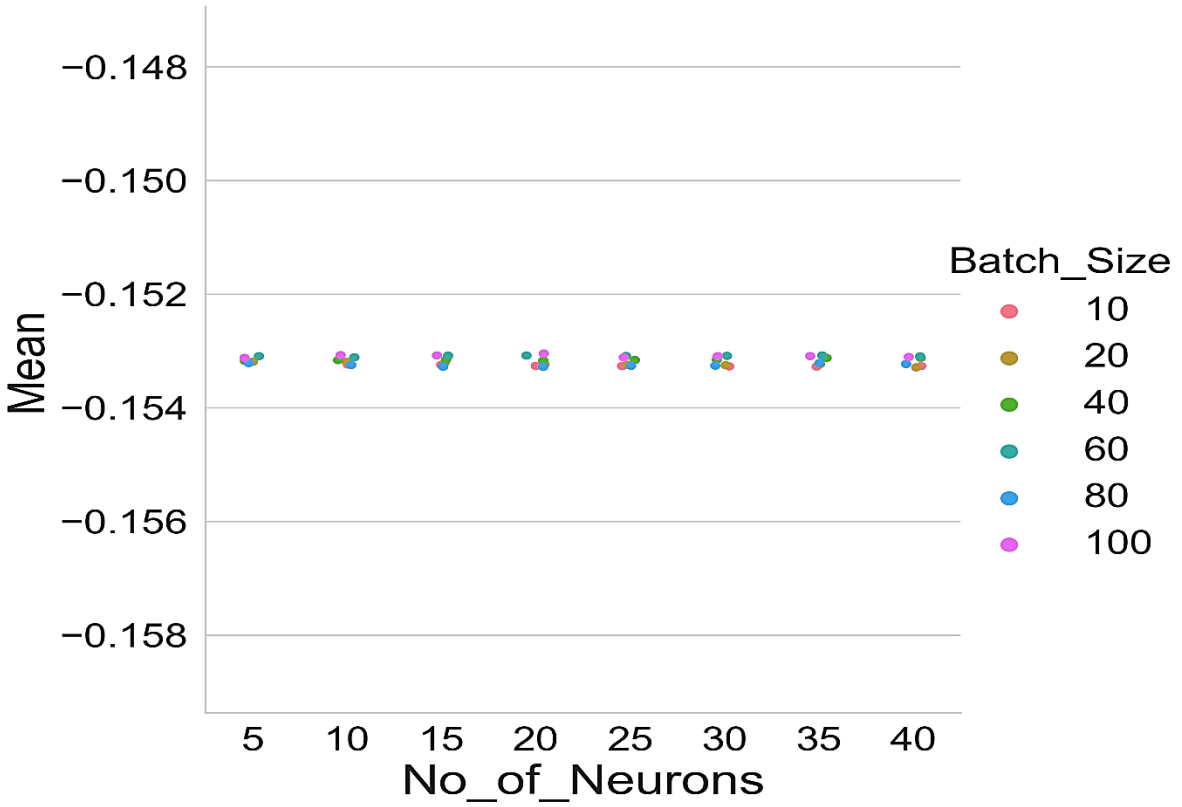


Figure A.14 FFBNN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect to No of Neurons and Batch Sizes.

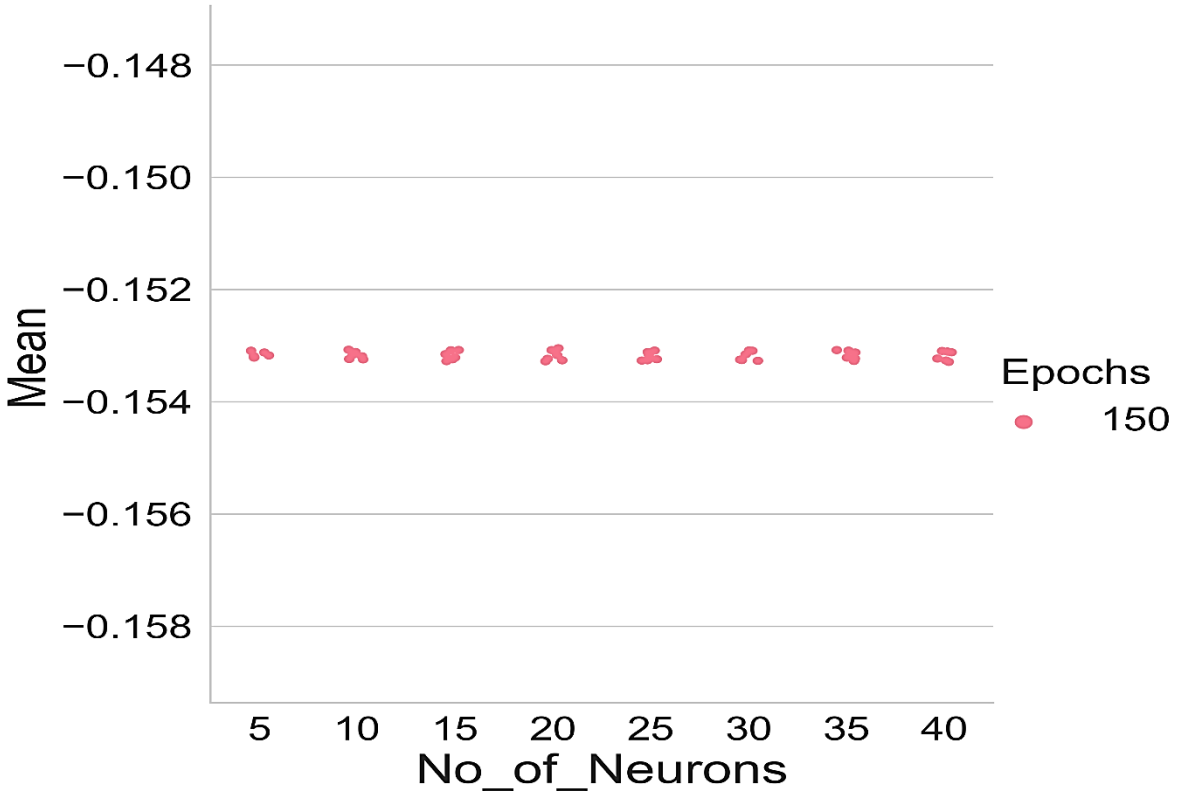


Figure A.15 Figure FFBNN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect to No of Neurons and Epochs.

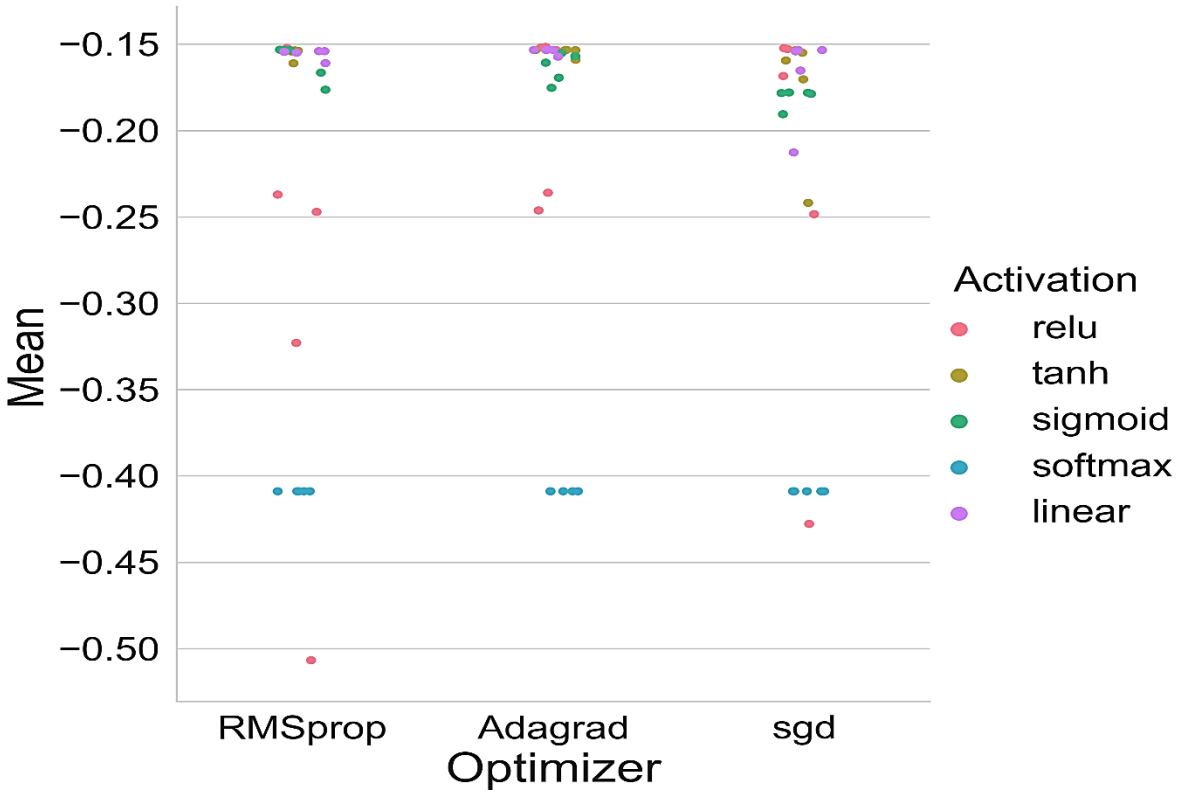


Figure A.16 Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect to Optimizers and Activation Functions.

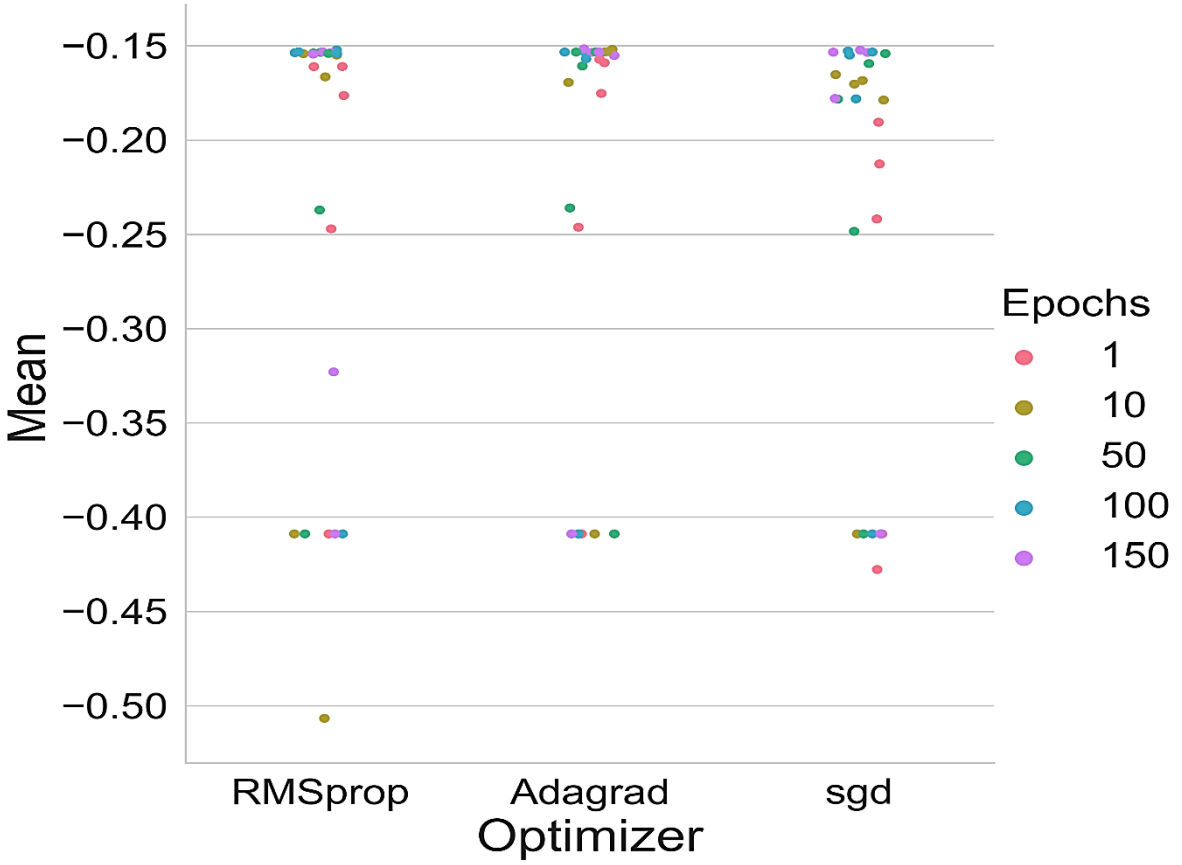


Figure A.17 FFBNN Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect to Optimizers and Number of Epochs.

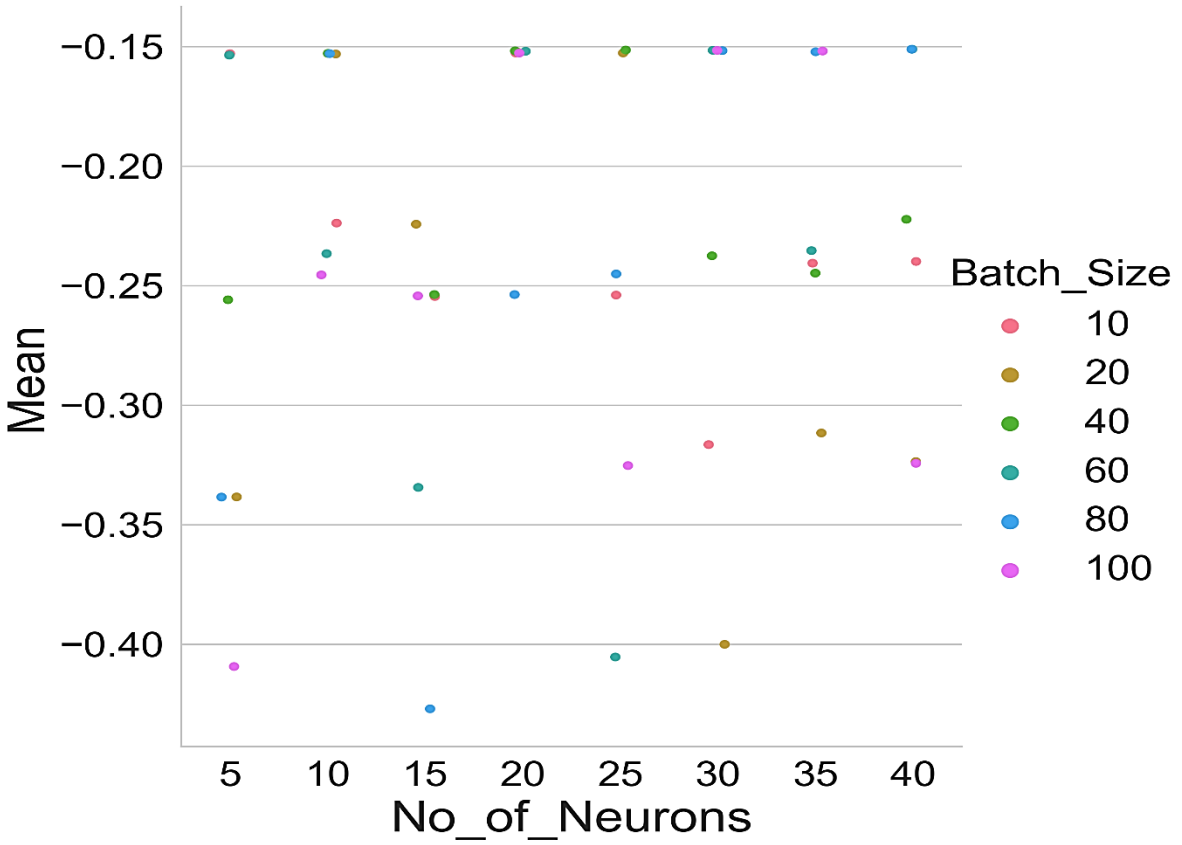


Figure A.18 FFBNN Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect to No of Neurons and Batch Sizes.

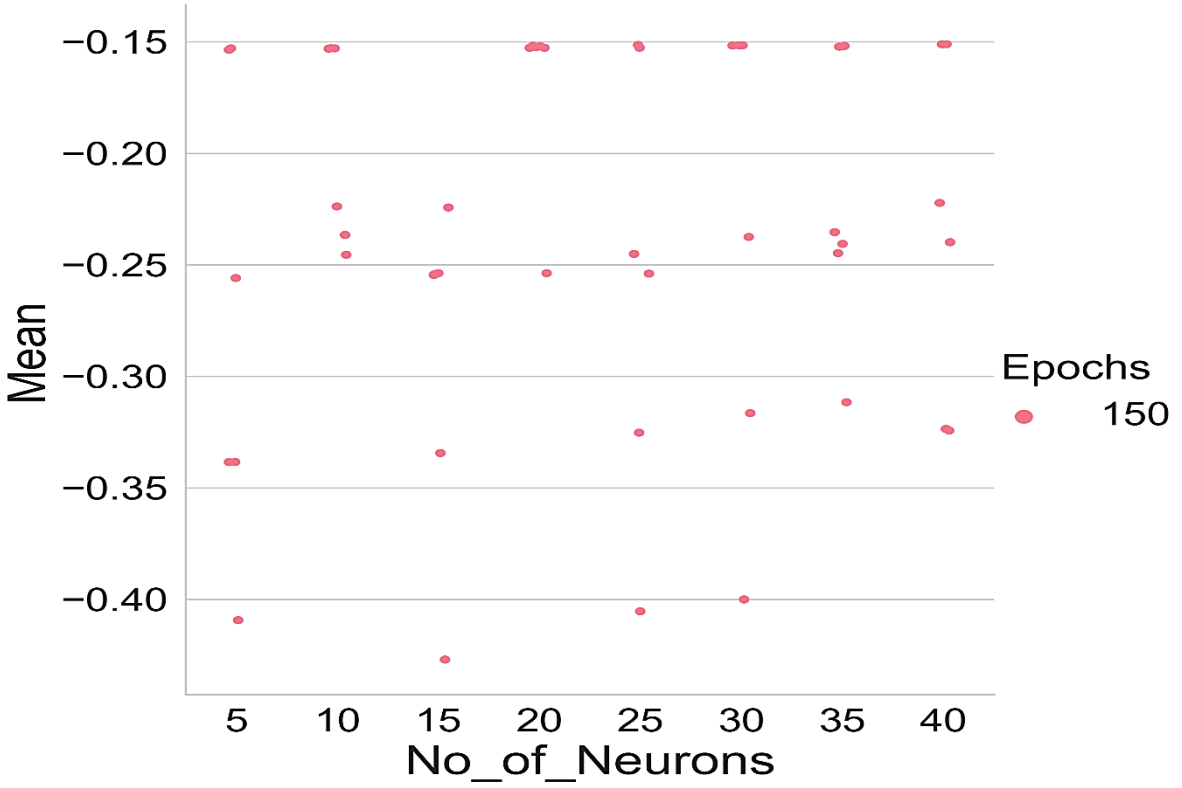


Figure A.19 Figure FFBNN Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect to No of Neurons and Epochs.

