Disaster Management in Smart Cities by Forecasting Traffic Plan Using Deep Learning and GPUs

Muhammad Aqib[1(✉)], Rashid Mehmood[2], Aiiad Albeshri[1], and Ahmed Alzahrani[1]

[1] Department of Computer Science, FCIT, King Abdulaziz University, Jeddah 21589, Saudi Arabia
aqib.qazi@gmail.com, {aaalbeshri,asalzahrani}@kau.edu.sa
[2] High Performance Computing Center, King Abdulaziz University, Jeddah 21589, Saudi Arabia
RMehmood@kau.edu.sa

Abstract. The importance of disaster management is evident by the increasing number of natural and manmade disasters such as Irma and Manchester attacks. The estimated cost of the recent Irma hurricane is believed to be more than 80 billion USD; more importantly, more than 40 lives have been lost and thousands were misplaced. Disaster management plays a key role in reducing the human and economic losses. In our earlier work, we have developed a disaster management system that uses VANET, cloud computing, and simulations to devise city evacuation strategies. In this paper, we extend our earlier work by using deep learning to predict urban traffic behavior. Moreover, we use GPUs to deal with compute intensive nature of deep learning algorithms. To the best of our knowledge, we are the first to apply deep learning approach in disaster management. We use real-world open road traffic within a city available through the UK Department for Transport. Our results demonstrate the effectiveness of deep learning approach in disaster management and correct prediction of traffic behavior in emergency situations.

Keywords: Smart cities · Disaster management · Deep learning GPUs · Convolution neural networks

1 Introduction

A large amount of world's population is currently living in cities because of increased trend of urbanization. The provision of educational, health, social, cultural and other facilities is a main reason behind this trend. To provide best facilities to the citizens, companies and government authorities relies on latest technologies and use devices that collects and generates a lot of data. These devices not only include personal devices like smart phones, GPS devices etc. but also include the devices that are used by government departments like sensors to switch on and off the lights on streets, sensors and cameras to control traffic, smart health care devices and systems and many more.

R. Mehmood et al. (Eds.): SCITA 2017, LNICST 224, pp. 139–154, 2018.
https://doi.org/10.1007/978-3-319-94180-6_15

All this together makes foundations of smart city where a tremendous amount of data is collected from devices and processed to improve the life style of its citizen and to improve their productivity by providing them a secure and peaceful environment [1, 2].

Disaster management systems or emergency response systems in a smart city are very important to efficiently handle disaster conditions in effective way, no matter the disaster was a manmade like 9/11 attacks, blasts etc. or it was natural like earthquake, tsunami etc. Traffic management in a disaster plays key role in evacuating the affected area and to monitor the traffic in other parts of the city to avoid congestion and road blockages. For efficient traffic management, traffic data could be collected from the sensors and cameras deployed on the road networks. In addition to this, data collected from GPS could also be used to monitor the traffic flow and to guide the people through different applications to take alternate routes to ensure their safety, avoid congestion and to provide the emergency services in affected area in efficient way.

In our earlier works, [3], by leveraging the advancements in the intelligent transportation systems, VANETs, and other technologies including mobile and cloud computing technologies, we proposed a disaster management systems for smart cities. This system was able to collect information from these sources and to propagate them to the vehicles, people and other components of the disaster management system in real time. In addition to this, it was able to ensure the security and safety of data and applications as well. In this work, a cloud based architecture was given and a microscopic traffic model Lighthill-Whitham-Richards (LWR) was used. The effectiveness of our model was tested by modeling the impact of a disaster on a real city and comparing it with a disaster management technique using traditional technologies. Later, this work was extended in [4] where we improved our model by introducing a message propagation through VANETs. Microscopic traffic models were used in this work as well in addition to our novel algorithm. Extended simulation results were used to demonstrate the effectiveness of newly proposed system. In continuation to these works [3, 4], we developed a model in [5] to evaluate the performance of disaster management systems on evacuation operations. In this work, microscopic models were used for the design and evaluation of our system. Two main evacuation strategies, demand strategy (DS) and speed strategy (SS) has been reported in this work.

