Short-Term Traffic Prediction Using Long Short-Term Memory Neural Networks

Zainab Abbas*, Ahmad Al-Shishtawy†, Sarunas Girdzijauskas*†, Vladimir Vlassov*
*KTH Royal Institute of Technology, Stockholm, Sweden. Email: {zainabab, sarunasg, vladv}@kth.se
†RISE SICS, Stockholm, Sweden. Email: {ahmad.al-shishtawy, sarunas.girdzijauskas}@ri.se

***Abstract*—Short-term traffic prediction allows Intelligent Transport Systems to proactively respond to events before they happen. With the rapid increase in the amount, quality, and detail of traffic data, new techniques are required that can exploit the information in the data in order to provide better results while being able to scale and cope with increasing amounts of data and growing cities. We propose and compare three models for short-term road traffic density prediction based on Long Short-Term Memory (LSTM) neural networks. We have trained the models using real traffic data collected by Motorway Control System in Stockholm that monitors highways and collects flow and speed data per lane every minute from radar sensors. In order to deal with the challenge of scale and to improve prediction accuracy, we propose to partition the road network into road stretches and junctions, and to model each of the partitions with one or more LSTM neural networks. Our evaluation results show that partitioning of roads improves the prediction accuracy by reducing the root mean square error by the factor of 5. We show that we can reduce the complexity of LSTM network by limiting the number of input sensors, on average to 35% of the original number, without compromising the prediction accuracy.**

***Keywords*-LSTM; neural networks; traffic prediction**

I. Introduction

Smooth road traffic flow in urban cities is an ongoing research challenge as the demand for the road infrastructure increases faster than the speed at which cities can expand it. The increase in demand leads to traffic congestion that has a direct negative impact on the society in many aspects such as reduced traffic safety, increased pollution, wasted fuel and time, increased cost for businesses, etc.

Road transport administrations of developed large cities acquire real-time traffic data from multiple sources such as infrastructure sensors, mobile data, bluetooth sensors, and traffic cameras and apply state-of-the-art techniques to monitor and analyze road traffic. These systems and techniques altogether became known as Intelligent Transportation System (ITS) which aims at providing innovative solutions to tackle the traffic management problem and achieve smarter utilization of the transport network.

Short-term traffic prediction is a vital component in any ITS. Being able to accurately predict the state of the traffic in the near future enables ITS to be proactive rather than reactive by actively mitigating potential problems before they happen.

The common schools of thought on studying traffic prediction involve traffic flow theory-based models [1], [2], statistical techniques [3]–[5] that commonly use regression [6]–[8] and neural networks [9]. One of the limitations of conventional statistical methods is the increase in complexity when modelling spatial dependencies that involves the effect on traffic flow from the surrounding points of interest. To tackle this, multivariate methods were used that capture the effect of correlated regions of interest [10], [11]. In parallel to this, neural networks (NNs) for short-term traffic prediction were being explored [12]–[14].

Simple NNs are too shallow in structure to capture spatio-temporal data dependencies efficiently. Deep learning has proven to provide more accurate results in terms of learning the complex and deep dependencies for the traffic data [15]. For example, deep architectures were employed to predict traffic flow in [16]. Similarly, [17] used deep architectures to predict congestion. Deep Belief Networks were introduced in [18] for traffic flow prediction. Stacked encoders were used to learn traffic flow features [19] and for traffic data imputation [20]. Deep convolution NNs were used for traffic speed prediction in [21]. Authors of [22] introduced the use of Long Short-Term Memory (LSTM) Networks for traffic prediction and shown that LSTMs are more accurate compared to the other models considered in [22].

Considering these deep learning approaches, our work is related to [22]–[25], which use LSTMs. It is unique in a way that we further exploit LSTM capabilities that were not fully utilized in the current approaches. We use more fine-grained and high resolution data, which makes training of a single LSTM based model over the whole highway network challenging because the model parameters can increase significantly. Therefore, we provide a way to partition the road network and train LSTMs with data streaming from sensors in those partitions. We also reduce the complexity of our model by using only strategically important sensors for prediction.

In this paper, we take a data driven approach to provide accurate and scalable short-term traffic density prediction for Motorway Control Systems (MCS). An MCS is part of an ITS that focuses on monitoring and controlling highways in a city due to their importance in keeping a smooth flow in the city. We use data from the MCS in Stockholm which monitors major highways and provides flow and speed information per lane every minute measured from radar sensors (Figure 1) spread only around 150-400 meters apart from each other (Figure 6).

978-1-5386-7232-7/18/$31.00 ©2018 IEEE
DOI 10.1109/BigDataCongress.2018.00015

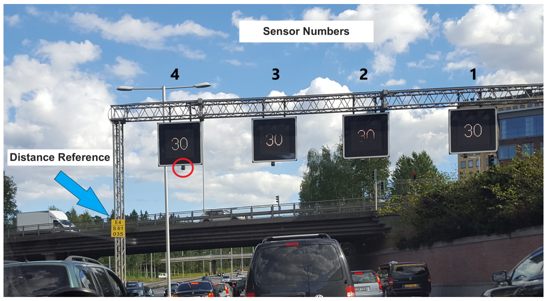


Figure 1: Traffic sensors placed on Stockholm highway.