	Prediction Horizon	Score	clf__activation	clf__optimizer	clf__batch_size	clf__epochs	clf__neurons
1	Short-Term	-0.153748	linear	RMSprop	60	50	25
2	Medium-Term	-0.153047	linear	Adagrad	20	150	100
3	Long-Term	-0.151165	relu	Adagrad	60	150	40

Table A.5 FFBNN Model Fit Final Best-Chosen Parameters.

Deep Belief Network (DBN):

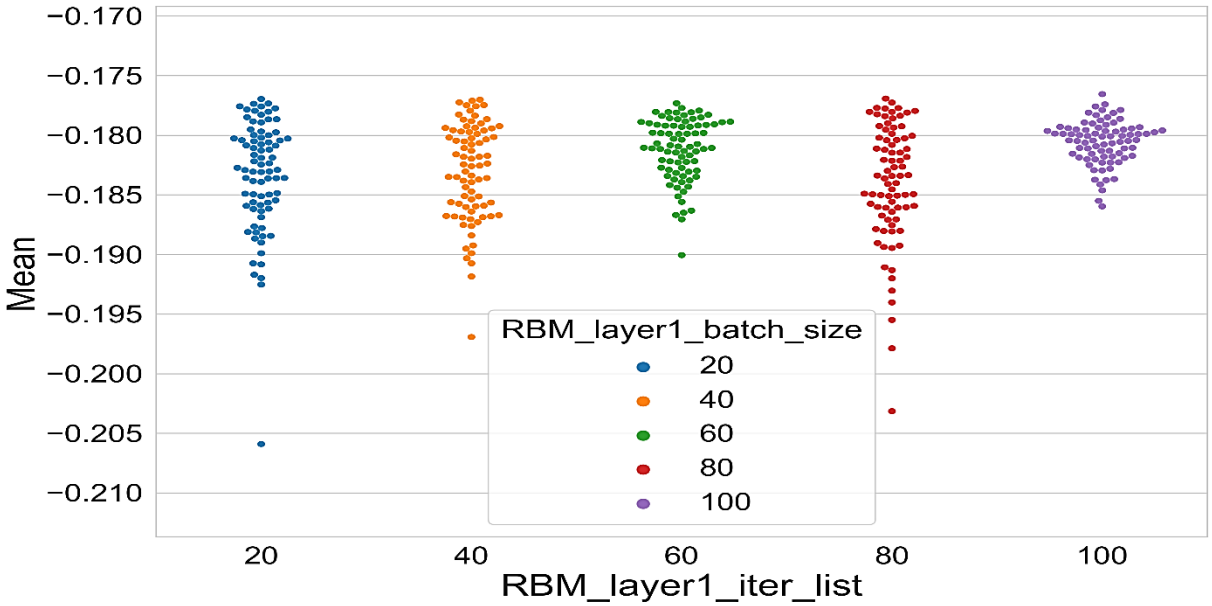


Figure A.20 DBN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect to the First RBM Layer Iterations and First Layer RBMs Batch Size.

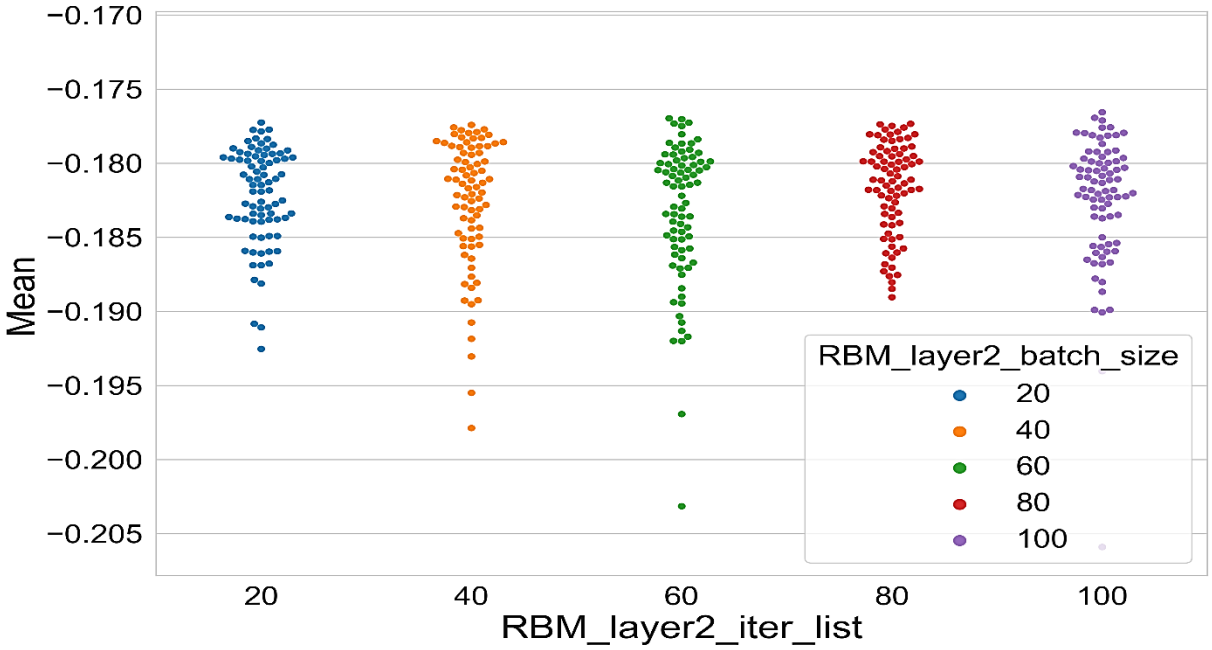


Figure A.21 DBN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect to The Second RBM Layer Iterations and Second Layer RBMs Batch Size.

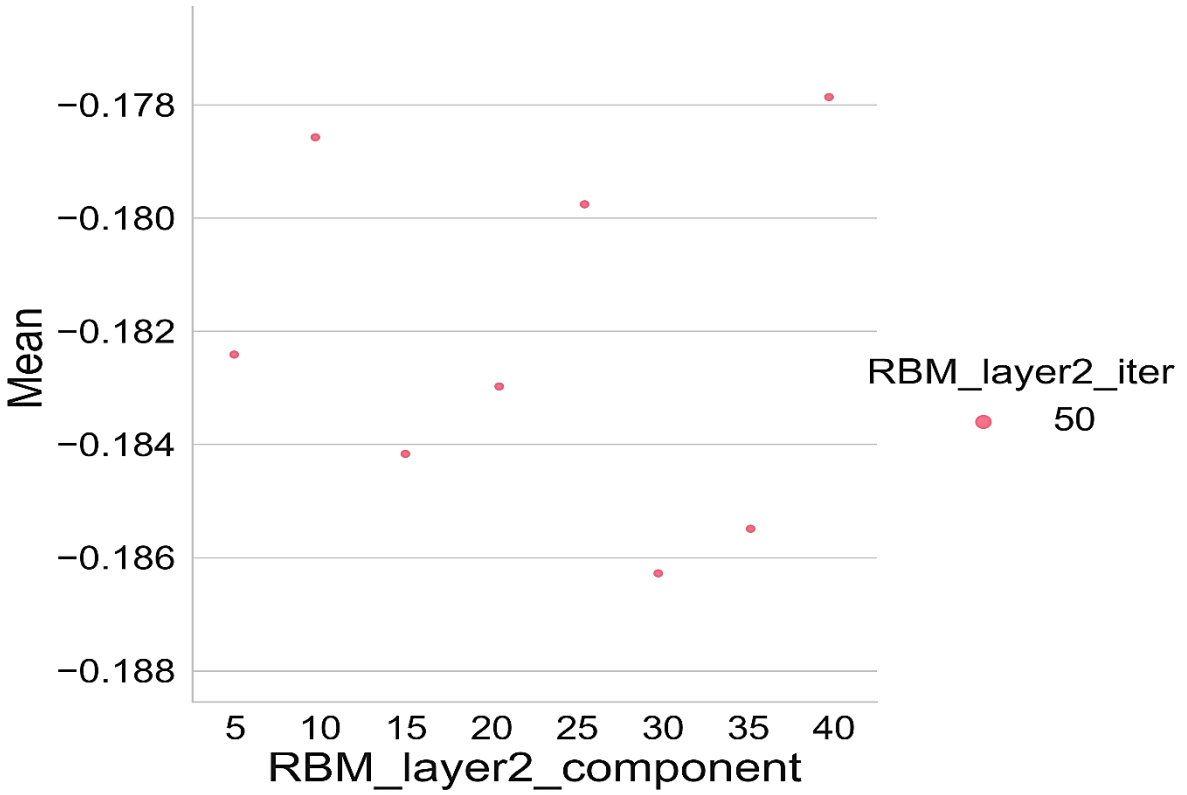


Figure A.22 DBN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect to The Second RBM Layer Iterations and Second Layer RBMs Numbers.

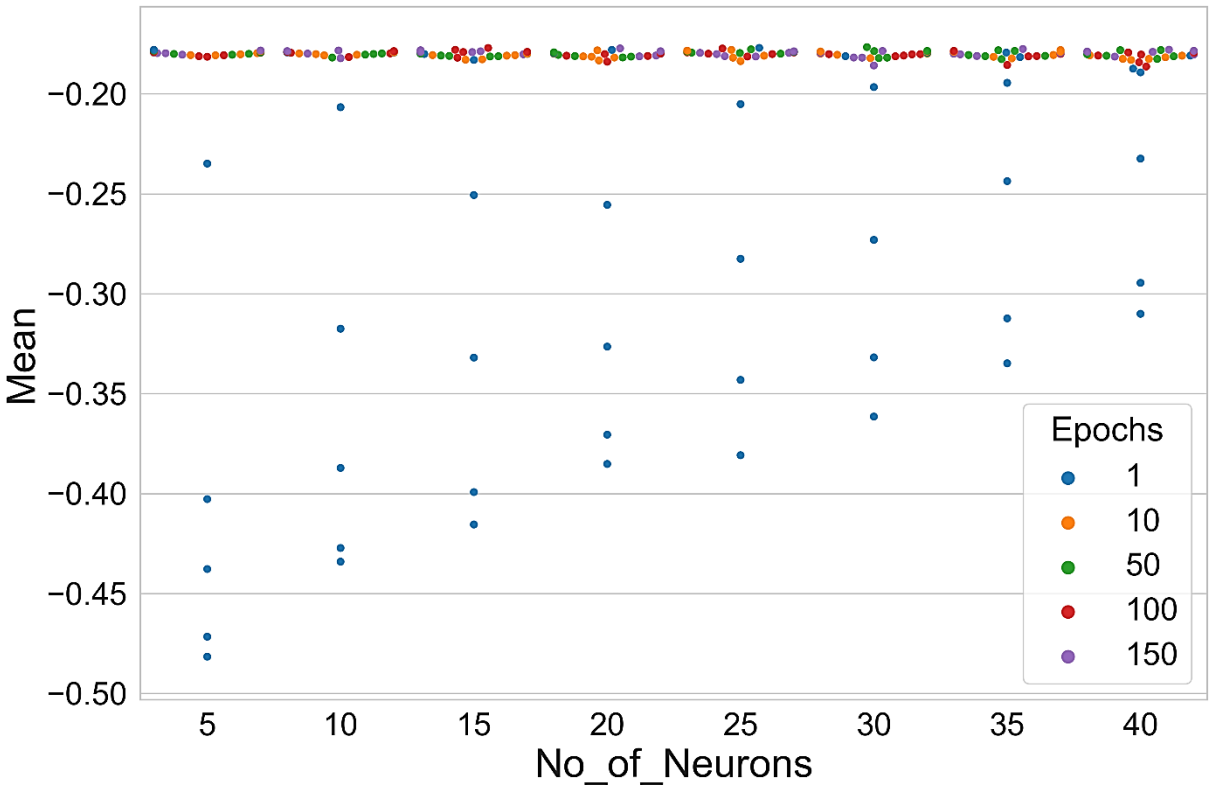


Figure A.23 DBN Hyperparameter Grid Search, Mean Results for Short Term Prediction Horizon with Respect To The Number OF Neurons and Epochs.

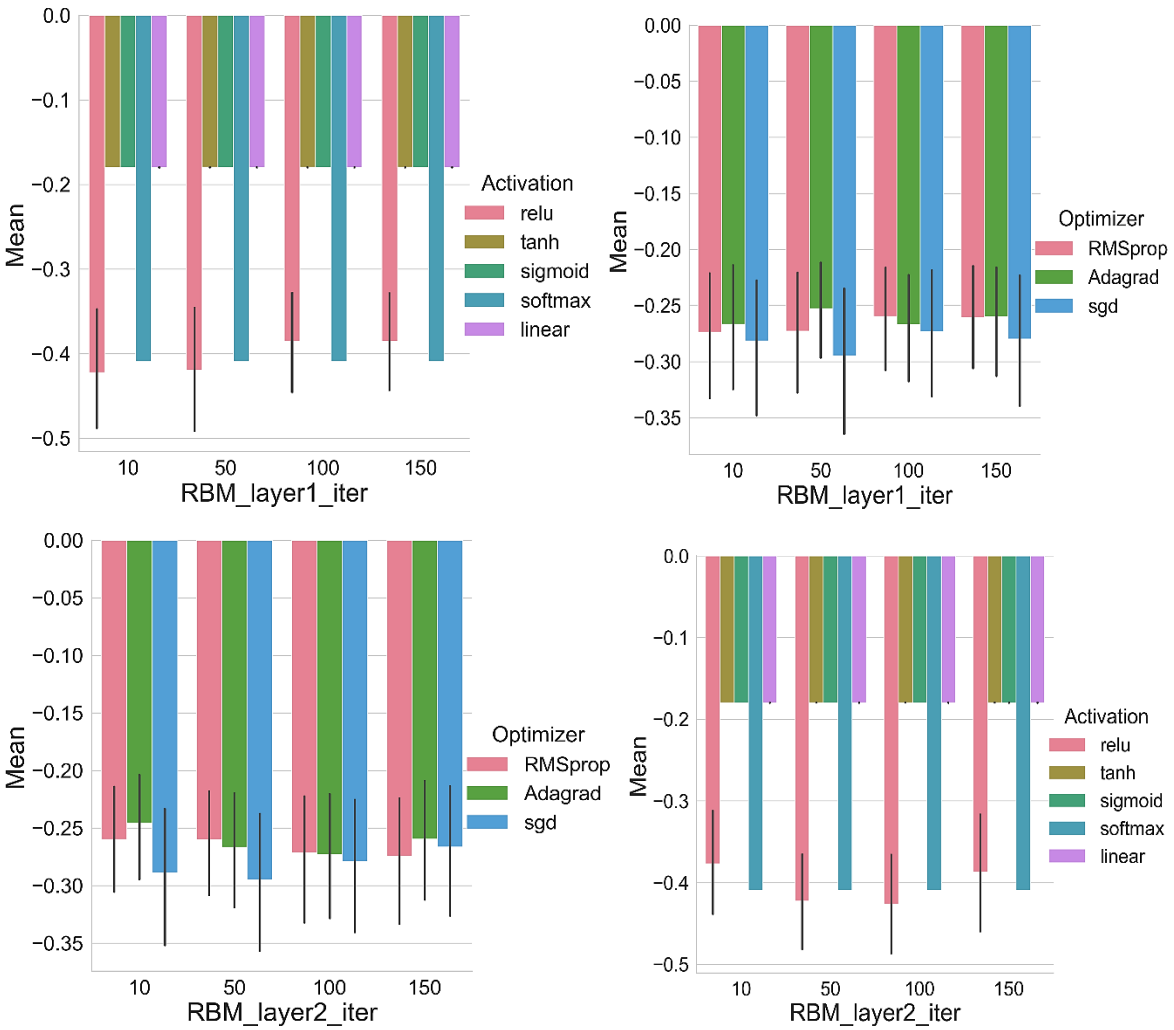


Figure A.24 DBN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect to The First & Second RBM Layer Iterations and RBM numbers and the Model Activation and Optimizer Functions.

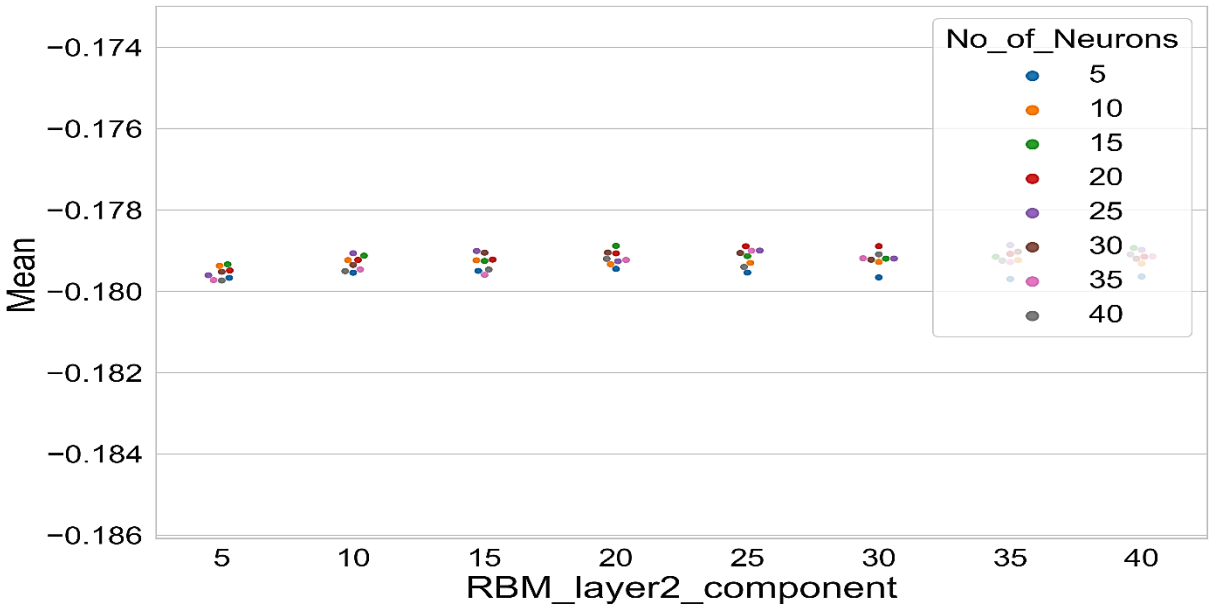


Figure A.25 DBN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect To The Number OF Neurons and Second RBM Numbers.