Due to the availability of tremendous amount of data collected by devices in a smart city, many machine learning approaches could be applied on that data to get useful insights and based on the collected information from that data, the future steps taken by individuals or authorities in specific conditions could be predicted. The idea of machine learning or artificial intelligence techniques is not new but it was not applicable because it requires a huge amount of data for learning phase. Deep learning is also a machine learning approach that could be used to train the models using historic or real-time data and then those trained models could be used to predict the expected values.

In this work, we are using deep learning for traffic management in smart cities in disasters. Deep learning requires a large amount of data to train the model and therefore takes long in training process. It is more accurate but is intensive in computation, hence we need GPU [6]. Therefore, we are using GPUs to expedite the training process and to provide results in close real-time fashion. Traffic data for this purpose is collected from data.gov.uk that provides annual average vehicles flow values on roads in a city in UK.

The rest of the paper is organized as follows. Section 2 provides background material that defines the tools and technologies used in our work. Work done by others in this area is presented in Sect. 3. Our proposed framework is presented in Sect. 4. In order to find the suitable city data, we have examined a number of datasets and their details are given in Sect. 5. Performance evaluation and analysis of the proposed system is given in Sect. 6 and, finally, in Sect. 7 we have concluded the discussion with directions for future work.

2 Background Material

In this section, we will give a brief introduction to the tools and technologies used in our model in specific and some tools and simulators that are used for traffic modeling in general.

2.1 Graphical Processing Units

In this section, we will give an overview of the GPU architecture. A GPU chip contains multiple multi-processors (MP) and each MP contains many stream-processors (SP). Instructions are executed in SP like ALU in CPU. Different tasks are performed on MPs and they are mutually independent to each other whereas the SPs in an MP executes the same operations on different data items. To store data, each SP has its own register to store variables and temporal data. An SP cannot access the registers of other SPs in an MP. For this purpose, there is a shared on-chip memory that is accessible to each SP in that MP. In addition to this, an off-chip shared memory, called global memory is also available and it can be accessed by all the SPs in all the MPs. This global memory is connected externally to the GPU chip and it is much larger in size but the access to this memory is much more expensive than that of the on-chip shared memory inside the MPs.

Programs in GPU are executed with the help of Compute Unified Device Architecture (CUDA) toolkit offered by Nvidia and detailed execution flow of a CUDA, the logical structure of kernel threads and logical to physical mapping in GPU is also part of the discussion.

2.2 Deep Learning

A branch of computer science that gives the computers, the ability to learn themselves like human beings is known as machine learning. Machine learning does not require programmers to program something explicitly to tell computers to perform a specific task. Instead, machine learning algorithms train computers using different algorithms to predict the output when a specific input is given. Techniques that enable computers to learn something without explicit programming are divided into two main categories in machine learning. These are known as supervise learning and unsupervised learning techniques. Artificial neural network, clustering, genetic algorithms and deep learning etc. are some examples of machine learning techniques. In this section, we will focus on the deep learning techniques and work done in this domain.

Deep learning approaches have been classified into different categories based upon the nature and training and testing strategies. These include Convolutional Neural Networks (CNNs), Restricted Boltzmann Machines (RBMs), Autoencoders and Sparse Coding techniques [7]. In this work, we are using CNNs for training and testing purposes. So, we will discuss them in detail in the following paragraph.

2.2.1 Convolutional Neural Networks (CNNs)

In the CNNs, multiple layers including convolutional, pooling, and connected layers are used for training purpose in a robust manner. Authors in [7] have defined a general architecture of CNN for image classifications. This architecture is shown in Fig. 1. In this figure, the whole process is divided into two main phases, forward phase that include convolutional and pooling layers and backward phase where fully connected layers are used to produce the output.