However, even with a relatively small city as Stockholm, a deep neural network can quite rapidly get very complex as the number of inputs (sensors) increase. This complexity, in turn, might lead to infeasibility due to time and resource requirements for training, updating, and real-time prediction. We propose, evaluate, and discuss various design choices and architectures, first, to improve the accuracy of prediction, and second, to reduce the complexity (and potentially the cost) while maintaining the accuracy.

We propose a novel way of exploiting LSTM networks by partitioning the road network into smaller sections containing on average 20-30 sensors and applying LSTMs on each section. In particular, we show that after training the LSTMs with the data from all the sensors, in the operational phase our technique is capable of successfully predicting short-term traffic by using only a small fraction of the initial sensors (on average up to 35% of sensors). This allows to significantly reduce the costs for ITS by deploying only a small number of permanent sensors and relying on temporary (mobile) sensors for the training phases only, instead of deploying a dense network of sensors permanently.

The main contributions of the paper are:

- We provide different prediction models based on deep learning approach using LSTMs. These include a single sensor model that is trained only for one sensor and two multi-sensor models that take into account various adjacently placed sensors. Moreover, we compare their prediction quality and execution time.
- We show that taking into account the spatio-temporal dependencies by using neighbouring sensors in our multi-sensor models, allows improving the prediction accuracy.
- We exploit the potential of our deep neural network in the last model by training it in a way that can help us predict the traffic of an area by using only a small fraction of total sensors deployed in that area.

The rest of the paper is organized as follows: Section II explains the background work. Section III introduces our models. Section IV contains the experimental methodology. Section V explain the models in detail and the experimental results. Finally, conclusion and future work is in Section VI.

II. Background

A. Traffic Data

There exist different methods, such as mathematical and statistical models, simulations and visualization, to study, understand, and analyze road traffic in order to plan, design and operate transportation systems. The analysis can be done at a microscopic scale where individual vehicles are modelled or at a macroscopic scale where the aggregated traffic behaviour is being modelled.

The common factor of all methods is that they require measured traffic data. The main reason is that road traffic depends on the collective human behaviour, interactions, and habits which differ widely between different areas because of various reasons such as the characteristics of the road users (e.g, age, driving experience), type of vehicles (e.g., cars or trucks) and their physical properties, environmental aspects that affects behaviour (e.g., weather, road shape and type, nearby points of interests) etc. All this makes analyzing road traffic more challenging and evolving over time.

Road traffic data consists of a large number of space-time parameters. In its most basic form, it consists of traffic counters which count the number of vehicles passing at specific points (flow) on the road. Traffic data typically include other parameters such as speed, vehicle mix (e.g., car/truck ratio), road occupancy, origin-destination, vehicle trajectory. Traffic data can benefit from auxiliary data such as information about accidents, road work, events and holidays, weather, and road properties (lanes, type, speed limits).

There exists a variety of sensing techniques used to collect traffic data. Infrastructure or road-side sensors such as inductive loops and radars are used to collect macroscopic flow data at fixed points on the road. GPS and cellular network data (known as floating car data) are used to get vehicle trajectory for microscopic analysis. Bluetooth sensors and automatic number plate recognition can be used to obtain origin-destination and trip time information. Many other techniques such as audio/video based vehicle detection are also used to obtain traffic data.

Floating car data (FCD) is obtained mainly from participating passengers carrying cell phones in the vehicle. FCD can provide a good estimate of the traffic speed but might fail at providing an accurate estimate of the traffic flow and density. The main advantages of FCD are the wide coverage and small cost. Infrastructure sensors are more expensive to install and maintain and they measure data at a fixed location limiting their coverage. However, data from infrastructure sensors are more accurate and complete as they measure and count all vehicles that pass them in real-time. Because of the improved accuracy that comes at an increased cost, infrastructure sensors are typically deployed only on critical road sections such as highways. Macroscopic traffic data comes at different aggregation levels. The main parameters of the aggregation are: 1) the frequency of aggregation (e.g., flow and average speed per minute vs. per hour). 2) aggregation over lanes (i.e., data per lane or across all lanes). 3) spacing between sensors (e.g., every kilometer).

B. Elements of Traffic Flow Theory

Traffic flow theory is the study of dynamic traffic behaviour over the roads. It depends upon the driver's reaction towards different traffic conditions [26], [27]. It is a common practice to show the traffic behaviours using three traffic variable, namely: $flow\ q$ (vehicles per unit time), $density\ k$ (vehicles per unit distance) and $speed\ v$ (distance per unit time). The relation between these variables can be represented by the following equation:

$$q = k \times v \tag{1}$$

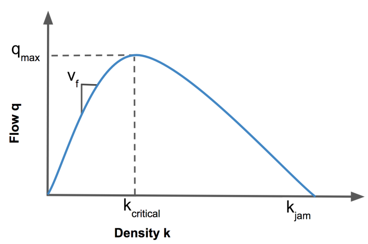


Figure 2: The fundamental traffic flow diagram.