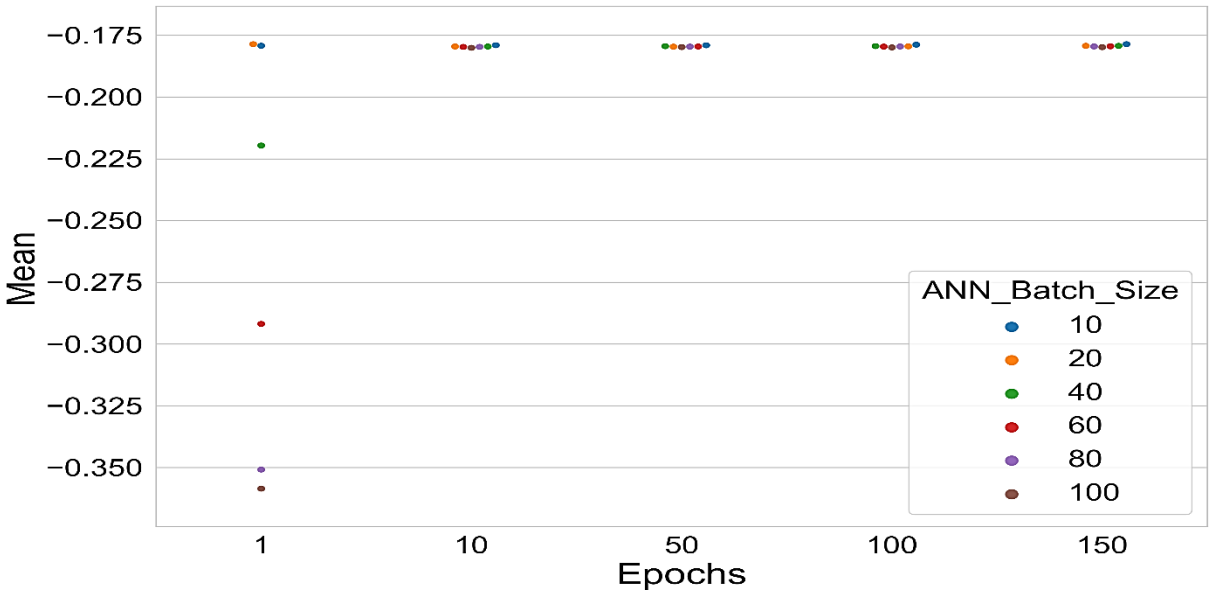


Figure A.26 DBN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect to The Neural Layer Batch Size and Number of Epochs.

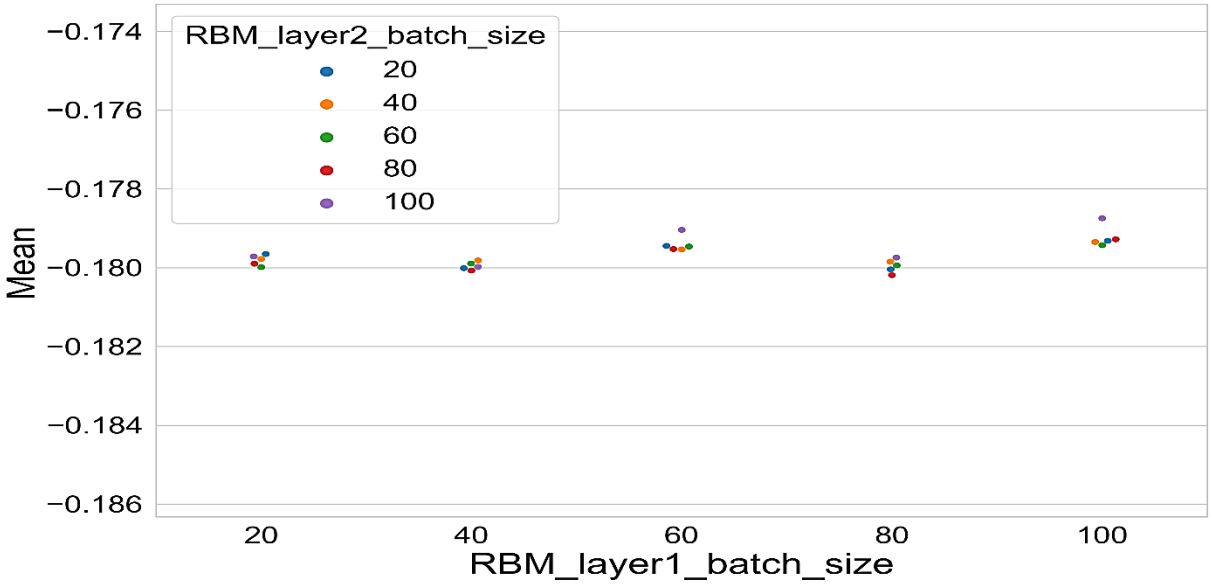
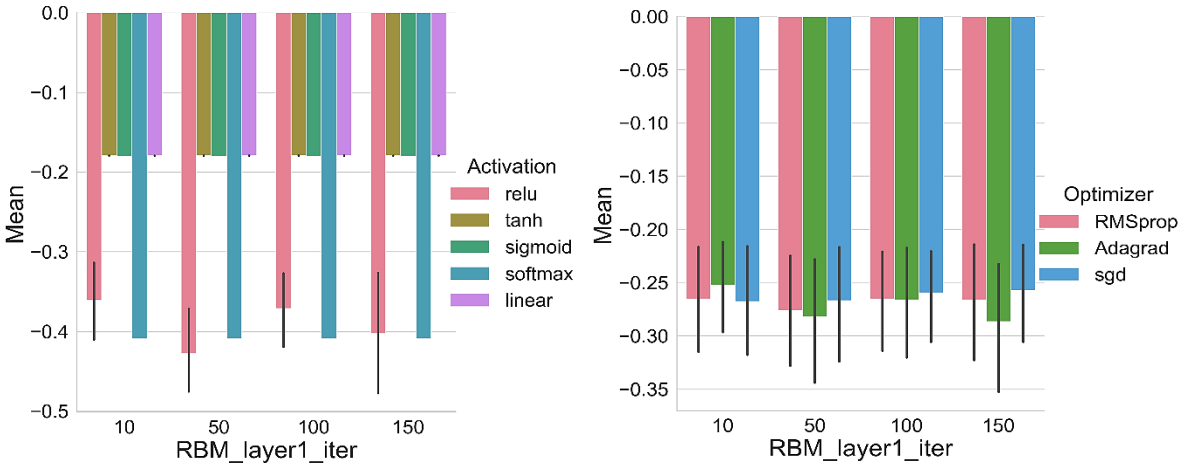


Figure A.27 DBN Hyperparameter Grid Search, Mean Results for Medium Term Prediction Horizon with Respect to The RBM Layer Batch Sizes.



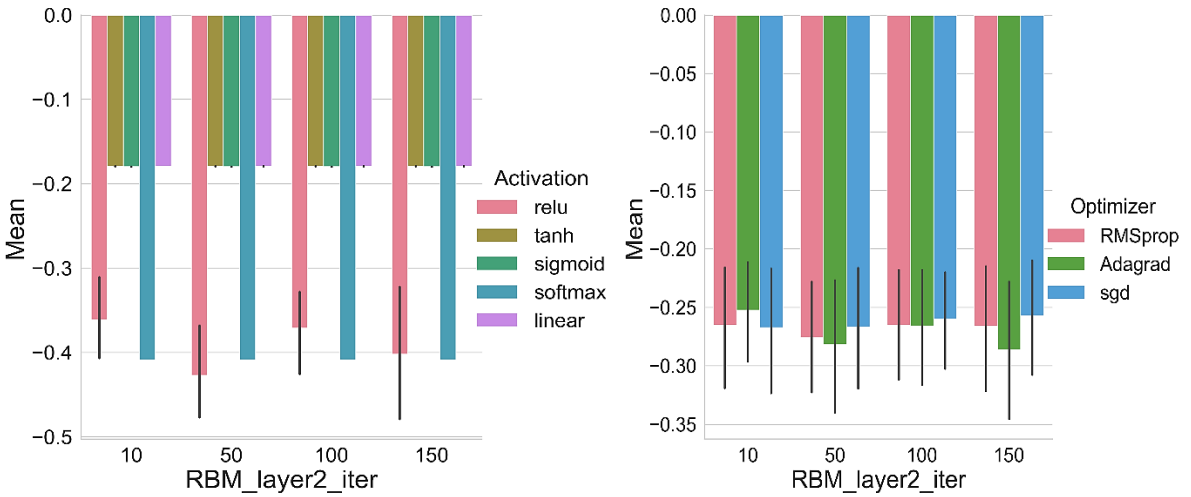


Figure A.28 DBN Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect to The First & Second RBM Layer Iterations and RBM numbers and the Model Activation and Optimizer Functions.

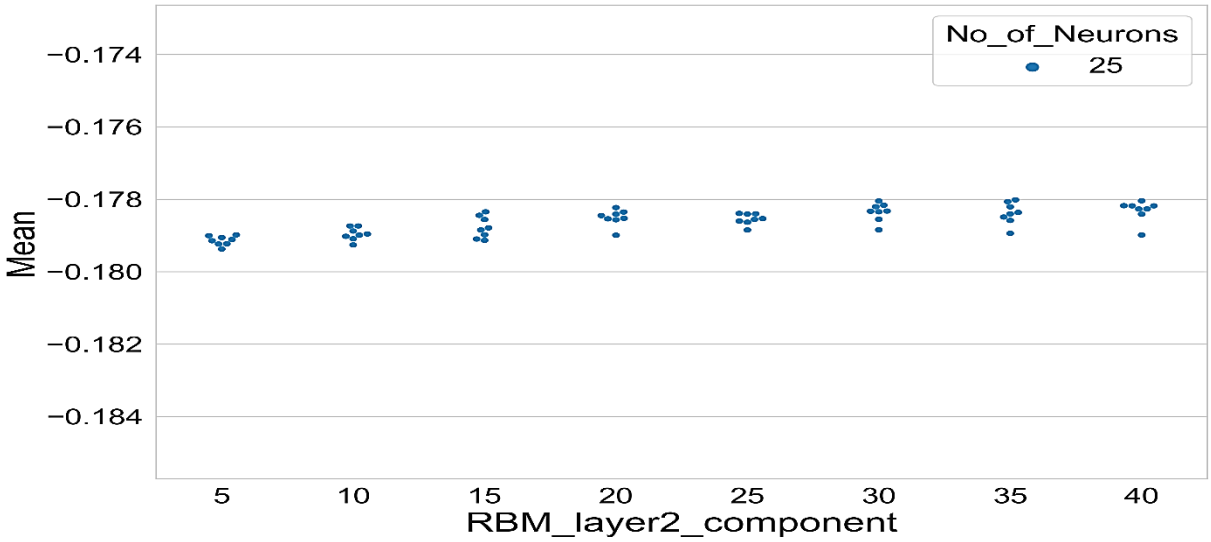


Figure A.29 DBN Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect To The Number OF Neurons and Second RBM Numbers.

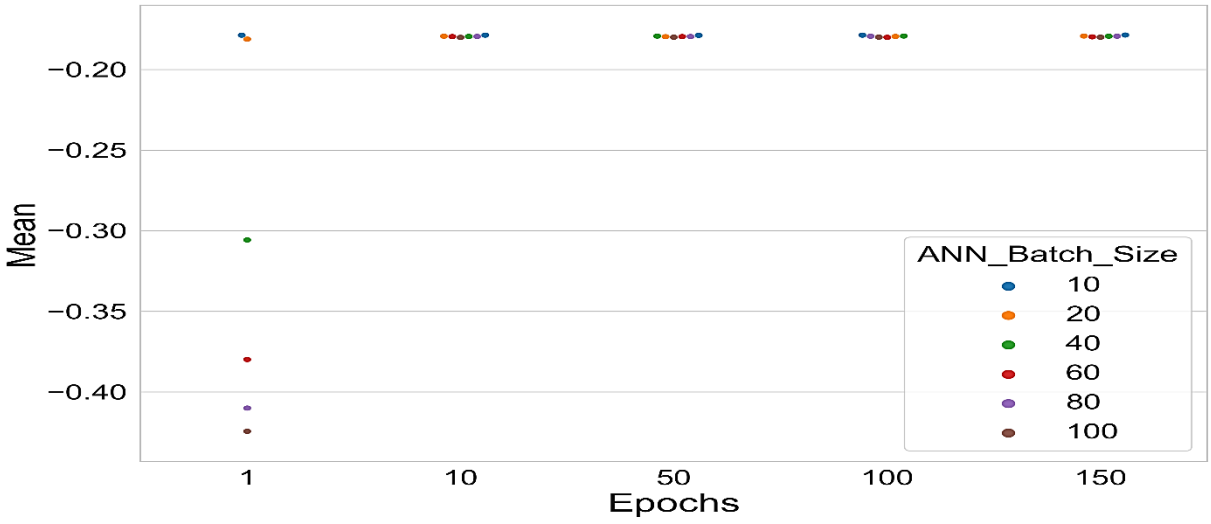


Figure A.30 DBN Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect to The Neural Layer Batch Size and Number of Epochs.

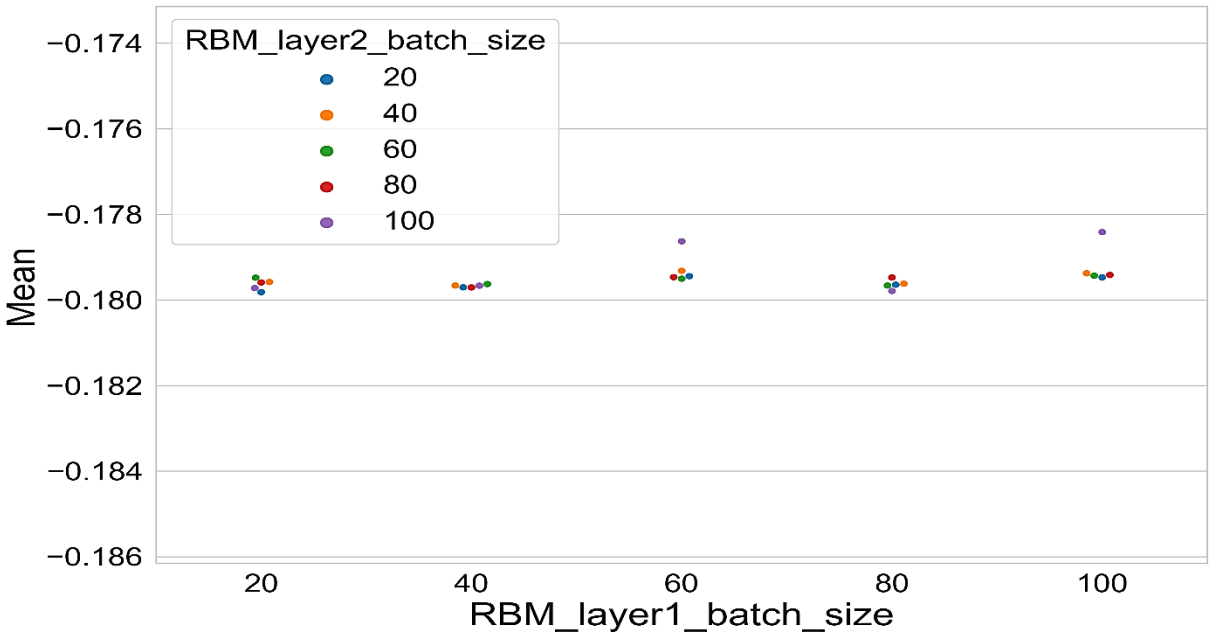


Figure A.31 Figure A.27 DBN Hyperparameter Grid Search, Mean Results for Long Term Prediction Horizon with Respect to The RBM Layer Batch Sizes.

	Prediction Horizon	Score	clf__activation	clf__optimizer	clf__neurons	clf__batch_size	clf__epochs	rbm2__n_components	rbm1__batch_size	rbm1__n_iter	rbm2__batch_size	rbm2__n_iter
1	Short-Term	-0.176597	linear	RMSprop	30	60	50	40	100	50	100	50
2	Medium-Term	-0.178747	tanh	RMSprop	25	10	150	35	100	100	100	100
3	Long-Term	-0.178415	linear	RMSprop	10	10	150	35	100	150	100	150

Table A.6 DBN Model Fit Final Best-Chosen Parameters.