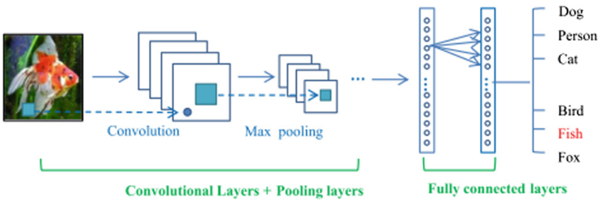

Fig. 1. General CNN architecture [7]

Convolutional neural networks are the hierarchical neural networks and their convolutional layers alternate with subsampling layers like simple and complex cells in the primary visual cortex. CNNs vary in how convolutional and sub-sampling layers are realized and how the nets are trained [8].

3 Related Work

In this section, we are presenting the work that deals with the traffic management plans during emergency conditions in smart cities. Some people focus mainly on traffic management in smart cities using any approach and some have focused on the approach i.e. deep learning with smart city scenario on low priority. As we are combining traffic management in smart cities with the deep learning approach, so both are useful for us and therefore we are presenting some approaches for better understanding of the work done in this area.

Authors in [9] have proposed an adaptive traffic management plan to ensure the provision of secure and efficient emergency services in case of disaster in a smart cities. In this work, a framework has been proposed introduces some components of traffic management system like traffic management controllers (TMC), local traffic controllers (LTC), adaptive traffic light controllers, environmental sensor controllers etc. The goal of this framework is to collect information from communication and other devices about the severity of the disaster that has been divided into three categories in this work; low, medium, and high, and then act accordingly by using these controllers. For example, in case of high emergency condition, traffic signals could be controlled to ensure the timely arrival of emergency vehicles e.g. ambulance and fire brigade and to reroute the non-emergency traffic. SUMO [10] has been used to simulate this process. In this work, focus is mainly on the provision of emergency services and their security and the plan has been simulated but no practical scenario or data has been used to handle the traffic and it also lacks the plan to manage the general traffic in case of disaster.

Smart cities are characterized by advanced and integrated ICT systems, such as smart logistics solutions [11], autonomic transportation [12]. Internet of things (IoT) could be considered as the back bone of future smart cities [13]. [14] proposes a ubiquitous learning system for smart societies. This approach can be used to educate and prepare citizens for disasters. In particular to vehicles, internet of vehicles (IoV) includes all the devices that could be used to monitor the vehicles and for inter vehicle communication as well. Data from different types of sensors placed on road networks, vehicles, and other smart devices [15] is collected to traffic management. There are many studies that use IoT and IoV to propose a traffic management plan as in [16, 17]. In addition to this, a lot work has done in the area of autonomic transport management in smart cities [18]. Work in [19] also shows the importance of Fog and other cloud technologies in dealing with emergency situations in smart cities. In [20] a parallel transportation management and control system for smart cities has been presented that not only use the artificial intelligence technologies but also uses massive traffic data and uses big data technologies or frameworks like MapReduce. Thus, shows the importance of these technologies in traffic management in smart cities.

A traffic flow prediction approach has been proposed in [21]. Authors have used the deep learning approaches for prediction purpose using a large amount of data. They have proposed a model that uses autoencoders for training and testing purpose to make predictions. The model is named as stacked autoencoder (SAE) model. To predict traffic flow at time t, traffic flow data at previous time intervals has been used. The proposed model has been used to predict 15, 30, 45 and 60 min traffic flow. Data for this purpose was collected from Caltrans Performance Measurement System (PeMS) [22]. Three months' data, collected every 30 s was used for training and testing purposes. In this data, vehicle flow was collected where two directions of same freeway were treated as different freeway. Support vector machines (SVM) have been used for comparison purpose.

Authors in [23] have proposed a deep learning based approach for traffic flow prediction and they have used unsupervised learning approach using deep belief networks. They have categorized the traffic prediction approaches into three main categories that include time-series approaches, probabilistic approaches, and non-parametric approaches such as neural network based approaches etc. Authors in this work have used Restricted Boltzmann Machines (RBMs) for training purpose which are stacked one on other. For training and testing purposes, inductive loop dataset is obtained from the PeMS [22]. In addition to this, authors have used data from highway system of China (EESH) as well. A data of 12 months has been collected and the first 10 months' data is used for training whereas the data of remaining two months has been used for validation purpose. Prediction results have been compared with other four methods for top 50 roads having high flow rates. The results shows that deep learning based architecture is more appropriate and robust in prediction and could be used for practical prediction system.