Figure 2 plots the relation between q and k. At low density, the speed does not depend on the density, and vehicles move with the free flow speed v_f. When the density increases, the flow can reach the maximum value q_{max} based on road capacity. The density at this point is called the critical density $k_{critical}$. Beyond this, the speed decreases because it becomes difficult for vehicles to overtake. Finally, density reaches to k_{jam} where maximum vehicles that can fit the road are stuck in a traffic jam. This makes density k an important parameter to indicate congestion.

C. Long Short Term Memory Networks

Recurrent Neural Networks (RNNs) recently became popular for learning and capturing latent patterns and behaviour in the sequential data. In contrast to classic NN, the output of RNN depends not only on the current input but also on the previous state of the network, which acts as a memory. Such configuration makes RNNs naturally suitable for modelling tasks involving sequential data and time series, such as: handwriting recognition, natural language processing, speech recognition, machine translation etc. However, RNNs have major limitations, since in practice RNNs fail to remember longer dependencies, as well as are difficult to train due to the vanishing gradient problem [28].

In our work, we use Long Short-Term Memory Networks (LSTMs) [29] that are a variant of RNNs. LSTMs are capable of remembering long-term information by differently computing the hidden state of the network. The hidden state of LSTMs contains the chain of memory blocks which have special gates to control the information maintained in each cell of the memory block, effectively allowing LSTMs to selectively decide what to keep or erase from the memory. The outputs of LSTM are calculated by combining the memory together with the previous state of the network as well as the current input.

Complexity: The basic LSTM architecture (Figure 4(a)) consists of three layers: input, LSTM, and output layers. Data from the input layer is fed to the LSTM layer, where it recurrently flows and the memory cells are updated with values based on the input, output and forget gates. Next, data from the output unit is sent to the output layer.

The computational complexity of an LSTM network per time step and weight of LSTM is $O(1)$ [29]. Therefore, the learning complexity of it is $O(W)$, where W is the number of weights in the network that can be computed by the equation [30]:

$$W = n_c^2 \times 4 + n_i \times n_c \times 4 + n_c \times n_o + n_c \times 3 \tag{2}$$

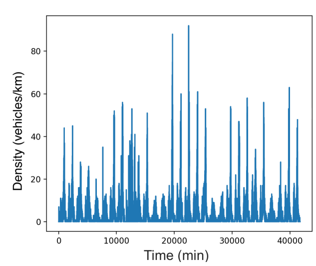
Here, n_c is the number of memory cells, n_i is the number of inputs fed into the LSTM layer and n_o is the number of outputs from the LSTM layer.

III. Traffic Prediction

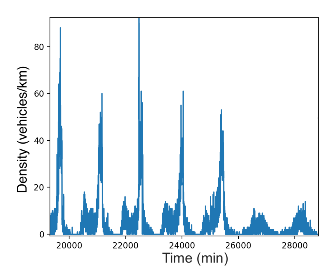
In this section, we talk about the input data and the prediction models used in our work.

A. Time Series Data

Our input data consist of different traffic parameters measured by the sensors placed on the lanes of highways. These sensors record the flow q and speed v of vehicles per minute passing the sensors. The density k is computed from these parameters using equation1. The density can be presented in the form of time-series as shown in Figure 3 (a) and (b).



(a) One month data.



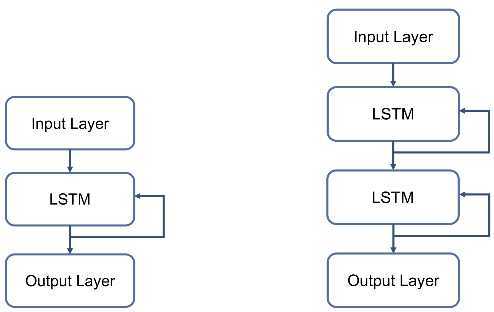
(b) One week data.

Figure 3: Density values for a sensor per minute.

We can see there is a pattern in weekly data shown in Figure 3 (b) from a single sensor, where the high peaks are weekdays and low ones are weekends. Similarly, there is a pattern with respect to the time of the day. We want our neural network to learn this density pattern in previous time stamps and make predictions for future timestamps. To achieve this, we use LSTMs to remember the pattern in data.

Since the traffic contains spatial dependency, we take neighbouring sensors into account for learning the traffic behaviour. In order to do this, we partition the highway network into areas containing long road stretches and junctions. Next, we deploy our models over these partitions. Details about choosing the neighbouring sensors and highway partitioning are given in Section IV.

B. Model Design



(a) Normal LSTM architecture. (b) Stacked LSTM architecture.

Figure 4: LSTM architectures.