Convolutional Neural Network (CNN):

Appendix B : Continuation of Discussion of Selected Models

B.1 Long Short-Term Memory (LSTM)

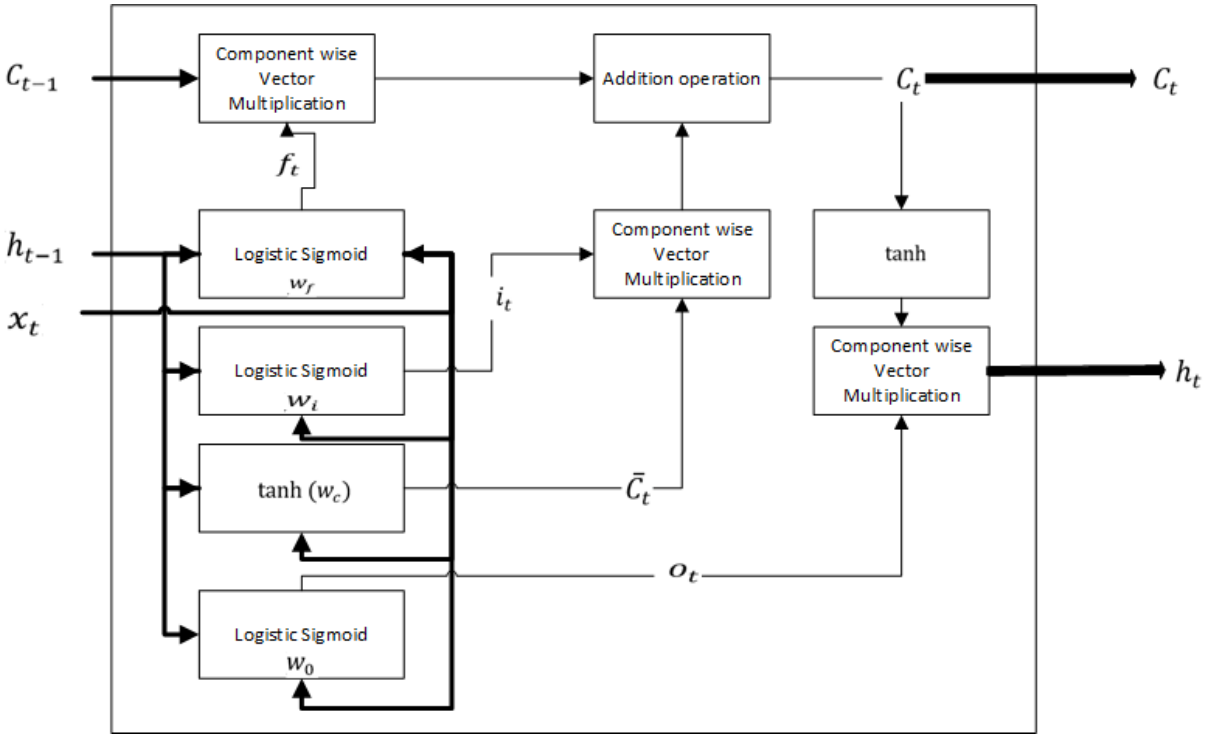


Figure B.1 Data Flow and Operations in Long Short-Term Memory (LSTM) Unit Structure which Contains The Forget, Input, Output, and Update Gates.

B.2 Traditional Neural networks (NN) VS Recurrent Neural Networks RNN:

Traditional neural networks consider each input as an independent piece of data with no relation with the next input in line during the learning phase. It's sort of a shortcoming that the machine learning model must face, if we are to consider a more sequential form of data learning with NN. The level of reasoning for the previous event or value in a traditional NN is not enough to predict for the next events in line and thus decreased classification and prediction accuracy. As all learning ML algorithms traditionally NNs have the cost function that remains single purposed and single valued whereas the structurally inherent capability of LSTM-NN and GRU-NN overcomes this problem by memorising the previous event or values. Long short-term memory and gated recurrent units neural networks (LSTM and GRU respectively) are better performing variants of RNNs. Recurrent Neural Networks are designed to address the memorisation problem because they act like the loop chain of information for each neuron in the layer that retains the information in the form of hidden memory cells. A more general recurrent neural network is shown in figure B.1. RNN can be considered a natural form of NN for the sequential based learning of data structures.

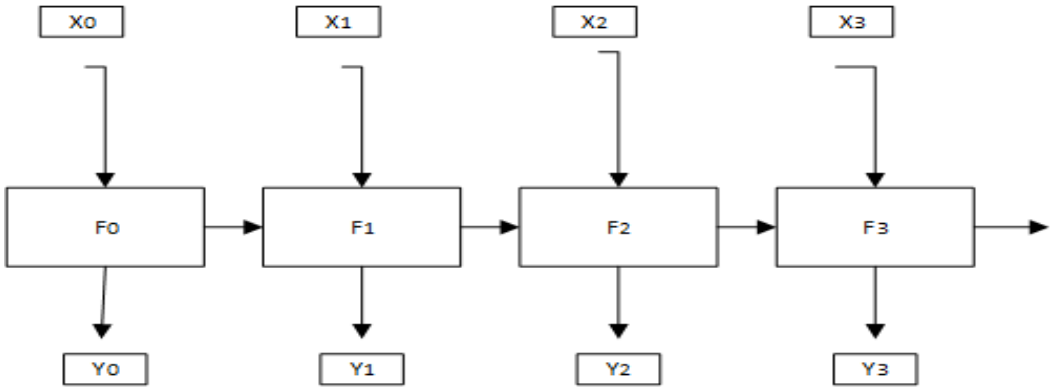


Figure B.2 General Recurrent Neural Network Structure. Unlike a feed forward neural network each neuron not only feeds its output to the next neuron in next layer but also to the next in line neuron in the same layer. So each neuron have two sources of input, the most recent and the recent data which is combined to determine how they respond to the new data.

The difference between a feedforward and recurrent neural network is that a trained feed forward network can be trained further for as many data structures for one class, but the main thing is it won't necessarily alter the classification accuracy for the other training data class. RNN ingest their own memory (F (n-1)) with information in a sequence itself. RNN use this past information to perform decisions which a feedforward network is incapable of performing.

Why Using GRU And LSTM As the Modified RNN? A larger simple RN networks exhibit the gradient exploding phenomenon while the learning of long sequences during gradient descent. LSTM and GRU solve this issue by controlling the flow of the information in RNN using various gates [83]. The basic structures of an LSTM and GRU are shown in figure B.1. Only GRU structure with mathematical is discussed here. Although the final approach is to compare the based RNN algorithm performance in our proposed model against the LSTM and GRUs and if possible propose the change in the gated recurrent units in the end. A gated recurrent unit (GRU) resembles in structure and working to that of LSTM except it doesn't contains the output gate. Which makes sure the content from the memory cell is written to the output at every time step.

A normal LSTM data flow model (refer figure B.1) mimics the operations of forget gate (f_t), output gate (o_t), input gate (i_t), hidden memory update gate (c_t) by using the mathematical operations using sigmoid function, *tanh* and vector element multiplication and additions operations. LSTM model unit inherits an additional input sequential input (c_t) for better sequence memory keeping. But the whole learning process becomes complex over time due to too much parameters although with an increased performance then simple RNN. GRU on the other hand presented after LSTM, makes its structure less complex by eliminating the need to pass an additional sequential data value (c_t) instead by just determining the hidden input (h_t) update through an update gate (W_z) and reset gate (W_r) thus eliminating the need for the output gate. Reset gate resembles in functioning to pretty much to that of the forget gate in LSTM. The overall computational complexity with less parameters involves compared to LSTM while still performing better than LSTM makes it a favourable model. Various other version of GRU have been presented in literature but we presented here just the base models of both LSTM and GRU. The mathematical model representing the GRU data flow through a single unit as shown as arrows and operations in figure 4.10 is presented in equation in 5-9.

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad (5)$$

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (6)$$

$$\tilde{h}_t = tanh(W_h[r_t \cdot h_{t-1}, x_t] + b_h) \quad (7)$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \quad (8)$$

Equation 5 gives the representation of the update gate z_t that acts as a sigmoid function on the inputs parameters W_z, b_z, were b_z, is the bias element. The reset gate r_t (6), mimics the role of the input gate as in LSTM. The next hidden state (h_t) (8), value passed to the next unit depends on both the update (z_t) or forget gate and the reset or input gates (r_t). Eliminating the need to pass another hidden memory parameter (c_t) as in LSTM model.

GRU-NN Prediction based Node Links Flow Rate Probability Estimations:

After the GRU-NN model is developed next step is to train the model on the individual links flow rate dataset that we have already filtered in the Data preliminary analysis (refer section 3.6). According to the overall proposed model (refer section 5.7 methodology framework) different instances of GRU-NN model are trained separately on time series flow rate dataset (refer table 5.5) for all the links with inflows and outflows separate data trained models. Utilising the flow rate probability prediction for the link whose unidirectional flow rate data is used for the training considering, the idea is to identify the bottleneck link at each node considering the flowrate likelihood probability of all other links on a single node. The bottle neck link is the one that limits the flow of traffic when compared to other links in a node. The bottleneck link identification is done relative to all the link in a node this is because every node has different flow rates due to a number of factor mainly being their location in the road network.

Gaussian Mixture Model Distribution Estimation (GMM) On Historical Links Flow Rate Data:

The predicted flow rate probability $P\left(f_{i,t}\right)_{L1}$ gives the probability of the link 1. The effect of other links in a node are to be taken care of by estimating their likelihood flow rate probabilities on at a time for all the instances. Noe Links can be regarded as the cause links and the effect links somewhat same methodology have been proposed in [101][102], which describes Bayesian network as a Gaussian mixture model (GMM. We opt to use the GMM model for each individual link flowrate estimation using past data for different time instances using the predefined parameters as given in equation (9).

Appendix C : Future Works

C.1 Flow Rate Network Bottleneck Identification

The future implementation of this thesis work may include the combination of deep learning in combination with statistical methods including but not limited to Recurrent Neural Networks based Gaussian Bayes (RNNGB) model-based estimates for the flow rate probability for each link using the historical flow rate data. Flow rate constraint patches can be found within each patch. The idea is to inspect each link patch to identify the link that is behaving as a bottleneck for restricting flow through it at any time interval. For example, a link in a certain direction may act as a flow limiter while at a different time interval the same link will work as a non-restricting flow link. The bottleneck identification is done for each node links in a relative fashion. Figure C.1 shows the general model working principle on the whole network. Based on the congestion bottleneck identification, nodes are classified for free flow or congested flow nodes at different time intervals.

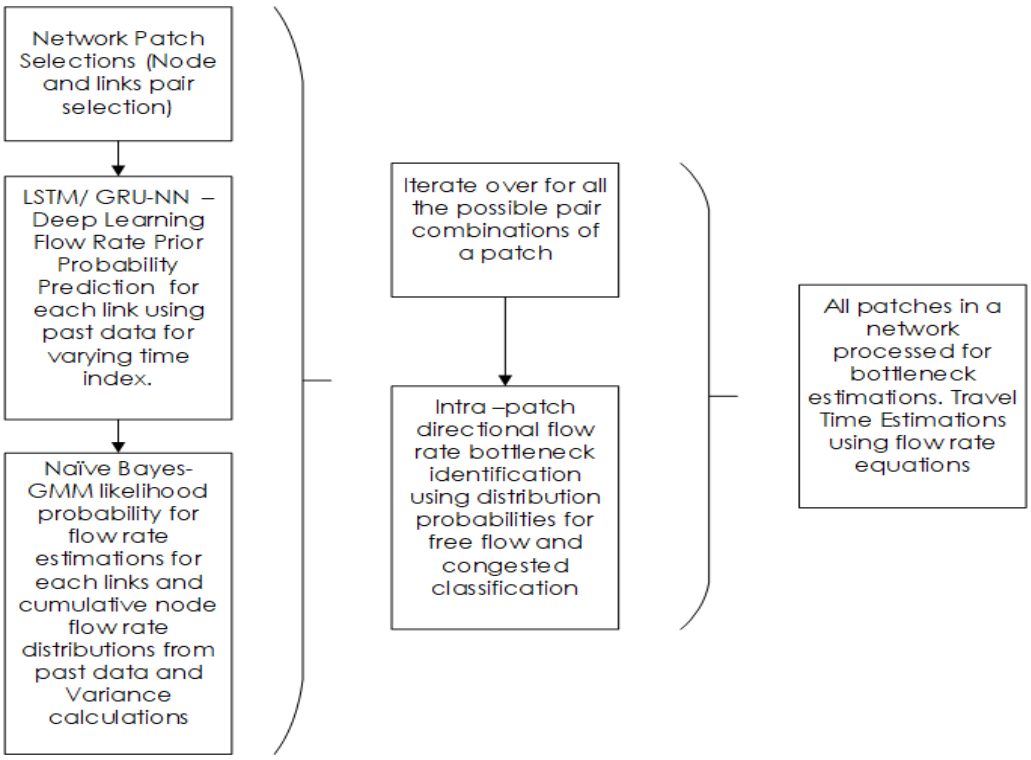
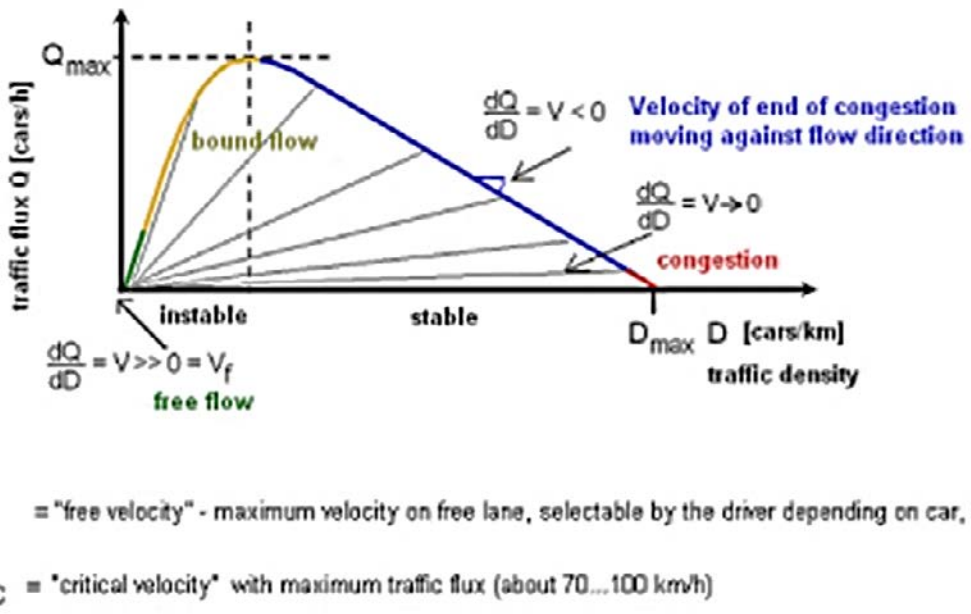


Figure C.1 Systematic layout of GRNGB Model.

As a general understanding, when a flowrate bottleneck is created in a node position, the resulting congestion then propagates patch by patch, right from the point of origin of constriction and in the opposite direction of the traffic flow. Since the RNNGB model considers the direction of congestion waves so the bottleneck identification becomes easy for each direction of traffic flow. Figure C.2 explains the basic concepts of the bound-flow Q_{max} (bottleneck point) to be found for each individual directional link using RNNGB model. As general approach shown in figure C.2, traffic flow link is considered congested if it falls below Q_{max}, into the stable portion of the traffic density with ever decreasing traffic flux. The point of location of Q_{max} and D_{max} will differ for each link at different interval time aggregations for each node.



V_f = "free velocity" - maximum velocity on free lane, selectable by the driver depending on car, skill etc.

V_C = "critical velocity" with maximum traffic flux (about 70...100 km/h)

Figure C.2 Traffic Flux versus Traffic Density generalised observation with optimum traffic flux point Q (max) differentiating the instable and stable unidirectional flow for a single link in any node [99].

C.1.2 Average Congestion Speed and Average Travel Time Calculations:

After successfully establishing the flow rate distributions and congested flow densities for links at different time intervals we can use the congested flow densities for all the links to determine the congested speeds and average travel time for all the links. Figure C.3 represents the interval flow rate mapped in a space-time graph for better understanding and density estimations. It is important to note that flow values obtained from RNNGB for congestion (Q_{cong}) are to be considered further only for the flow equations.

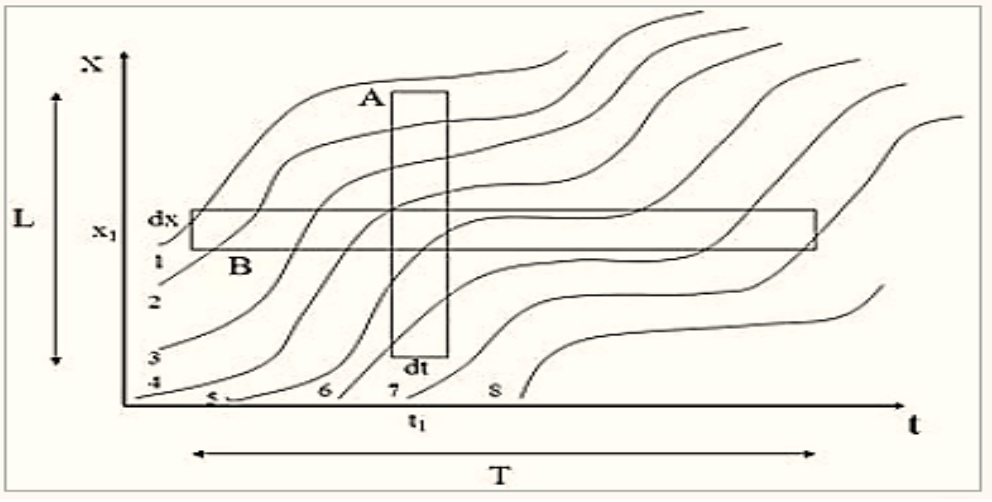


Figure C.3 Flow Rate Space-time Diagram for a single link in a node considering one direction only.

Considering the space-time critical congestion flow (Q_{cong}) graph for a single link, the corresponding space-time mean congestion link density $(\rho_{cong})_{mean}$ is given by equation C.1.

$$(\rho_{cong})_{mean} = \sum_{i=1}^{n} d_i(B) / nTL \qquad \text{(C.1)}$$

Where n is the numbers of links in a node being considered, B is space-time area shown in figure C.3, T is the total time indexes being considered and L corresponds to the total link lengths. Length L is considered from the point of the data sampling sensor close to the node of interest to its farthest end installed sensor of the same link. Based on the mean congestion density we can calculate the corresponding congestion speeds ($\overline{v}_{cong}$) and average travel times for the length L of the links as well as given by equations C.2 and C.3 respectively.

$$\overline{v}_{cong} = {\overline{q}_{cong}}/{\overline{\rho}_{cong}} \tag{C.2}$$

$$(t)_{mean} = {d}/{\overline{v}_{cong}} \tag{C.3}$$

Where d is the average distance that the user travels through the considered study area. The dependence of bottleneck and effects of bottleneck propagates from links to nodes and finally to the patches.