A deep learning based approach has been used in [24] to model the traffic flow. In this work, authors have developed deep learning predictors to predict the traffic flow data from the road sensors. Real-time traffic data has been used and by using the proposed model, they have predicted the traffic flow during a Chicago Bears football game and a snowstorm. They have used the number of locations on the loop detectors and traffic flow at a time (say t). They first have developed a linear vector auto regressive model for predictors selection. These predictors are later used to build a deep learning model. Stochastic gradient descent (SGC) method is used to know to structure and weights of parameters. They also have applied three filtering techniques (exponential smoothing, median filter, loess filter) on traffic data to filter noisy data from the sensors. Data for this purpose is collected from 21 loop detectors on five minutes' interval basis. This data includes speed, flow and occupancy. They have built a statistical model to capture the sudden changes from free flow (70 mph) to congestion (20 mph). In case of bottlenecks, they predict that how fast it will propagate on the network i.e. loop detectors. For predictor selection, deep learning model estimates an input-output map with the assumption that they need the recent. So, they collect last 12 readings from each sensor. The performance of DL model has been compared with sparse linear vector autoregressive (VAR). Both accurately predict morning rush hours on normal day but VAR miss-predicts congestion during evening rush hour. On the other hand, DL predicts breakdown accurately but miss-estimates the recovery time.

Authors in [25] also have used deep learning approach to predict the traffic congestion. They have used recurrent neural networks by using Restricted Boltzmann Machine (RNN-RBM). For comparison purposes, authors have used Support Vector Machines (SVMs) and found that prediction accuracy was increased by at least 17%.

4 Disaster Management System

In this section, we will discuss the proposed deep learning based disaster management system in detail. Figure 2 depicts the architecture of our proposed system. Proposed framework consists of three main layers, input layer, data processing and prediction layer and output layer.

4.1 Input Layer

Input layer manages the traffic data that is used to training and testing of deep learning model in the data processing layer. The input data could be either offline i.e. historical data or it could be real-time or streaming data. Input layer gets the traffic data and extracts the key features from it like flow, speed, occupancy, density etc. These features play a key role in prediction making using the proposed model. The role of input layer becomes more important especially when we are dealing with the real-time data. In this case, it takes the data from the source, and provides it to the processing layer for further data formatting.

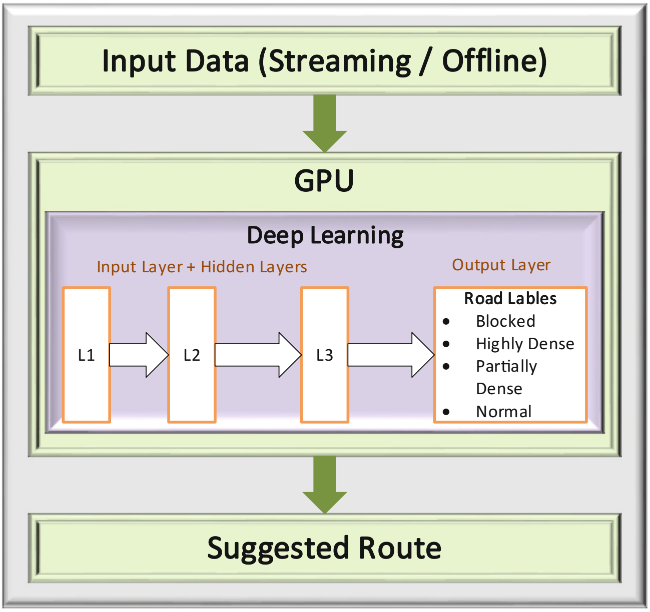

Fig. 2. Proposed disaster management framework.