We propose three prediction models: 1) The (1-1) single-sensor model that takes into account only one sensor and predicts traffic for the location of that sensor; 2) the (n-n) multi-sensor model that considers n sensors on a given area of road and gives predictions for the locations of all n sensors; and 3) the (m-n) multi-sensor model that uses only m significant sensors from an area to make predictions for all n sensors. The detailed working of these models is explained in Section V.

Our models use deep RNNs to capture the complex non-linear relation in the data more efficiently by making use of the hierarchical layers compared to simple RNN [31]. Stacked LSTMs refers to the architecture where multiple layers of LSTMs are placed over each other, as shown in Figure 4 (b), to give a more powerful and deep network compared to the conventional architecture in Figure 4 (a). In order to estimate the traffic density, we empirically found that the stacked architecture improves our results compared to the normal architecture. We used two layers in our model, more than two layers did not improve the accuracy due to over-fitting.

The input density data that we fed into the network is represented in the form of a space-time window. Consider an area over the highway containing n sensors, labelled as $S1, S2, S3, ..., Sn$. We take a look-back of L time stamps. If t is the current time stamp, then the look-back of L time-stamps means $t, t-1, t-2, ..., t-L$ previous density values. Figure 5 shows the input data representation, where each entry $k_{t,s}$ denotes the density value of a sensor s, i.e, $S1, S2, S3, ..., SN$, at time t, i.e,$t, t-1, t-2, ..., t-L$.

$$\begin{array}{c c}
 & \begin{array}{ccccc} S1 & S2 & S3 & \dots & SN \end{array} \\
\begin{array}{c} t \\ t\text{-}1 \\ t\text{-}2 \\ \vdots \\ t\text{-}L \end{array} &
\begin{bmatrix}
k_{11,} & k_{12,} & k_{13} & \dots & k_{1N} \\
k_{21,} & k_{22,} & k_{23} & \dots & k_{2N} \\
k_{31,} & k_{32,} & k_{33} & \dots & k_{3N} \\
\vdots & \vdots & & \dots & \vdots \\
k_{L1,} & k_{L2,} & k_{L3} & \dots & k_{LN}
\end{bmatrix}
\end{array}$$

Figure 5: Input data representation.

The neural network is trained to predict the density of the respective sensors corresponding to time stamps $t+1, t+2, t+3, ..., t+P$, where P controls the prediction interval. After experimenting with different values of L, we chose the value of 10 min. Less than this provided too little information and resulted in less accuracy. Beyond this made the input size large and the model did not give any improved results.

IV. Experimental Methodology

This experimental work is focused on evaluating different prediction models that we have proposed. Our experiments are based on answering the following general questions:

- **Accuracy**: How accurately the road traffic can be estimated using neural networks?
- **Accuracy Refinement**: Can the accuracy be improved by considering the neighbouring sensors?
- **Execution Time**: Can the execution time (the training time and prediction time) be improved by reducing the complexity of a neural network?
- **Scalability**: How the prediction models can be deployed over the highway network?

We later explain the dataset we use, followed by the implementation and metrics that we measure.

A. DataSet

We use real-world traffic data set from the Swedish Transport Administration [32] that consists of readings from sensors placed on Stockholm highways. Each lane of the highway contains sensors that are separated by few hundred meters. We have used one month data, which consists of sensor readings per minute during that month, i.e, total 44640 minutes readings as shown in Figure 3 (a). The data is further split into 70% training, 15% validation and 15% test data. The entire highway network of Stockholm, for which we have the sensor data, is shown in Figure 6 (a). This highway network consists of long road stretches connected together by different junction points. We took one of the long stretch and one complicated junction for our experiments. Figure 6 (b) contains the area with long stretch and Figure 6 (c) contains the area with the junction.

(a) Stockholm highway. (b) *Area 1*: Long road stretch. (c) *Area 2*: Triangular junction.

Figure 6: Sensors placed on Stockholm highways.

B. Implementation

In our experiments we used a system with Intel(R) Core(TM) i7-4980HQ CPU @ 2.80GHz processor, 16 GB RAM and macOS 10.13 High Sierra. We built our machine learning model using Python version 3.6.1. The libraries used are Keras 2.0.9 and Tensorflow 1.3.0.

C. Metrics

We evaluate the following metrics for our models:

- **Accuracy:** We evaluate the accuracy of prediction models by computing the **Root Mean Square Error (RMSE)** and **Mean Absolute Error (MAE)** between the predicted and actual traffic density time series.
- **Execution Time:** We evaluate the execution time of models by measuring their **training time** and **prediction time**.
- **Estimation Interval:** We evaluate the change in accuracy of prediction models for different time intervals.

V. Prediction Models

In this work, we propose different models for short-term traffic prediction. For every model we discuss the parameters that include: 1) The number of sensors a model covers for prediction, i.e, the output units of the model, 2) the computational complexity of the model, 3) the input units used by the model and, 4) the number of memory blocks for the LSTM network required by the model. Table I contains values of these parameters for *Area 1* (long road stretch) and *Area 2* (triangular junction) shown in Figure 6 (b) and (c). The number of memory blocks mentioned in the Table I are for a single LSTM layer, and our models have two stacked LSTM layers. These memory blocks are empirically chosen. We pick the number of memory blocks after which the accuracy stops increasing.