Equation C.4 is the general Gaussian Mixture Distribution estimation for a function where M is the number of components and $\beta_m\,(.|\mu_{m,c_m})$ is the $m-\text{th}$ Gaussian distribution term and μ_{m,c_m} represents the vectors of means values and the covariance matrix respectively. These parameters are estimated using the expectation-maximum (EM) algorithm. The cause links are the ones being used for the flow estimations of the effected ones (effect links) that is under consideration and the final output represents the flow states for all the links in node. One link can a part of the two or more-effect links depending upon the network structure.

$$p(f_{i,t}) = \sum_{m=1}^{M} \alpha_m \beta_m \left(f_{i,t}|\mu_{m,c_m}\right) \tag{C.4}$$

C.1.3 Naïve Bayes Based Links Flow Rate Estimations:

We base our model flow bottle neck detection for the node by estimating the flow rate probability distribution for each link in a node using gaussian mixture model (GMM) for each interval index. The estimation is done based on the historical link flowrate inflows and outflows data. Using Bayesian estimation of $f_{i,t}$ for $i = 1, \dots, N$, in the traffic flow data we need to estimate the joint distribution of flow states for all the links in the node unit $\boldsymbol{f} = \{\, f_{i,T} \;:\; i = 1, \dots, L\,; T = 0, \dots N - a\,\}$, where a is the time instance or time index for which the flow state of the node is being considered. According to the Bayesian theory [103], the distribution of f for a particular node link can be expressed as in (C.5) while considering all the node links with directional flow rate data, the bold $\boldsymbol{f}$ represents the bidirectional property of the all the links:

$$p(f) = \prod_{i,t} p\,(\, f_{i,t} \mid \boldsymbol{f}_{links\,(i,t)}) \tag{C.5}$$

The basic Bayesian theory assumptions suggest that the flows rate data gathered for all the links is independent of each other at any time intervals i.e. $P(A)$ is independent of $P(B)$ and vice versa. If the each link instance is considered as an independent event then the marginal conditional distribution for a link L is given as $p(f_{i,t} \mid \boldsymbol{f}_{links\,(i,t)}) = \frac{p(\boldsymbol{f}_{i,t})}{p(\,\boldsymbol{f}_{links\,(i,t)})}$ because of the overall joint probability density or distribution $p\,(\,\boldsymbol{f}_{i,t})$ for all the links in a node.

The conditional probabilities cannot be usually considered commutative i.e. $P\,(A \mid B) \neq P(B|A)$. The conditional probabilities relation is given by naïve bayes theorem as $P\,(A|B) = {P(A\,\cap B)\,P(B)}/{P(A)} \leftrightarrow {P(B|A)}/{P(A|B)} = {P(B)}/{P(A)}$ if and only if $P(A) \approx P(B)$ i.e. that is the probability likelihood of flowrates for the links being considered are the same. Once we have calculated the likelihood probability using GMM using equation C.4 for the effect links each for different time intervals we then estimate the conditional distribution using equation C.5, where $\boldsymbol{f}_{links\,(i,t)}$ being the prior probability is obtained from GRU-NN

predictions model for each link and normalised all the links in a node. Equation C.5 is the joint distribution or probability density of all the conditional probabilities of each link on a given node provided that the normalised probabilities of all the cause links simultaneously.

C.1.4 Flow Rate Trend Analysis in Probability Distributions at Nodes:

Traffic dataset exhibits both recursive and non-recursive trends through the lot of big data gathered over a long period of time. The trend is to be looked for are the 1) hourly patterns 2) daily patterns 3) monthly and yearly patterns. Further the dataset also allows us to look for the traffic flows for the specific days of the week, public holidays, school days and non- school days.

C1.5 Initial Insights into Conservation of Travel Time Delays:

According to the proposed RNNGB model as an initial understanding the model is developed to capture the recursive and non-recursive traffic flow characteristics based on the past gathered on road data. ML model helps model determining the prior probability of the flow rate to happen at any time instance from past data. And Naïve bayes consider predicting it based on prior probability and likelihood probabilities from Gaussian mixture model. From the model we can say that change is flow rate gives rise to variably appearing bottlenecks in a single node and this accounts for the changing travel time and consequently travel delays in real-time. RNNGB model will allow not only using the data to be exploited for trends that are independent to locational and spatial influences. But instead gives model its power from generating variations in time interval based gaussian distributions considering the covariance among data. So as an initial inference we can say that travel delay is conservative in time and that it is the function of flow trends. Other words to put the inference is that traffic travel delay in one patch can affect the travel delay in other patch but the overall travel time would remain constant for a section of the road network. And if some uncertainty arises in the model this model will do good to learn those uncertainties utilising the power of gated recurrent neural networks (GRU-NNs). Whereas the overall model version can be compared with the LSTM based version as well.

Appendix D: Published Work

Empirical Formulation of Highway Traffic Flow Prediction Objective Function Based on Network Topology

Arsalan Rahi[1], Soodamani Ramalingam[2]

[1,2*] Doctoral Student, [3] Senior Lecturer

[1,3] Centre for Engineering Research, School of Engineering and Technology, University of Hertfordshire (UH), Hatfield, Hertfordshire, United Kingdom, AL10 9AB.

[2] Research Assistant at University Bus Limited (UNO), Hatfield, Hertfordshire, United Kingdom, AL10 9BS.

* Corresponding authors: arsalanrahi92@gmail.com

ABSTRACT: Accurate Highway road predictions are necessary for timely decision making by the transport authorities. In this paper, we propose a traffic flow objective function for a highway road prediction model. The bi-directional flow function of individual roads is reported considering the net inflows and outflows by a topological breakdown of the highway network. Further, we optimise and compare the proposed objective function for constraints involved using stacked long short-term memory (LSTM) based recurrent neural network machine learning model considering different loss functions and training optimisation strategies. Finally, we report the best fitting machine learning model parameters for the proposed flow objective function for better prediction accuracy.

KEY WORDS: Intelligent Transportation Systems, Machine Learning, LSTM, Flow Estimation, Hyper Parameter Optimisation.

I.INTRODUCTION

With the understanding of how intelligent transport systems (ITS) operate in a modern city, their reliance on an accurately predicted regional traffic flow and congestions changes have become inevitable. This gives rise to the quest for finding the better formula to forecast traffic parameters for as close as possible to the real world observed parameters [104]. But for ITS and transport operators to rely on traffic parametric forecasts, systems must be reliable, and this is only possible when the forecasting systems represent the traffic network on a smallest unit as offered by the network which consists of junction and the inter road links. Based on this criterion we set out the flow of this paper. We report the unique significance of the proposed system in section II, section III sheds a detailed light on what has already been done in the relevant subject in response to the advancements in machine learning technique and traffic flow predictions. Section IV list the proposed strategy along with the subsequent subsections detailing the dataset and pre-processing involved along with the system design and performance metrics are considered. Sections V and VI deal with the experimental results and their conclusion with future suggestions respectively.

II. SIGNIFICANCE OF THE SYSTEM

The paper mainly focuses on predicting the real traffic flow based on retaining the traffic network topology in the form of a dynamic objective function and using data driven time series spatiotemporal machine learning model to optimise it for more accurate highway network individual road flow predictions.

III. LITERATURE SURVEY

Traffic flow forecasting has been in research discussions for quite some time. Traffic flow forecasting can be broadly classified into two distinct categories which are as follows:

Parametric: Conventional approaches that use statistical methods for time series forecasting are normally termed as parametric model approaches. The prior knowledge of data distribution is assumed in parametric approaches. Most notable of these approaches are auto regressive integrated moving average (ARIMA) and its variant seasonal auto regressive integrated moving average (SARIMA) [45], Kalman filters [46] and exponential smoothing [47]. The problem with most of these parametric approaches is that they can effectively be employed for only one-time interval prediction and cannot predict well enough due to the stochastic and nonlinear nature of the traffic data. This can better suit short term forecasts only which are well biased towards the most recent observations in the data, thus this makes the parametric approaches incapable of handling real world trends.

Non-Parametric: A few years ago machine learning (ML) strategy based traffic parameter prediction algorithms have been utilised [48]. These data driven approaches are also termed as non-parametric approaches. The most commonly tested non-parametric approaches for spatiotemporal traffic forecasting includes the k-nearest neighbours (KNN) [49][50] and support vector regression (SVR)[47]. However, these shallow ML algorithms work in a supervised manner which makes their performance dependent upon the dataset manual feature selection criteria.

With the advancement in the ML algorithms, a bit more sophisticated dense supervised learning approach is applied for traffic predictions by using back propagation techniques in artificially connected neural networks (ANN) [25][51]. Although ANN out performs conventional linear parametric models but struggles with simple time series data learning and finding global minimum. Recently, deep recurrent neural networks (RNN) have shown some great promises for dynamic sequential modelling especially in the field of speech recognition [81][82]. Simple RNNs however suffer from gradient explosion for extra-long sequence training which results in information loss and reduced performance [83]. *Fu R et al* [42], have used the RNN variants called long short term memory (LSTM)[84] and gated recurrent units (GRU) for the traffic forecasting because of their ability to retain and pass on the information that is necessary and forget what is redundant using the output and forget gates. *Haiyang Yu et al.* proposed the spatiotemporal traffic feature learning utilising the deep convolutional LSTM network where LSTM network learns the temporal dependent patterns in the data. This makes the LSTM vanishing gradient problem during back propagation problem to fade off during error training with the usage of LSTM memory blocks and makes it able to predict with much accuracy for longer sequences [52]. For the very reason we employ LSTM in our proposed methodology to learn the temporal features whereas to keep the training and the model architecture simple we incorporate the feed forward connected ANN layer at the end for the spatial feature learning and then we train the whole architecture in a back-propagation manner. This is further discussed in the system design section.

IV. METHODOLOGY

In this Section, we represent a traffic model as consisting of a set of nodes and input-output links. The traffic flow of a set of input links will have an influence on the traffic flow of the output links. This model acts as a black box interpreting and manipulating the system inputs. A system is governed by a set of rules associated with a combination of the inputs fixed and dynamic states mapped to outputs and represented in mathematical terms [105]. Such a system can be modelled as an objective function consisting of variable parameters is shown in figure 1.

Figure 1. A general function definition.

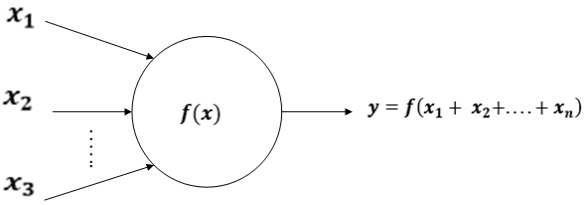
A) Definitions

We consider a highway junction spatially with inflows and outflows to be an independent system and designate each junction system as a node denoted by N. The links L serves as both the inputs as well as outputs of a node in bidirectional highway links. As an example, consider a single sample node of an actual highway junction in Hertfordshire, UK, shown in figure 2.a and its equivalent representation using the nodes and links configuration is given in figure 2.b. Further, the bidirectional arrows indicate bidirectional traffic flow of the node. Here outflow implies traffic flow moving away from the node and inflows to those moving into the node.

a) **b)**

Figure 2. a) Highway junction under consideration (Google Maps, 2018). b) Node illustration retaining junction original topology.

B) Flow Estimation Function

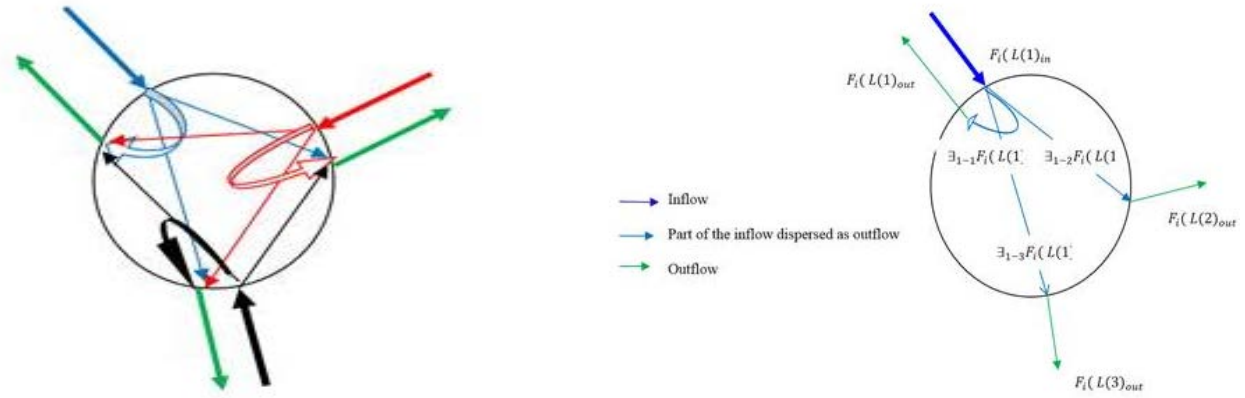


To predict the outflow of traffic for each individual link on a single node, all the incoming link flows are to be considered for the output flow forecast objective function. The outflow of a node's link is determined by the summation of inflows of individual links of the node. Figure 2.b shows that the output flow associated with a link is dependent on the inflows of every other link in the same node. The estimated traffic outflow for link $L1$ is given by equation (1) showing the

dependency of the objective function on the inflows associated with the rest of the links of the same node. Equation (2) is a more general objective function mathematical representation which describes the conservation of flow with a node system where x is the link for which the flow is being calculated and n is the total number of links on the same N. This makes the objective function retain the correlations in the flow characteristics for each individual node link when the single node is considered as a basic unit level in the traffic network.

$$L1_{out} = f(L2_{in} + L3_{in} + L1_{in}) \quad \{L1, L2, L3\ \varepsilon\ (same\ N)\} \tag{1}$$

$$L(x)_{out} = f(L(n-x)_{in}) \quad \begin{cases} x, n\ \varepsilon\ (same\ N) \\ and\ x < n \end{cases} \tag{2}$$



a) **b)**

Figure 3. a) Extension of traffic network at node i showing three links and their associated inflows and outflows. b) A simple traffic network at a node i with 3 links. It shows the distribution of incoming traffic dispersed as outgoing traffic at the node.

With reference to figure 2.b, let us consider a node *i* consisting of a set of links $L(j)$ that are associated with bi-directional traffic flow F_i. Each link $L(j)$ at the node *i* is associated with traffic inflow $F_i(L(j)_{in})$ and a corresponding outflow indicated by $F_i(L(j)_{out})$. The function for the traffic flow of links $L(j)$ considers the fact that the traffic inflow of every link contributes partially (to a certain degree) to the outflow of each of the other links at the same node. In other words, the traffic outflow of a link is a function of the traffic inflow of all the other links including its own at the node. This notion is modelled as follows:

$$F_i(L(j)_{out}) = \sum_{j=1}^{n} {F'(L(j)_{in})}\Big/{F_i(L(j)_{in})} \tag{3}$$

where ${F'(L(j)_{in})}\Big/{F_i(L(j)_{in})}$ represents a fraction of the traffic inflow that contributes to an outflow of a specific link.

As an example, consider figure 3.b, in which the circle represents a node *i* with three links. The thick blue arrow indicates the traffic inflow of link $L(1)_{in}$ that gets dispersed into the node and flows through the rest of the links. They contribute to the outflows of the rest of links including itself. This dispersion is indicated by thin blue arrows in Fig 3.b. The outflow of each of the links in shown in green arrows. The symbol $\exists_{1-j}$ indicates that part of the inflow of link $F(L(1)_{in})$ contributes to the outflow of the links $L(j)_{out}$. The sum of the traffic flow of $F(L(1)_{in})$ inside the node represented by thin blue arrows is equal to the traffic inflow of $L(1)$ represented by a thick blue arrow, at a time instant. This applies to the traffic inflow of all other links at the node as shown in figure 3.b.

We will show in the sub-section E, the use of the above model in the proposed system design.

C) Dataset Description

We perform all the experiments on the traffic flow dataset for the chosen Hatfield Hertfordshire UK area junction as shown in figure 1. The dataset is obtained from Gov.uk open datasets which contains public sector information licensed under the Open Government Licence v3.0 [93]. The used dataset contains traffic flow information for two-hour timed

aggregated intervals from start of 1st April 2015 to the end of 31st Dec 2015 for the highway roads. First three and last three raw dataset plots for links are shown in figure 4. The data is collected for the number of passing vehicles using the loop detectors installed on both the ends of the selected highway links.

D) Dataset Pre-processing

Figure 4. First and last three days of pre-processed data.

The raw dataset is taken through a series of data preprocessing steps:

Data Cleaning: As with every real world gathered data the links flow raw dataset had approximately 15% of values that were missing. Due to the ongoing trends comprising of seasonality and other environmental factors it is very important to retain the inherit trends in the traffic data. So, these values are imputed using the backward fill approach. The backward filling approach takes the value from next interval logged value and make an imputation for the previous interval. This imputing process continues until all the missing imputes are done through which all the inconsistencies are resolved.

- **Data Integration:** A total of 3252 data samples are used for each considered link. Using equation (1) they are reshaped to form an array of dimensions 3252x4. Where 4 corresponds to the links considered as given by equation (1). The sample plot from dataset containing the newly shaped $L1_{in}$ and two outflows i.e.$(F(L(1)_{out}, L(2)_{out}))$ for first and last three days of the gathered dataset are shown with twelve two-hour intervals as shown in figure 4.
- **Data Transformation:** After the data aggregation and reshaping is done it is further generalized and normalized by scaling for the minimum and maximum values among each data column. i.e. intra flow links normalization. Further the reshaped dataset is lagged by one-time interval to make it suitable for supervised training.
- **Data Reduction:** With the aim to generate the training and validation sets to train and validate the ML model we consider 20% of the original dataset as the validation set. Since it's a time series consecutive interval data the order of training and validation ensemble is very important. Therefore, we consider the tail end 20% for the validation of trained model after each training iteration.
- **Data Discretization:** Among the originally reported dataset there are twelve intervals in a twenty-four-hour time window we consider only the twelve intervals which are two hours apart each to make the ML model training not only fast but a more generalized representation of the sequential data throughout the day

E) System Design

In this section the machine learning model used to fit the pre-processed data is discussed. We discuss the architecture of LSTM and the proposed architecture based on the combination of LSTM and the NN architectures.