We categorize our models into three types based on the categorization criterion aforementioned.

		Area 1		Area 2	
Model	**Memory blocks**	**Inputs units**	**Output units**	**Input units**	**Output units**
Single Sensor (1-1)	50	1	1	1	1
Multi-Sensor (n-n)	200	33	33	20	20
Multi-Sensor (m-n)	150	10	33	8	20

Table I: Parameters for different prediction models.

A. Single Sensor (1-1) Model

This model works for predicting the traffic density for a single sensor. The input and output for this model being traffic density time series from a single sensor make it less complicated because the LSTM network has to deal with one time series. Figure 7 shows the single sensor model, where the input density is taken from one sensor $S1$ to estimate the future density. In this simple model, the prediction only depends upon the readings of the single sensor, without considering any neighbouring sensors information.

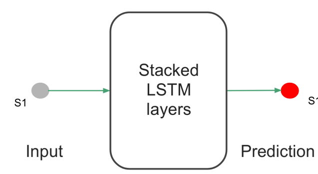


Figure 7: Single Sensor (1-1) Model.

Experimental Setup: We consider a random sample of sensors from *Area 1* and *Area 2* shown in Figure 6 (b) and (c) for the single sensor model. The model is used for each sensor and the execution time in terms of its training and prediction time is measured. Next, the accuracy for different estimation intervals, i.e, 10 min, 20 min and 30 min is computed. We measure the accuracy as the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). We compare our model (LSTM-2 with two stacked layers) with other classical baseline statistical models that include Auto Regression (AR), Autoregressive Integrated Moving Average (ARIMA) [3], Support Vector Regression (SVR) [33], and neural network based models that include, Recurrent Neural Network (RNN) with two layers, Feed Forward Neural Network (FFN) with two layers and LSTM-1 with a single LSTM layer.

Experimental Results: Table II shows the RMSE and

MAE values for different time intervals. As the results indicate the stacked LSTM neural network (named LSTM-2 in the table) performs better than other prediction models. The error eventually increases with the increase in estimation interval. Next, we want to evaluate if the accuracy improves by taking multiple sensors into account during prediction.

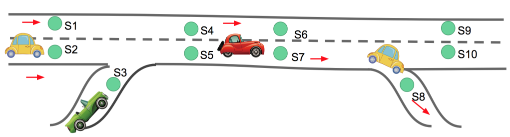
	10 min		20 min		30 min	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
AR	6.87	5.9	7.46	6.31	8.09	6.85
SVR	8.30	7.68	9.19	8.71	10.61	10.19
ARIMA	7.67	6.74	9.40	8.34	10.86	9.81
RNN	5.60	2.63	6.65	3.32	7.75	3.39
FFN	5.62	2.46	6.87	3.46	7.86	**3.34**
LSTM-1	5.63	**2.41**	7.13	3.65	7.71	3.73
LSTM-2	**5.49**	**2.41**	**6.62**	**3.07**	**7.60**	3.45

Table II: Accuracy of different models for a single sensor.

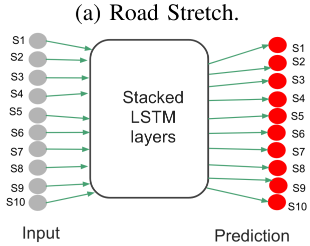
B. Multi-Sensor (n-n) Model

In the multi-sensor, model we consider an area over the highway and predict the density values for the sensors that fall in that area. In this case, the prediction is done by taking the neighbouring sensors into account. The neighbouring sensors provide more data for prediction. Figure 8 (a) and (b) show a road stretch with 10 sensors on it, all these sensors are taken as input for this model.

This model is complex because the number of inputs and the number of outputs is equal to the total number of all the sensors that fall in that area. The more the number of sensors, the more memory blocks are required and the greater is the complexity of the model according to Eq. 2.



(a) Road Stretch.



(b) Neural Network.

Figure 8: Multi-sensor (n-n) model.

Experimental Setup: We take sensors that fall in *Area 1* (long road stretch) and *Area 2* (triangular junction) shown in Figure 6 (b) and (c). For *Area 1* we consider the highway path going towards North. *Area 2* is complicated because it consists of vehicles going in different directions, making it hard for the neural network to learn the relation between sensors. Our experiments show RMSE up to 10 without partitioning *Area 2*, which is reduced by a factor of 5 after partitioning, i.e, RMSE $\approx$ 2. Therefore, we partition this area into paths consisting of cars going towards the same direction. For example one of such paths is shown in Figure 9. The red path is for cars going towards North from West and East.

We compare our model (LSTM-2 with two stacked layers) with neural network based models that include: Recurrent Neural Network (RNN) with two layers, Feed Forward Neural Network (FFN) with two layers and LSTM-1 with single LSTM layer. We did not use statistical models because of their poor accuracy results in the previous experiment (Section V-A) and their complexity to implement a multi-variate model.

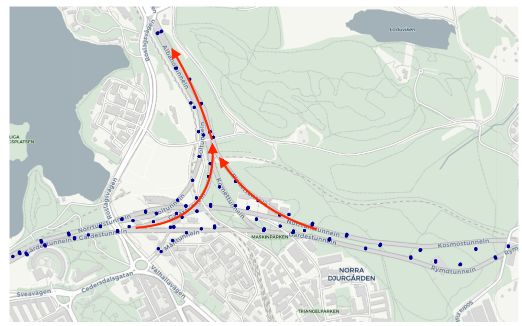


Figure 9: *Area 2*: Path in the triangular junction.

The road section of the considered paths can be divided further into three sections: 1) the *entrance*: it consists of beginning two groups of sensors (a group contains sensors from all the lanes placed at the same distance reference), 2) the *exit*: it consists of last two groups and 3) the *middle*: it contains all the remaining sensors. We evaluate our model over these section of roads.

Experimental Results: Tables III and IV show the average RMSE and MSE values for *Area 1* (long road stretch) and *Area 2* (triangular junction) over different estimation intervals. According to our results, stacked LSTM with two layers (LSTM-2) has better accuracy in most of the cases compared to other models.

In order to check the accuracy distribution along areas, we measure the prediction accuracy at the entrance, middle and exit sections of areas. Figure 10 and 11 contain RMSE for *Area 1* (long road stretch) and *Area 2* (triangular junction) over 10 min, 20 min and 30 min estimation intervals. For both areas, the error is higher at the *entrance* of an area, followed by the *middle* section of the highway area; whereas, the *exit* section has the lowest error. Furthermore, the error is increasing with the increase in estimation interval. This increase is more for the *entrance* section compared to other sections. The reason for the least prediction error at the *exit* section because the model has more information for prediction towards the end of the area. Stacked LSTM model (LSTM-2) has less error compared to others.

	10 min		20 min		30 min	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
RNN	3.33	2.35	4.6	3.41	5.0	3.49
FFN	3.36	2.36	4.4	3.39	5.2	3.44
LSTM-1	3.14	2.22	3.8	2.67	4.1	3.29
LSTM-2	**2.94**	**2.06**	**2.94**	**2.06**	**3.22**	**2.24**

Table III: Prediction accuracy for multiple sensor using different models in *Area 1, (long road stretch).*

	10 min		20 min		30 min	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
RNN	2.48	1.49	2.52	1.51	2.67	**1.07**
FFN	2.48	1.45	2.56	1.52	3.04	1.53
LSTM-1	2.38	**1.40**	2.46	**1.45**	2.57	1.55
LSTM-2	**2.35**	1.43	**2.45**	1.49	**2.51**	1.52

Table IV: Prediction accuracy for different models for multiple sensor in *Area 2, (triangular junction).*

C. Multi-Sensor (m-n) Model

The multi-sensor (m-n) model is a variant of the multi-sensor (n-n) model introduced in V-B using the stacked LSTM (LSTM-2) model. Instead of n sensors that fall in the area under consideration, we take only m sensors from those n sensors and predict the output for all n sensors. The sensors in the m set include boundary sensors, and sensors located at exits and entry points to the highway. The reason to include those sensors is that they are more important in terms of affecting the traffic flow. Intuitively, if we know the behaviour of cars entering and exiting the highway, we have to guess what happens inside the highway. Therefore, we consider the entry and exit sensors as inputs for our neural networks to predict density for all the sensors. Figure 12 (a) and (b) show a road stretch with 10 sensors on it, only the boundary sensors $S1, S2, S9, S10$, and the sensors located at entry and exit points $S3$ and $S8$, are taken as input for this model. In this way, we reduce the complexity of the neural network by reducing the number of inputs units and the memory blocks based on Eq. 2.

Experimental Setup: The experimental setup for the Multi-sensor (m-n) model is similar to one used for the Multi-sensor (n-n) in Section V-B. The purpose of this experiment is to evaluate if we can reduce the complexity of the LSTM network by limiting the number of input sensors without compromising the prediction accuracy.

Experimental Results: Figure 13 (a) and (b) show RMSE of the (m-n) model for *Area 1* and *Area 2* over 10 min, 20 min and 30 min estimation intervals. The error is increasing with the increase in estimation interval. The *entrance* has highest error compared to other sections.

Congestion Detection: Our density predictions are useful for detecting congestion in the road traffic. From our experiments we find the critical density, $k_{critical}$ (see Figure 2), to be between 35 and 40 vehicles per km. Using our model, we were able to correctly predict congestion, i.e, density values near to k_{jam} (see Figure 2), 94% of the time.