- **Feed Forward-Long Short-Term Memory (LSTM):** As the first part we just consider the recurrent neural network (RNN) variants called long short-term memory (LSTM) units in training for feed forward data iteration as the main time series data learners of our ML architecture along with conventional connected feed forward neural networks (NN). The hybrid LSTM-NN architecture is shown in figure 6. This part of the architecture consists of two layers of LSTM units and one layer of densely connected NN. In between each layer is an activation function. The LSTM model is defined [12] by the following set of equations:

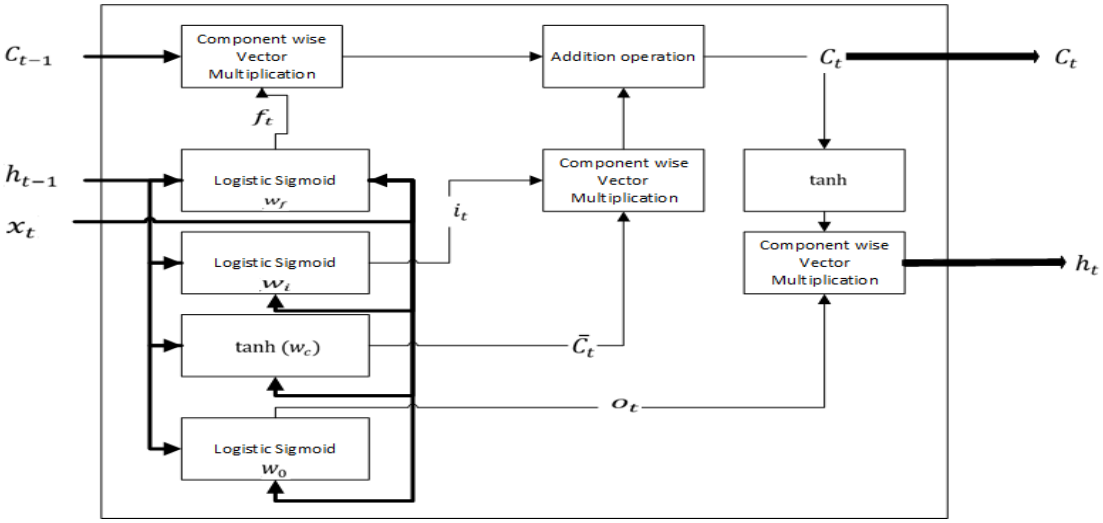


Figure 5. Structural data flow in a Long Short-Term Memory (LSTM) unit [11].

$$f_t = \sigma\left(w_f.[h_{t-1}, x_t] + b_f\right) \quad (4), \qquad i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i) \quad (5),$$

$$\bar{C}_t = \tanh(w_c.[h_{t-1}, x_t] + b_c) \quad (6), \qquad C_t = f_t \otimes C_{t-1} + i_t \otimes \bar{C}_t \quad (7),$$

$$o_t = \sigma(w_0[h_{t-1}, x_t] + b_0) \quad (8), \qquad h_t = o_t \otimes \tanh(C_t) \quad (9).$$

LSTM's general purpose can be defined as the estimation of the conditional probability $p\left(y_1, y_2, \ldots y_{T'} \mid x_{1,}x_2, \ldots x_T\right)$ given that $(x_{1,}x_2, \ldots x_T)$ is an input sequence and $(y_1, y_2, \ldots y_{T'})$ is the corresponding output sequence. The lengths of T' and T may differ. The deep LSTM computes the conditional probability by first computing the fixed dimensional input representations v, of the input sequence, from the last hidden memory state of the LSTM layer [106]. The hidden states h_t for each individual LSTM unit is calculated as given by the equation (9). Accordingly, for the proposed objective function in

(3), standard LSTM network for the i^{th} node with internal hidden states v of corresponding inputs $\left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)$ is given by equation (10) :

$$\left(F_i(L(k)_{out})_1,\ F_i(L(k)_{out})_2, \ldots,\ F_i(L(k)_{out})_{T'} \mid \left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)\right)$$

$$= \prod_{t=1}^{T'} p\ (F_i(L(k)_{out})_t \mid v,\ F_i(L(k)_{out})_{t-1}) \qquad (10)$$

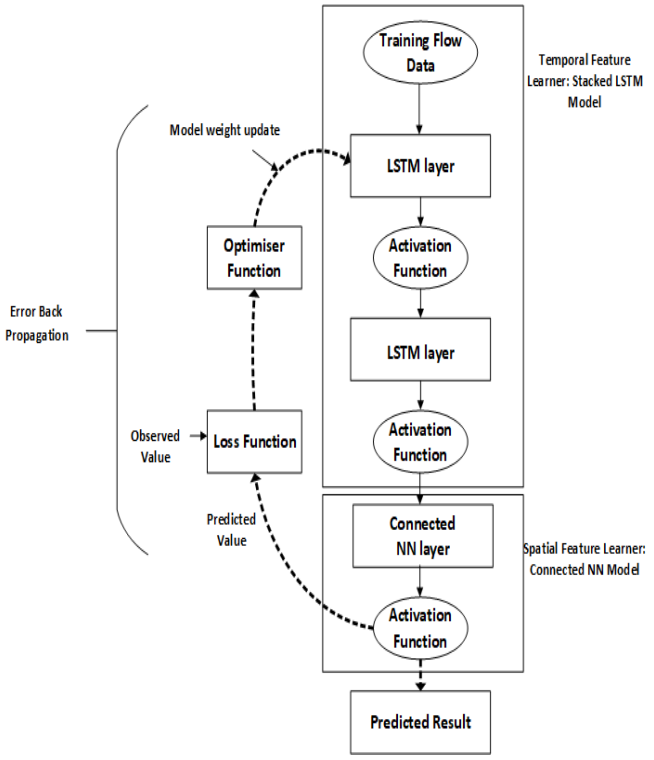


Figure 6. Proposed System Architecture

where k in the $F_i(L(k)_{out})$ represents the output, link being considered by the LSTM for the output flow conditional probability estimations. As shown in the system design (refer figure 6.) the LSTM layers are cascaded with NN layers. Equation (10) can now be interpreted for our flow problem as given by equation (11) which forms the model for traffic flow. Note that in equation (13), f_o and f_h represent the output and hidden layer activation functions respectively. H_j in equation (12) and O_k in equation (13) define the hidden layer and output layer outputs.

It is to be noted that there are two LSTM layers stacked model followed by a NN model in this architecture. We later show that the choice of the number of nodes of the hidden layers in each of these models can impact the system performance. Both models try to learn spatial and temporal features respectively.

From (10), we have the input $X_j =$

$$p\left(F_i(L(k)_{out})_1,\ F_i(L(k)_{out})_2, \ldots,\ F_i(L(k)_{out})_{T'} \,\middle|\, \left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)\right) \quad (11)$$

$$H_j = f(I_j);\quad I_j = \sum_{j=1}^{n} W_{kj} X_j \qquad (12) \qquad\qquad O_k = f(I_k);\quad I_k = \sum_{j=1}^{n} W_{kj} H_j \qquad (13)$$

Substituting (11) and (12) in (13), we get:

$$O_k = f_o\left(\sum_{j=1}^{n} W_{kj}\, f_h\left(\sum_{j=1}^{n} W_{kj}\, p\left(\begin{matrix} F_i(L(k)_{out})_1,\ F_i(L(k)_{out})_2, \ldots,\ F_i(L(k)_{out})_{T'} \mid \\ \left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \right. \\ \left. \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)\end{matrix}\right)\right)\right) \qquad (14)$$

The activation functions F_i tested for the scope of this paper are given in table 1 along with their mathematical representation. In our model the pre-processed data of shape (2602, 1, 4) with three inflows and one outflow according to equation (1) is be fed into the model and the respective link inflow and outflow values for the next time interval can be generated through the LSTM-NN. The shape dimensional values in (2602,1,4) represents the number of samples, batch number, variable features or corresponding link values, respectively. For each model iteration a separate validation set of similar shape (650, 1, 4) as of training data is used for the performance analysis measures. The final model parameters including the number of LSTMs and NNs chosen along with activation function are further discussed in the experiments section.

	Activation Function (*g*)	Mathematical Implementation
6.	sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}};\ \sigma(x) \in [0,1]$
7.	softmax	$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}};\ j = 1,2,\dots,K;\ \sigma(x)_j \in [0,1]$
8.	tanh	$\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}};\ \tanh(x) \in [-1,+1]$
9.	relu	$f(x) = \max(0,x)\,;\ f(x) \in [0,\infty)$

Nomenclature: softmax represents the normalised exponential function for multiclass logistic function flow values in our case, that makes K-dimensional vector x to have values in range [0, 1] that all add up to 1.

Table 1. Layer activation functions considered.

- **Feed Backward-Loss and Optimiser Function:** The second part of the system design considers the optimisation function and the loss function while updating the feed forward model weights before the next iteration. The iterative back-propagation allows the LSTM architecture to learn the temporal correlations amongst the intra node links whereas as the connected NN layer help learns the spatial dependencies. A set of optimisation strategies and loss functions considered in the experiments are given in table 2 & 3, respectively whose relative performances are evaluated in the process.

	Optimisation Function (*X*)	Mathematical Representation
27.	Stochastic Gradient Descent (SGD)	$w_{t+1} = w - \eta \left[\sum_{i=1}^{N} \frac{\nabla Q(w_i)_t}{\mathrm{N}}\right] + \alpha\Delta w;$
28.	Adaptive Gradient Algorithm (Adagrad)	$w_{t+1} = w_t - \frac{\eta}{\sqrt{G_t+\varepsilon}} \odot g_t$
29.	Root Mean Squared Propagated Gradient Descent (RMSprop)	$w_{t+1} = w_t - \frac{\eta}{\sqrt{E(G_t)+\epsilon}} \odot g_t$

Nomenclature: $\boldsymbol{w_i} = (\bar{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2$, η is the learning rate, α is the learning momentum factor, g_t is the iteration gradient, $G_t = \sum_{i=1}^{N} g^2{}_{t,i}$ is the diagonal.

Table 2. Optimisation Strategies considered.

	Loss Function (*J*)	Mathematical Loss Representation
1.	Mean Squared Error (L2 loss)	$MSE = \frac{1}{N}\sum_{i=1}^{N}(\,y_i - \bar{y}_i)^2$
2.	Mean Absolute Error (L1 loss)	$MAE = \frac{1}{N}\sum_{i=1}^{N} abs\,(\,y_i - \lambda(x_i))$
3.	Mean Squared Logarithmic Error	$MSLE = \frac{1}{N}\sum_{i=1}^{n}(\log(\bar{y}_i + 1) - \log(y_i + 1))^2$
4.	Poisson	$poisson = \frac{1}{N}\sum_{i=1}^{N}(\bar{y}_i - y_i) * log\,(\bar{y}_i + \varepsilon)$
5.	Cosine	$cosine = cosine(\bar{y}_i - y_i)$
6.	Cosine Proximity or Cosine Distance	$cp = -\frac{\sum_{i=1}^{N}(\bar{y}_i * y_i)}{\sqrt{\sum_{i=1}^{N}(y_i)^2} * \sqrt{\sum_{i=1}^{N}(\bar{y}_i)^2}}$
7.	Logarithmic Hyperbolic Cosine	$logcosh = \sum_{i=1}^{N}\log(\cosh(\bar{y}_i - y_i))$
8.	Hinge	$hinge = \frac{1}{N}\sum_{i=1}^{N}\max(0, m - \bar{y}_i * y_i)$
9.	Kullback Leibler Divergence	$kl = \frac{1}{N}\sum_{i=1}^{N}(y_i * \log(y_i)) - \frac{1}{N}\sum_{i=1}^{N}(y_i * \log(\bar{y}_i))$

*Nomenclature: $\bar{\boldsymbol{y}}_i$ is the model last layer predicted value, $\boldsymbol{y_i}$ is the actual value, $\boldsymbol{\lambda}$ is the rate of absolute change set initially **m** is the threshold margin value already set for the hinge cost function.*

Table 3. Cost / loss estimation functions considered.

A shallow LSTM-NN architecture is effective in capturing the spatio-temporal dependencies on node level with defined topological link order and this can be extended to further inter connected nodes and links. Thus, in the next section we perform experiments with varying parameters including loss function and activations which are given in table 1 and 2 respectively. The experimental run involves searching for the best parameters for both the two defined stages from that we hope to analyse the performance measures for best data driven objective function determination.

F) Performance Metrics

For the performance measure for the proposed model, we consider the root mean square error (RMSE) as widely used by researcher's community in the field of machine learning. We consider validation RMSE as our major model performance indicator. The formula given in equation (15) is the mathematical representation of RMSE.

$$RMSE = \left\{\frac{1}{N}\sum_{n=1}^{N}(|\overline{y_n} - y_n|)^2\right\}^{1/2} \quad (15)$$

where in equation (15), N represents the number of validation samples used for the error calculation, $\overline{y_n}$ is the predicted output and y_n is the original value observed by model.

V. EXPERIMENTAL RESULTS

In this section we show how the hyper-parameters of the proposed LSTM-NN network are optimised based on the network's performance using the Hatfield node junction data. The following notation is observed. Let

$g \rightarrow A$ctivation Function, $X \rightarrow$ Optimisation Function, $J \rightarrow$ Loss Function, $n \rightarrow$ Number of nodes in hidden layer, $\boldsymbol{J_{opt}}$ and $\boldsymbol{X_{opt}}$ are the optimised output values of J and X respectively.

Hyper-parameters optimisation is carried as a three-stage process whereby we first determine optimal values of J and X using Algorithm A. These optimal parameters are in turn used by Algorithm B to determine the optimal parameters of n. It is worth noting that n takes only 2 sets of values in Algorithm A to determine $\boldsymbol{J_{opt}}$ and $\boldsymbol{X_{opt}}$ whereas in principle several other combinations exist, and they are not considered at this point; instead they are optimised in the second stage using Algorithm B.

A) Finding Best Fitting Loss and Optimisation Functions

Firstly, we compare the performance measure by changing the loss functions Jalong with the optimisation techniques X. We compare nine different loss functions for our data model including the most common ones majorly used in data regression problems like mean square error, mean absolute error, mean squared logarithmic error, Poisson, cosine and the probability based logarithmic hyperbolic cosine, cosine proximity, hinge and lastly the cross entropy based Kullback-Leibler divergence. The best performing loss function $\boldsymbol{J_{opt}}$ is declared based on the minimum RMSE error.

The hybrid LSTM-NN model training is carried out by two different layer configurations of n = (35, 5, and 5) and (45, 20, 20) at different instances each with three different optimisers used. Each layer configuration corresponds to the (LSTM-layer1, LSTM-layer2, and NN-layer) respectively. But for each of them the activation function g for the respective layers was taken as constant i.e. (sigmoid, sigmoid, sigmoid) for the loss function versus the optimiser function performance

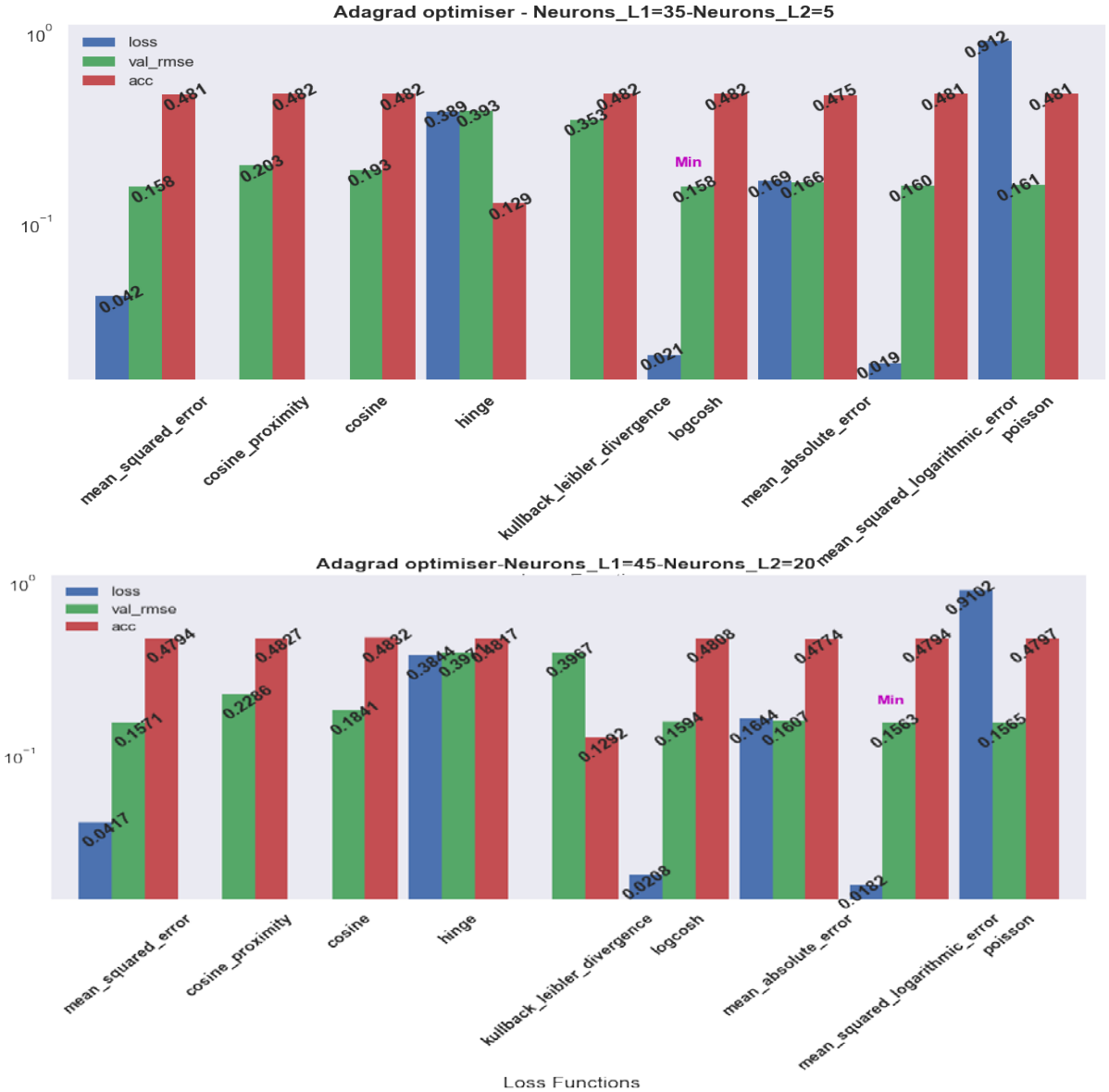
test. The optimiser we used are the simple stochastic gradient descent (SGD), to the adaptive gradient algorithm (Adagrad) and running average-based root mean squared propagated gradient descent (RMSprop). Performance bar graphs in figure 7 shows that the minimum validation RMSE is achieved by the RMSprop among all the three optimiser which indeed is true in our case as the learning rate of the optimiser better adapts to the running average of time series then just simply considering the previous time interval. And the least RMSE is achieved by the (45, 20, 20) layer configuration. The training loss, accuracy and validation RMSE for each of the instances are shown in figure 7. All three metrics reflect one and the same result.