D. Comparison

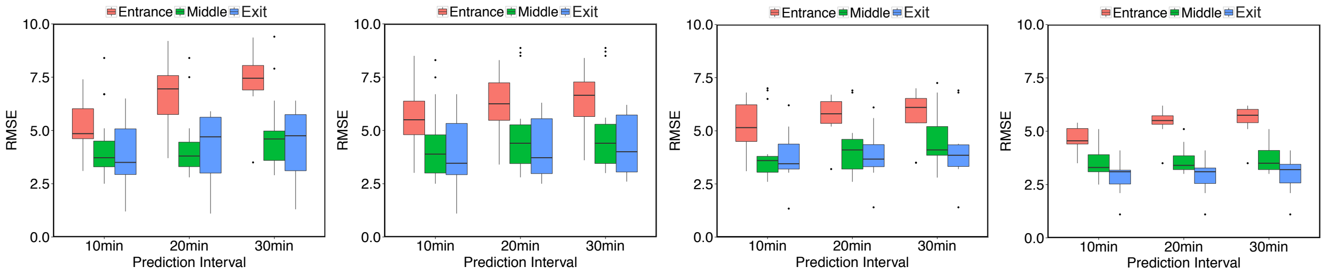
Now that we know the accuracy of our models, we want to know how fast do they perform. For this reason, we compared the execution time of our models by measuring the training time and prediction time, shown in Figure 14. The single sensor (1-1) model is fast because it is considering one sensor at a time. It might take longer execution time if we run several such models together for multiple predictions over limited resources of a system. The (m-n) multi-sensor model takes less training and prediction time compared to the (n-n) multi-sensor model. This is because the (m-n) model has less input and memory units which reduce its complexity and improve its execution time.

E. Discussion

Our experimental results show that using neighbourhood sensor information gives higher prediction accuracy than using a single sensor data. This is because the neural network is fed with more information. It learns the behaviour of traffic better by using sensors placed together over a path of the highway. The reading of sensors placed at the entrance of highway indicates the traffic conditions that will propagate towards the middle and exit sensors. In other words, model learns more information for the middle and the exit section. Therefore, the prediction for these sections is better than the entry section. Additionally, we observed that improving the complexity of a model by reducing the input units and memory units improves its execution time. Such lower complexity model has a strong potential to be applied within edge computing domain in the future.

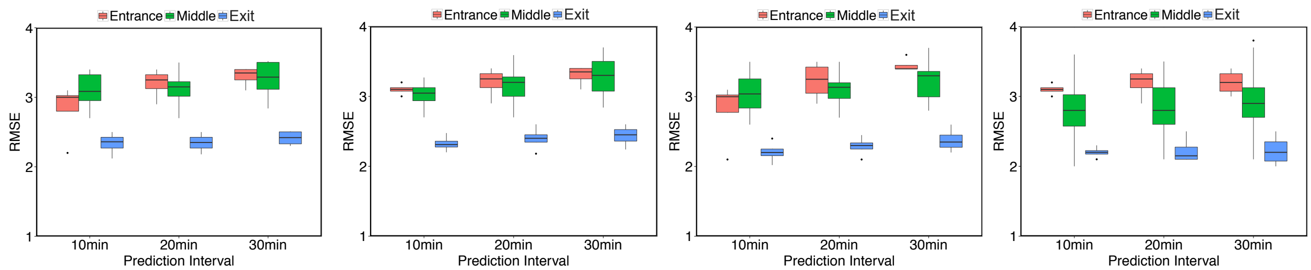
VI. Conclusion and Future Work

Our work comprises of three prediction models for estimating traffic density using stacked LSTM neural networks. We have implemented and compared these models over different sections of Stockholm highways using real datasets. Our multi-sensor (m-n) model that uses input readings from only m significant sensors rather than all n sensors, predicts density for all n sensors with acceptable accuracy comparable to the multi-sensor (n-n) model, which takes input from all n sensors. Initially, all sensors are required to train the model, and after training only significant sensors can be kept for prediction over all sensors with acceptable accuracy. To train the model, temporary sensors can be deployed together with significant sensors and then the former can be removed or shut down. This allows reducing the number of sensors and saving the infrastructure cost.



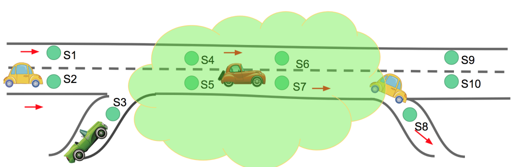
(a) RNN (b) FNN (c) LSTM-1 (d) LSTM-2

Figure 10: RMSE of (n-n) models for the *Area 1*: long road stretch.

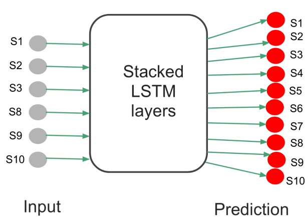


(a) RNN (b) FNN (c) LSTM-1 (d) LSTM-2

Figure 11: RMSE of (n-n) models for *Area 2*: triangular junction.

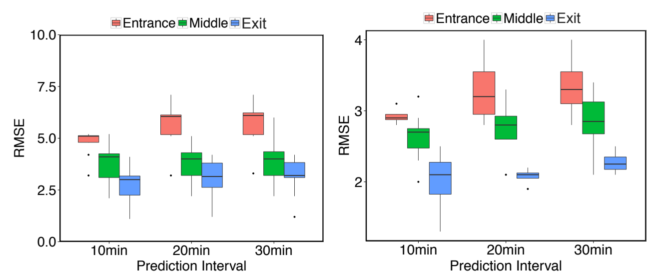


(a) Road Stretch.