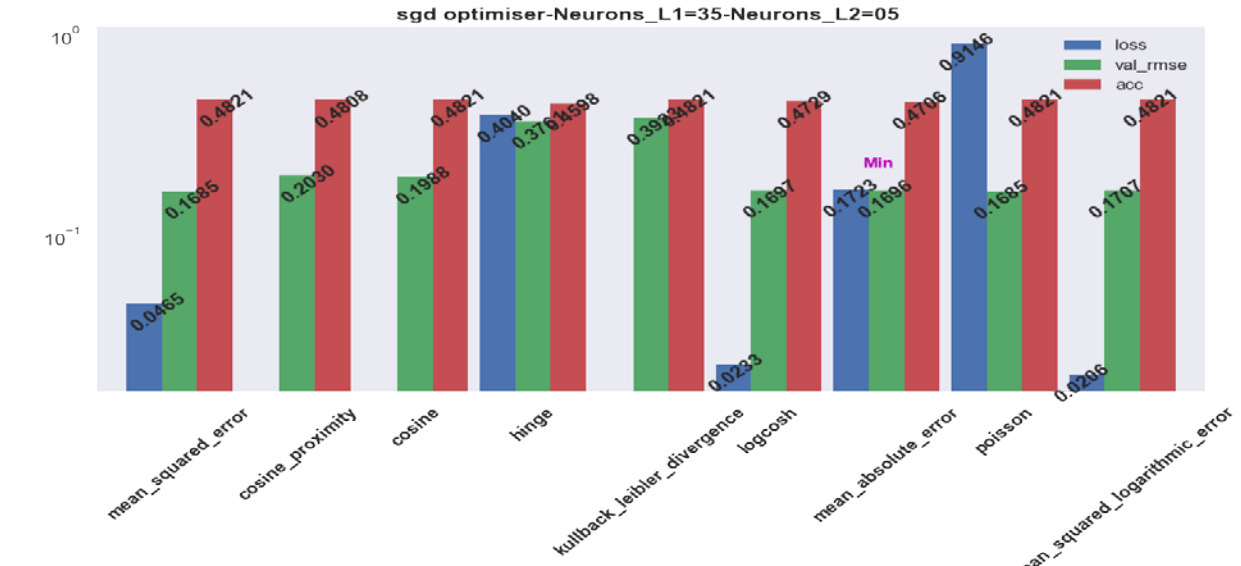
Algorithm A: Hyper parameter Optimization - Loss (J_{opt},) and Optimisation Functions (X_{opt})

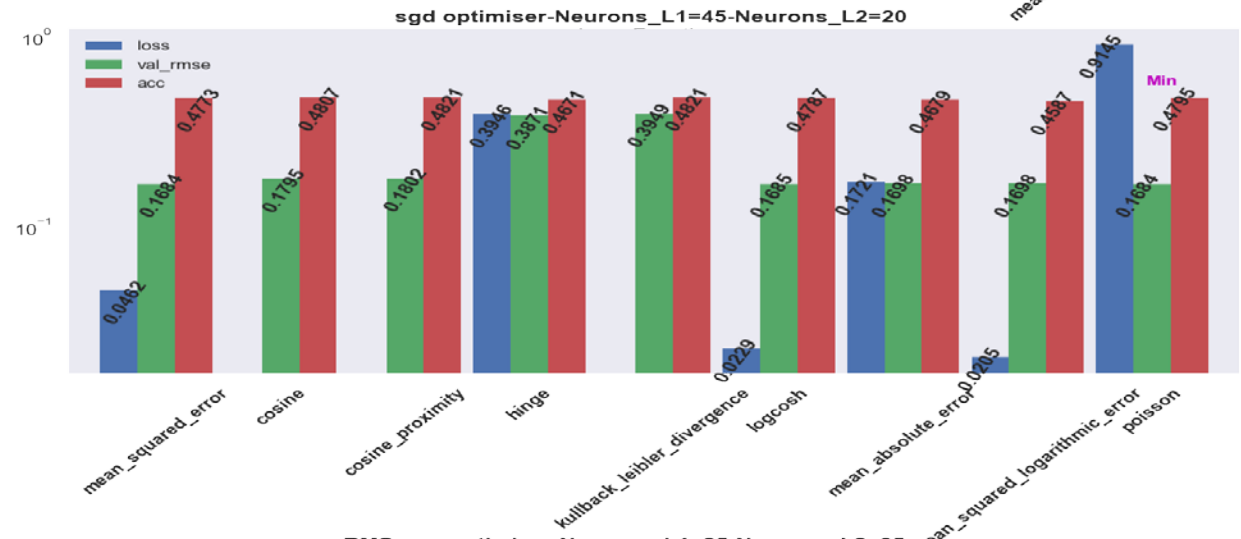
function Hyper parameter Optimisation $(J, X, g, n, J_{opt}, X_{opt})$

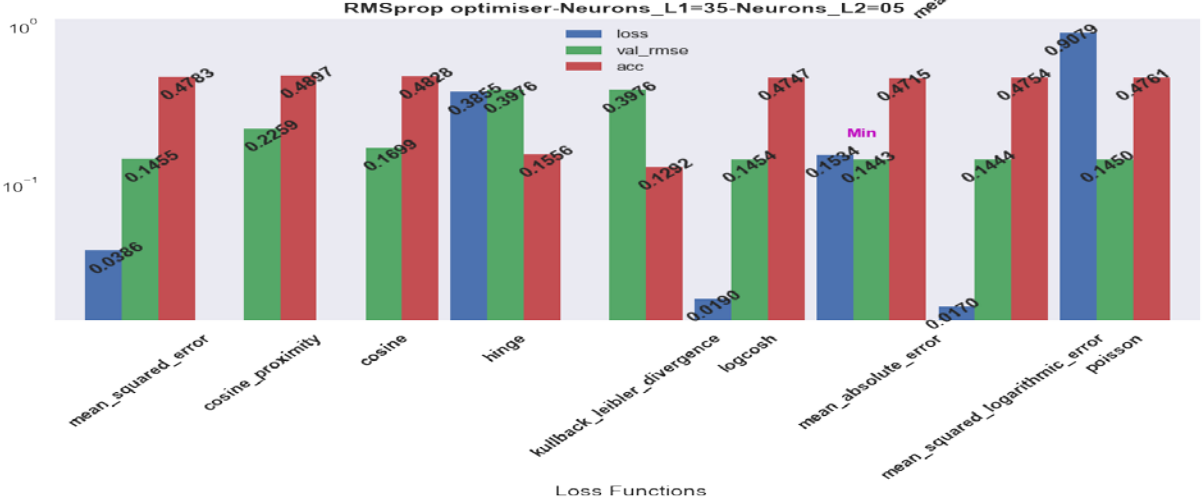
1. **Input:** *Performance evaluate loss functions* ***J (dimensionality=9)***
2. **Compute** RMSE
3. **Output:** J_{opt}

4. **Input:** $J_{opt}, g\ (dimensionality = 4), n\ (dinesionality = 2), X(dimensionality = 3)$




sgd optimiser-Neurons_L1=35-Neurons_L2=05
loss
val_rmse
acc
Min
mean_squared_error
cosine_proximity
cosine
hinge
kullback_leibler_divergence
logcosh
mean_absolute_error
poisson
mean_squared_logarithmic_error


sgd optimiser-Neurons_L1=45-Neurons_L2=20
loss
val_rmse
acc
Min
mean_squared_error
cosine
cosine_proximity
hinge
kullback_leibler_divergence
logcosh
mean_absolute_error
mean_squared_logarithmic_error
poisson


RMSprop optimiser-Neurons_L1=35-Neurons_L2=05
loss
val_rmse
acc
Min
mean_squared_error
cosine_proximity
cosine
hinge
kullback_leibler_divergence
logcosh
mean_absolute_error
mean_squared_logarithmic_error
poisson
Loss Functions

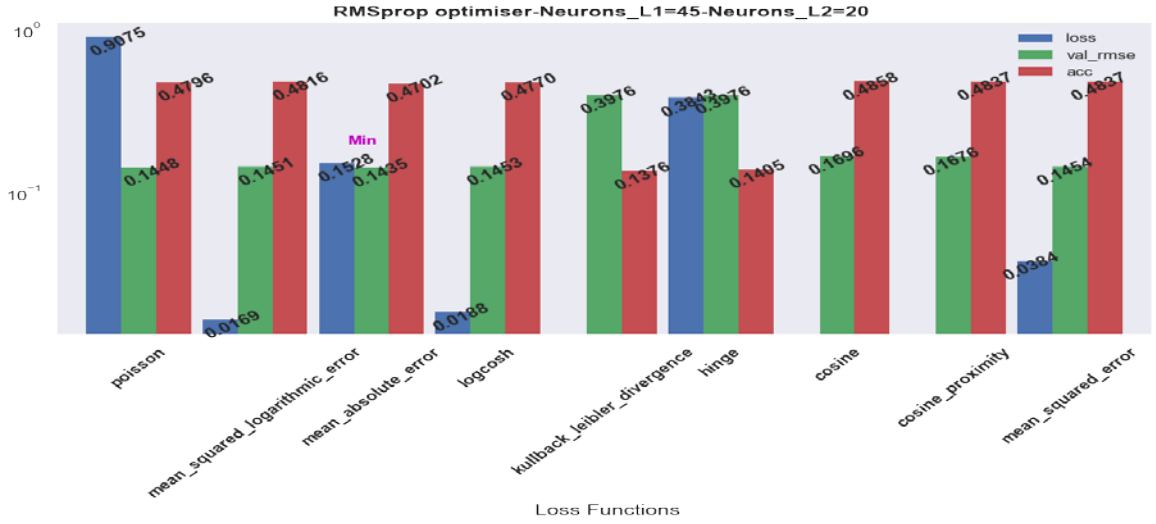


Figure 7. Performance report comparing three different optimisation techniques versus the loss functions with two different layer configurations.

B) Layers LSTM Units

Algorithm B: Hyper parameter Optimisation – Number of Hidden Layer Nodes, n_i of LSTM Layers

function Hyper parameter Optimisation $(J_{opt}, X_{opt}, n_{opt})$

5. **Input:** *Performance evaluate Number of Hidden Layer Nodes,* ***n (dimensionality =20)***
6. **Compute RMSE**
7. **Output:** n_{opt}

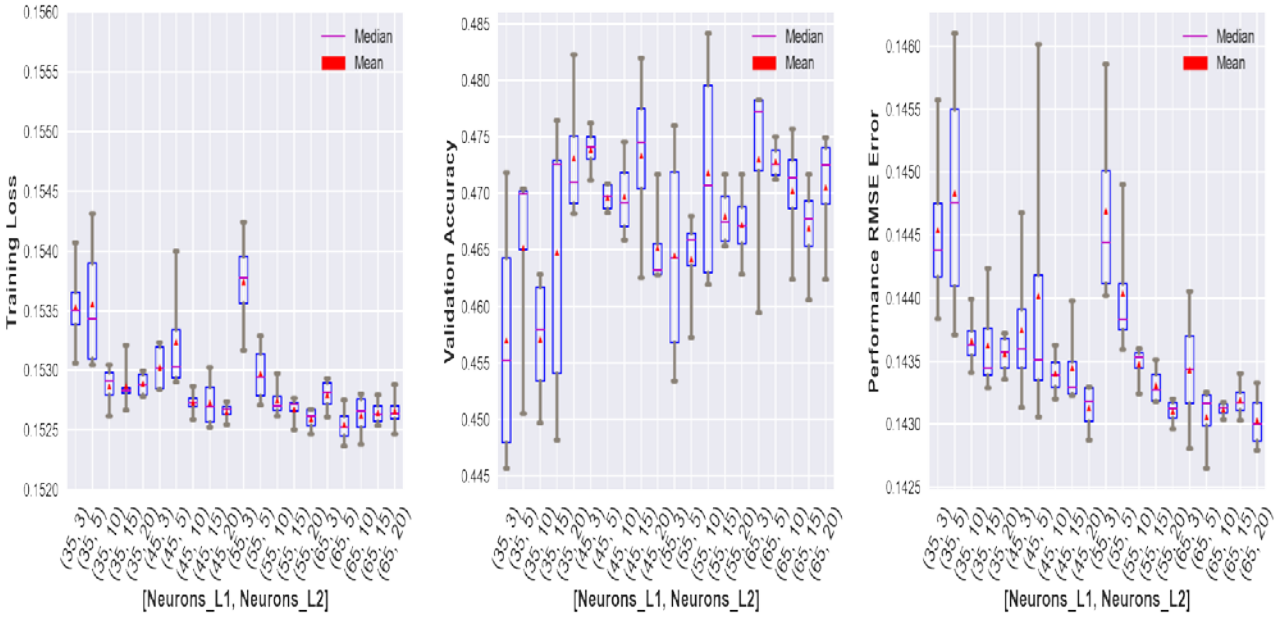


Figure 8. Performance Evaluation of Hyper-parameter $\boldsymbol{n}$ *(Layer1, Layer2)*

In the second stage, we consider experimenting with n , the varied number of layer one, layer two LSTM units and NN layers. Using the optimal performing (X_{opt}) RMSprop optimisation technique and the best performing (J_{opt}) Mean Absolute Error (MAE) as a loss function we quest for the best suited LSTM layer unit numbers (n_{opt}) that minimises the validation RMSE exhibited by the model. The final performance plot in the form of boxplot with mean and median of four iteration runs made with each configuration is shown in figure 8. As before, any of the performance metrics may be used, but we show all three metrics for better clarity. Figure 8 has a notation [Neurons_L1, Neurons_l2] in which Neurons_L1 refers to the number of units in the hidden layers of both LSTM layers and Neurons_L2 refers to that of the NN layer as shown in the system design in figure 6.

C) Effect of Layer Activation functions

In the third stage, we analyse the architecture based on the choice of different layer activation functions, g. From Algorithms A and B, we consider the determined optimum performing RMSprop optimiser (X_{opt}), MAE as a loss function (J_{opt}) and n_{opt} = (65,65, 5) as the chosen final layer LSTM unit configuration. This is because the (65,65,5) combination exhibits the lowest mean validation RMSE out of all the configurations tested as shown in figure 8. Algorithm C tests all the combination of layer activation functions from table 1. We find that the least validation RMSE of 0.1398 is exhibited by the relu-tanh-relu configuration as shown in figure 9. The experimental result heat map in figure 8 shows that tanh does generalise the objective function well enough compared to softmax and sigmoid. This is because tanh as given in table 1 has a range of [-1, 1] and the negative first derivative is not a constant which is the property common to both sigmoid and softmax activation functions.

Algorithm C: Hyper parameter Optimisation – Activation Function (g)

function Hyper parameter Optimisation $(J, X, n, J_{opt}, X_{opt}, g_{opt})$

1. **Input:** *Performance evaluate activation functions* g *(dimensionality=4)*
2. **Compute** RMSE
3. **Output:** g_{opt}

Figure 9. Performance Comparison of activation function combinations.

VI. CONCLUSION AND FUTURE WORK

To forecast the traffic flow in transportation networks several methods have been proposed by many researchers. During the survey it is seen that the flow prediction using conventional statistical and latest machine learning techniques starting from simple KNN to the latest deep ANN and time series LSTMs are highly effective in determining the spatiotemporal features which are crucial to traffic flow forecasting. In this paper we showed the spatiotemporal flow data remodelling in the form of topological objective function and exhibited the performance comparison of LSTM-NN with architecture parameter tunings. LSTM and ANN learns the temporal and spatial features respectively. The network is simple and fast enough for online data learning with dedicated geographical junction weight matrices for future training models. Future recommendations might include the local weather and incident data in combination with the objective function.

REFERENCES

[1] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, p. 78, 2012.

[2] R. Gupta and C. Pathak, "A machine learning framework for predicting purchase by online customers based on dynamic pricing," *Procedia Comput. Sci.*, vol. 36, no. C, pp. 599–605, 2014.

[3] R. C. Staudemeyer and C. W. Omlin, "Extracting salient features for network intrusion detection using machine learning methods," *South African Comput. J.*, vol. 52, no. July, pp. 82–96, 2014.

[4] M. Rabbani, R. Khoshkangini, H. S. Nagendraswamy, and M. Conti, "Hand Drawn Optical Circuit Recognition," *Procedia Comput. Sci.*, vol. 84, pp. 41–48, 2016.

[5] B. van Riessen, R. R. Negenborn, and R. Dekker, "Real-time container transport planning with decision trees based on offline obtained optimal solutions," *Decis. Support Syst.*, vol. 89, pp. 1–16, 2016.

[6] A. Verikas, A. Gelzinis, and M. Bacauskiene, "Mining data with random forests: A survey and results of new tests," *Pattern Recognit.*, vol. 44, no. 2, pp. 330–349, 2011.

[7] M. Schuh, J. Sheppard, S. Strasser, R. Angryk, and C. Izurieta, "An IEEE standards-based visualization tool for knowledge discovery in maintenance event sequences," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 7, pp. 30–39, 2013.

[8] A. S. Ahmad *et al.*, "A review on applications of ANN and SVM for building electrical energy consumption forecasting," *Renew. Sustain. Energy Rev.*, vol. 33, pp. 102–109, 2014.

[9] A. Anwar, T. Nagel, and C. Ratti, "Traffic origins: A simple visualization technique to support traffic incident analysis," *IEEE Pacific Vis. Symp.*, pp. 316–319, Mar. 2014.

[10] J. W. C. van Lint, "Reliable Travel Time Prediction for Freeways," te Delft, 2004.

[11] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, 2015.

[12] C. Hsu and F. Lian, "A Case Study on Highway Flow Model Using 2-D Gaussian Mixture Modeling," in *Proceedings of the 2007 IEEE Intelligent Transportaion Systems Conference*, 2007, pp. 790–794.

[13] S. Oh, Y. J. Byon, K. Jang, and H. Yeo, "Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach," *Transp. Rev.*, vol. 35, no. 1, pp. 4–32, 2015.

[14] C. Goves, R. North, R. Johnston, and G. Fletcher, "Short Term Traffic Prediction on the UK Motorway Network Using Neural Networks," *Transp. Res. Procedia*, vol. 13, pp. 184–195, 2016.

[15] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction in heterogeneous condition using artificial neural network," *Transport*, vol. 30, no. 4, pp. 397–405, 2015.

[16] Z. Abdelhafid, F. Harrou, and Y. Sun, "An Efficient Statistical-based Approach for Road Traffic Congestion Monitoring," in *5th Int. Conf. Electr. Eng. - Boumerdes*, 2017, vol. 2017–Janua, pp. 1–5.

[17] R. Li and G. Rose, "Incorporating uncertainty into short-term travel time predictions," *Transp. Res. Part C Emerg. Technol.*, vol. 19, no. 6, pp. 1006–1018, 2011.

[18] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting : Where we are and where we ' re going," *Transp.*

Res. Part C, vol. 43, pp. 3–19, 2014.

[19] C. Siripanpornchana, S. Panichpapiboon, and P. Chaovalit, "Effective variables for urban traffic incident detection," *IEEE Veh. Netw. Conf. VNC*, vol. 2016–Janua, pp. 190–195, Dec. 2016.

[20] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mob. Comput.*, vol. 12, no. 11, pp. 2289–2302, 2013.

[21] Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bajgain, "Improved Deep Hybrid Networks for Urban Traffic Flow Prediction Using Trajectory Data," *IEEE Access*, vol. 6, pp. 31820–31827, 2018.

[22] G. Fusco, C. Colombaroni, and N. Isaenko, "Short-term speed predictions exploiting big data on large urban road networks," *Transp. Res. Part C Emerg. Technol.*, vol. 73, pp. 183–201, 2016.

[23] F. Schimbinschi, L. Moreira-Matias, V. X. Nguyen, and J. Bailey, "Topology-regularized universal vector autoregression for traffic forecasting in large urban areas," *Expert Syst. Appl.*, vol. 82, pp. 301–316, Oct. 2017.

[24] F. Su, H. Dong, L. Jia, Y. Qin, and Z. Tian, "Long-term forecasting oriented to urban expressway traffic situation," *Adv. Mech. Eng.*, vol. 8, no. 1, pp. 1–16, 2016.

[25] S. Oh, Y. Kim, and J. Hong, "Urban Traffic Flow Prediction System Using a Multifactor Pattern Recognition Model," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2744–2755, 2015.

[26] Z. Yuan and C. Tu, "Short-term Traffic Flow Forecasting Based on Feature Selection with Mutual Information," in *Materials Science, Energy Technology, and Power Engineering I AIP Conf. Proc.*, 2017, vol. 020179, no. 1, pp. 1–9.

[27] A. Zeroual, N. Messai, S. Kechida, and F. Hamdi, "A piecewise switched linear approach for traffic flow modeling," *Int. J. Autom. Comput.*, vol. 14, no. 6, pp. 729–741, 2017.

[28] Q. Li, S. Li, and Y. Wang, "Traffic incident data analysis and performance measures development," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. 086, pp. 65–69, 2007.

[29] J. Wang, X. Li, S. S. Liao, and Z. Hua, "A Hybrid Approach for Automatic Incident Detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1176–1185, 2013.

[30] R. Kalsoom and Z. Halim, "Clustering The Driving Features Based On Data Streams," *IEEE*, pp. 89–94, Dec. 2013.

[31] H. Nguyen, C. Cai, and F. Chen, "Automatic classification of traffic incident's severity using machine learning approaches," *IET Intell. Transp. Syst.*, vol. 11, no. 10, pp. 615–623, Dec. 2017.

[32] C. E. L. Hatri and J. Boumhidi, "Fuzzy deep learning based urban traffic incident detection," *2017 Intell. Syst. Comput. Vis.*, pp. 1–6, Apr. 2017.

[33] J. Guo, Z. Liu, W. Huang, Y. Wei, and J. Cao, "Short-term traffic flow prediction using fuzzy information granulation approach under different time intervals," *IET Intell. Transp. Syst.*, vol. 12, no. 2, pp. 143–150, 2018.

[34] M. M. Rahman, S. C. Wirasinghe, and L. Kattan, "Analysis of bus travel time distributions for varying horizons and real-time applications," *Transp. Res. Part C Emerg. Technol.*, vol. 86, no. December 2017, pp. 453–466, 2018.

[35] R. Fernandez and R. Planzer, "On the capacity of bus transit systems," *Transp. Rev.*, vol. 22, no. 3, pp. 267–293, 2002.

[36] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic Flow Prediction with Big Data: A Deep Learning Approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2015.

[37] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS One*, vol. 10, no. 3, 2015.

[38] J. Y. Ahn, E. Ko, and E. Kim, "Predicting Spatiotemporal Traffic Flow Based on Support Vector Regression and Bayesian Classifier," *2015 IEEE Fifth Int. Conf. Big Data Cloud Comput.*, pp. 125–130, 2015.

[39] X. Ma, Z. Dai, Z. He, J. Ma, Y. Y. Wang, and Y. Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors (Switzerland)*, vol. 17, no. 4, p. 818, Apr. 2017.

[40] R. Al Mallah, A. Quintero, and B. Farooq, "Distributed Classification of Urban Congestion Using VANET," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2435–2442, Sep. 2017.

[41] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, "Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3204–3217, 2017.

[42] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *Proc. - 2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2016*, pp. 324–328, 2017.

[43] G. Yang, Y. Wang, H. Yu, Y. Ren, and J. Xie, "Short-term traffic state prediction based on the spatiotemporal features of critical road sections," *Sensors (Switzerland)*, vol. 18, no. 7, 2018.

[44] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "DeepTransport: Learning Spatial-Temporal Dependency for Traffic Condition

Forecasting," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018–July, pp. 1–8, 2018.

[45] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *Eur. Transp. Res. Rev.*, vol. 7, no. 3, pp. 1–9, 2015.

[46] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 50–64, 2014.

[47] M. T. Asif *et al.*, "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, 2014.

[48] J. Xin and S. Chen, "Bus Dwell Time Prediction Based on KNN," *Procedia Eng.*, vol. 137, pp. 283–288, 2016.

[49] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. Part C Emerg. Technol.*, vol. 62, pp. 21–34, 2016.

[50] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting," *Neurocomputing*, vol. 179, pp. 246–26, 2016.

[51] J. Amita, S. S. Jain, and P. K. Garg, "Prediction of Bus Travel Time Using ANN: A Case Study in Delhi," *Transp. Res. Procedia*, vol. 17, no. December 2014, pp. 263–272, 2016.

[52] X. Ma, Z. Tao, Y. Y. Wang, H. Yu, and Y. Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. Part C Emerg. Technol.*, vol. 54, pp. 187–197, 2015.

[53] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors (Switzerland)*, vol. 17, no. 7, pp. 1–16, 2017.

[54] C.-M. Hsu, F.-L. Lian, and C.-M. Huang, "A Systematic Spatiotemporal Modeling Framework for Characterizing Traffic Dynamics Using Hierarchical Gaussian Mixture Modeling and Entropy Analysis," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1126–1135, 2014.

[55] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting," *Proc. 2017 SIAM Int. Conf. Data Min.*, pp. 777–785, 2017.

[56] W. Fan and R. B. Machemehl, *Characterizing Bus Transit Passenger Waiting Times*, vol. SWUTC/99/1, no. 1. 1999.

[57] R. Fernández, "Modelling public transport stops by microscopic simulation," *Transp. Res. Part C Emerg. Technol.*, vol. 18, no. 6, pp. 856–868, 2010.

[58] National Research Council (U.S.) *et al.*, "Guidelines for the design and location of Bus Stops," *Transit Coop. Res. Progr.*, 1994.

[59] J. B. D.B.Hess, "Waitingfor the bus," *J. Public Transp.*, vol. 7, no. 4, pp. 67–84, 2004.

[60] P. G. Furth and T. H. J. Muller, "Service Reliability and Hidden Waiting Time: Insights from Automatic Vehicle Location Data," *Transp. Res. Board*, vol. 1955, 2006.

[61] F. McLeod, "Estimating bus passenger waiting times from incomplete bus arrivals data," *J. Oper. Res. Soc.*, vol. 58, no. 11, pp. 1518–1525, 2007.

[62] N. E. Myridis, "Probability, Random Processes, and Statistical Analysis, by H. Kobayashi, B.L. Mark and W. Turin," *Contemp. Phys.*, vol. 53, no. 6, pp. 533–534, Nov. 2012.

[63] O. C. Ibe, O. A. Isijola, O. A. Isijola-Adakeja, and O. C. Ibe, "M/M/1 multiple vacation queueing systems with differentiated vacations and vacation interruptions," *IEEE Access*, vol. 2, pp. 1384–1395, 2014.

[64] G. Xin and W. Wang, "Model Passengers' Travel Time for Conventional Bus Stop," *J. Appl. Math.*, vol. 2014, pp. 1–9, Apr. 2014.

[65] D. A. Wu and H. Takagi, "M/G/1 queue with multiple working vacations," *Perform. Eval.*, vol. 63, no. 7, pp. 654–681, Jul. 2006.

[66] H. Yu, Z. Wu, D. Chen, and X. Ma, "Probabilistic Prediction of Bus Headway Using Relevance Vector Machine Regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1772–1781, Jul. 2017.

[67] Z. Yu, J. S. Wood, and V. V. Gayah, "Using survival models to estimate bus travel times and associated uncertainties," *Transp. Res. Part C Emerg. Technol.*, vol. 74, pp. 366–382, 2017.

[68] H. Yu, D. Chen, Z. Wu, X. Ma, and Y. Wang, "Headway-based bus bunching prediction using transit smart card data," *Transp. Res. Part C Emerg. Technol.*, vol. 72, pp. 45–59, 2016.

[69] B. A. Kumar, L. Vanajakshi, and S. C. Subramanian, "Bus travel time prediction using a time-space discretization approach," *Transp. Res. Part C Emerg. Technol.*, vol. 79, pp. 308–332, 2017.

[70] M. Meng, A. Rau, and H. Mahardhika, "Public transport travel time perception: Effects of socioeconomic characteristics, trip characteristics and facility usage," *Transp. Res. Part A Policy Pract.*, no. xxxx, pp. 0–1, 2018.

[71] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich, "Traveling time prediction in scheduled transportation with journey segments," *Inf. Syst.*, vol. 64, pp. 266–280, 2017.

[72] A. Comi, A. Nuzzolo, S. Brinchi, and R. Verghini, “Bus travel time variability: Some experimental evidences,” *Transp. Res. Procedia*, vol. 27, pp. 101–108, 2017.

[73] Y. Wang, Y. Zheng, and Y. Xue, “Travel time estimation of a path using sparse trajectories,” *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’14*, no. 5, pp. 25–34, 2014.

[74] B. Yang, C. Guo, and C. S. Jensen, “Travel cost inference from sparse, spatio-temporally correlated time series using markov models,” *Proc. VLDB Endow.*, vol. 6, no. 9, pp. 769–780, 2013.

[75] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik, “Dynamic route planning with real-time traffic predictions,” *Inf. Syst.*, vol. 64, pp. 258–265, 2017.

[76] L. Gasparini, E. Bouillet, F. Calabrese, O. Verscheure, B. O’Brien, and M. O’Donnell, “System and analytics for continuously assessing transport systems from sparse and noisy observations: Case study in Dublin,” *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, no. April 2015, pp. 1827–1832, 2011.

[77] B. Sun *et al.*, “An improved k-nearest neighbours method for traffic time series imputation,” *©IEEE CAC 2017*, vol. 10, no. October, pp. 7346–7351, 2017.

[78] M. Moniruzzaman, H. Maoh, and W. Anderson, “Short-term prediction of border crossing time and traffic volume for commercial trucks: A case study for the Ambassador Bridge,” *Transp. Res. Part C Emerg. Technol.*, vol. 63, pp. 182–194, 2016.

[79] Y. Duan *et al.*, “An efficient realization of deep learning for traffic data imputation,” *Transp. Res. Part C Emerg. Technol.*, vol. 72, no. 10, pp. 168–181, 2016.

[80] O. D. Cardozo, J. C. García-Palomares, and J. Gutiérrez, “Application of geographically weighted regression to the direct forecasting of transit ridership at station-level,” *Appl. Geogr.*, vol. 34, no. 4, pp. 548–558, 2012.

[81] Q. V. Le I.lSutskever, OV.inyals, “Sequence to Sequence Learning with Neural Networks,” in *Neural Information Processing Systems Conference*, 2016, pp. 1–9.

[82] L. Deng and N. Jaitly, “Deep Discriminative and Generative Models for Pattern Recognition,” pp. 1–26, 2015.

[83] G. B. Zhou, J. Wu, C. L. Zhang, and Z. H. Zhou, “Minimal gated unit for recurrent neural networks,” *Int. J. Autom. Comput.*, vol. 13, no. 3, pp. 226–234, 2016.

[84] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, “Efficient Processing of Deep Neural Networks: A Tutorial and Survey,” *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[85] K. Yin, W. Wang, X. Bruce Wang, and T. M. Adams, “Link travel time inference using entry/exit information of trips on a network,” *Transp. Res. Part B Methodol.*, vol. 80, pp. 303–321, 2015.

[86] F. N. Savas, “Forecast Comparison of Models Based on SARIMA and the Kalman Filter for In ation,” 2013.

[87] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting , Second Edition Springer Texts in Statistics*. 2003.

[88] V. N. Vapnik, *Statistical learning theory*. 1998.

[89] Geoffrey E. Hinton, “Training Products of Experts by Minimizing Contrastive Divergence,” London, 2002.

[90] S. Hochreiter and J urgen Schmidhuber, “LONG SHORT-TERM MEMORY,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[91] T. M. Units, “National Traffic Information Service DATEX II Service,” 2018.

[92] DfT, “Road traffic statistics,” pp. 1–13, 2014.

[93] Highways England, “Highways England – Data.gov.uk – Journey Time and Traffic Flow Data April 2015 onwards – User Guide,” no. April, pp. 1–14, 2015.

[94] A. Rahi and S. Ramalingam, “Empirical Formulation of Highway Traffic Flow Prediction Objective Function Based on Network Topology,” *Int. J. Adv. Res. Sci. Eng. Technol.*, vol. 5, no. November, 2018.

[95] D. Zhang and M. R. Kabuka, “Combining Weather Condition Data to Predict Traffic Flow: A GRU Based Deep Learning Approach,” in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, 2017, pp. 1216–1219.

[96] Y. Jia, J. Wu, and M. Xu, “Traffic flow prediction with rainfall impact using a deep learning method,” *J. Adv. Transp.*, vol. 2017, 2017.

[97] M. Shardlow, “An Analysis of Feature Selection Techniques,” *Studentnet.Cs.Manchester.Ac.Uk*, pp. 1–7, 2007.

[98] D. A. Dickey, *Stationarity Issues in Time Series Models*. .

[99] W. Fan and Z. Gurmu, “Dynamic Travel Time Prediction Models for Buses Using Only GPS Data,” *Int. J. Transp. Sci. Technol.*, vol. 4, no. 4, pp. 353–366, 2015.

[100] Y. Liu and H. Wu, “Prediction of Road Traffic Congestion Based on Random Forest,” *2017 10th Int. Symp. Comput. Intell. Des.*, pp. 361–364, 2017.

[101] S. Sun, C. Zhang, and G. Yu, “A Bayesian network approach to traffic flow forecasting,” *Intell. Transp. Syst. IEEE Trans.*, vol. 7, no. 1, pp. 124–132, 2006.

[102] A. Pascale and M. Nicoli, “Adaptive Bayesian network for traffic flow prediction,” *2011 IEEE Stat. Signal Process. Work.*, pp. 177–180, 2011.

[103] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern classification (2nd edition),” in *John Wiley & Sons, Inc*, no. 2nd ed., 2000.

[104] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Short-term traffic forecasting: Where we are and where we’re going,” *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 3–19, 2014.

[105] W. Feng, Wei Feng, W. Feng, and Wei Feng, “PDXScholar Analyses of Bus Travel Time Reliability and Transit Signal Priority at the Stop-To-Stop Segment Level,” 2014.

[106] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” 2014.

AUTHOR’S BIOGRAPHY

Arsalan Rahi graduated in Electronics Engineering from Ghulam Ishaq Khan Institute (GIKI) Institute Pakistan in 2014.He finished his MSc in Embedded Intelligent Systems from Hertfordshire University (UH) United Kingdom in 2015. He is now a PhD candidate in Biometrics and Media Processing department in UH since 2016 and working as a data scientist at University Bus Limited (UNO). Field of interest includes smart transport management systems, IOT, Artificial Intelligence, data analytics with research interests lies in the latest machine learning algorithms implementations.

Dr Soodamani Ramalingam is a Senior Lecturer in the School of Engineering and Technology, University of Hertfordshire since 2006. She has several years of academic and research experience in the UK, Singapore and Melbourne. She received her PhD(CSE) award from the University of Melbourne, Australia in 1997 and her M.E.(CS) and B.E.(ECE) degrees from PSG College of Technology, Bharathiar University in India. Her research expertise is in Computer Vision and Machine Learning, Biometrics, Image Processing and Fuzzy Logic. Applications areas include Automatic Number Plate Recognition ANPR), 3D Face Recognition and Intelligent Transportation Systems and Energy. She has over 65 international conference and 30 journal publications in related areas of research. She is a member of IEEE and Biometrics Institute (UK).