(b) Neural Network.

Figure 12: Multi-sensor (m-n) model.



(a) RMSE (*Area 1*) (b) RMSE (*Area 2*)

Figure 13: RMSE of (m-n) model for different road sections of (*Area 1*: long road stretch and *Area 2*: triangular junction).

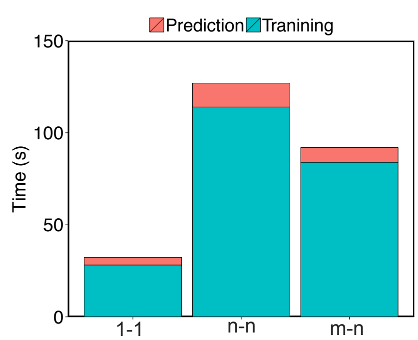


Figure 14: Execution Time Comparison.

Our future work includes investigating on how accuracy depends on the size of road segments and the number of sensors. We will also research on how aggregation levels impact accuracy. We expect that fine-grained aggregation used in this paper, captures more details but is more challenging to predict due to high noise levels compared to a smoother coarse-grain aggregation that captures only general trends. We intend to develop a method to optimally partition the road network and to place sensors in order to achieve high prediction accuracy while lowering the infrastructure cost.

ACKNOWLEDGMENT

This work was supported by the project BADA: Big Automotive Data Analytics in the funding program FFI: Strategic

Vehicle Research and Innovation (grant 2015-00677) administrated by VINNOVA the Swedish government agency for innovation systems, by the project BIDAF: Big Data Analytics Framework for a Smart Society (grant 20140221) funded by KKS the Swedish Knowledge Foundation, and by the Erasmus Mundus Joint Doctorate in Distributed Computing (EMJD-DC) programme funded by the Education, Audiovisual and Culture Executive Agency (EACEA) of the European Commission under FPA 2012-0030.

References

[1] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.

[2] A. Skabardonis and N. Geroliminis, "Real-time estimation of travel times on signalized arterials," Tech. Rep., 2005.

[3] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transportation Research Record Journal of the Transportation Research Board*, no. 722, 1979.

[4] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[5] N. Juri, A. Unnikrishnan, and S. Waller, "Integrated traffic simulation-statistical analysis framework for online prediction of freeway travel time," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2039, pp. 24–31, 2007.

[6] P. E. Pfeifer and S. J. Deutrch, "A three-stage iterative procedure for space-time modeling phillip," *Technometrics*, vol. 22, no. 1, pp. 35–47, 1980.

[7] S. Clark, "Traffic prediction using multivariate nonparametric regression," *Journal of transportation engineering*, vol. 129, no. 2, pp. 161–168, 2003.

[8] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, "Short term traffic forecasting using the local linear regression model," in *82nd Annual Meeting of the Transportation Research Board, Washington, DC*, 2003.

[9] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[10] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 2, pp. 121–135, 2003.

[11] B. Williams, "Multivariate vehicular traffic flow prediction: evaluation of arimax modeling," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1776, pp. 194–200, 2001.

[12] M. S. Dougherty, H. R. Kirby, and R. D. Boyle, "The use of neural networks to recognise and predict traffic congestion," *Traffic engineering & control*, vol. 34, no. 6, 1993.

[13] P. Vythoulkas, "Alternative approaches to short term traffic forecasting for use in driver information systems," *Transportation and traffic theory*, vol. 12, pp. 485–506, 1993.

[14] H. Zhang, "Recursive prediction of traffic conditions with neural network models," *Journal of Transportation Engineering*, vol. 126, no. 6, pp. 472–481, 2000.

[15] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[16] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[17] 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, p. e0119044, 2015.

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

[19] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[20] Y. Duan, Y. Lv, W. Kang, and Y. Zhao, "A deep learning based approach for traffic data imputation," in *Intelligent Transportation Systems (ITSC), IEEE 17th International Conference on*. IEEE, 2014, pp. 912–917.

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

[22] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[23] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[24] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," *arXiv preprint arXiv:1612.01022*, 2016.

[25] M. Fouladgar, M. Parchami, R. Elmasri, and A. Ghaderi, "Scalable deep traffic flow neural networks for urban traffic congestion prediction," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2251–2258.

[26] G. Whitham, "On kinematic waves ii. a theory of traffic flow on long crowded roads," in *Proc. R. Soc. Lond. A*, vol. 229, no. 1178. The Royal Society, 1955, pp. 317–345.

[27] P. I. Richards, "Shock waves on the highway," *Operations research*, vol. 4, no. 1, pp. 42–51, 1956.

[28] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE trans. on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[31] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Advances in neural information processing systems*, 2013, pp. 190–198.

[32] Trafikverket, https://www.trafikverket.se/, 2010.

[33] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.