4.2 Data Processing Layer

This layer is responsible to process the input data for making predictions in case of disaster. Our prediction model uses deep learning approach for this purpose. By using a deep regression model, we train a dataset which is further tested using another input dataset or a subset of the same dataset. Data processing layer, takes the data from the input layer and then process it to convert the input data into the format required by the deep learning algorithm. For example, if date attribute is included in the input dataset, it could be processed in this layer to get day, month, year, hour etc. The division of one attribute into multiple attribute could be useful in training process e.g. we can get peak hours, and can separate the data based on weekends etc. Furthermore, we may need to normalize the input data for our regression model. So, data normalization is also performed in this layer.

4.3 Deep Learning Layer

We have used deep regression model to estimate the vehicle flow value by using multiple input features. Initially we have trained our neural network by adding two hidden layers to the network. First layer is our input layer and the final one is the output layer and the two hidden layers are in between the input and output layers. Forward propagation scheme has been used for computation of weights and finally loss is calculated on the overall output.

Figure 3 shows a neural network including one input, two hidden and one output layer. In our case, we are using 9 input parameters, and output layer gives one output value because we are applying regression to get one vehicle flow value. We have used *relu* activation functions and *AdamOptimizer* has been used to optimize the generated results. We ran the training process for 1000 times by selecting a data size of 500 features at one time.

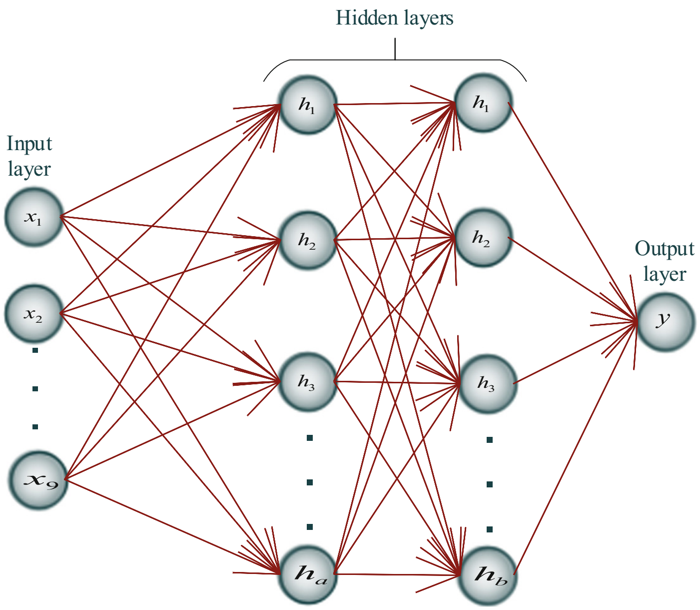

Fig. 3. Our deep neural network with two hidden layers.

5 Datasets

In this work, we are mainly working on UK traffic data. So, we have explored a variety of traffic data available through multiple sources in UK that could be used for different purposes to work on traffic management plans. Some data sources of same kind outside UK are also included in the list. In our deep learning model, we have used the data from data.gov.uk that provides the vehicles flow data for minor cities. This includes the average vehicle count or roads for different vehicle types. In Table 1, we have given some data sources that provide traffic data. Short data description and URLs to access the data are also given.

6 Analysis and Comparison

This section defines our deep model configurations and the performance metrics used for analysis purpose which is used for performance analysis of our model.

Table 1. UK traffic data sources.

	Data source	Description	URL
1	Transport for London (TFL)	Data could be accessed by using the provided API. Real-time data and status information of different sources of transportation could be accessed by using API	https://tfl.gov.uk/info-for/open-data-users/
2	London Datastore	Public data sharing portal that provides data related to different department of London government. Data from 1997 to 2015 is also available that provides number of vehicles on different roads in London	https://data.london.gov.uk/
3	Data.gov.uk	Data provided by different United Kingdom's government agencies could be accessed from this portal. Its transport data section provides many options to explore traffic data	http://data.dft.gov.uk/ https://data.gov.uk/dataset/gb-road-traffic-counts
4	Data from local government association UK	This is a research project and its purpose is to make data useful for LGA	http://www.local.gov.uk/web/guest/research/-/journal_content/56/10180/7783953/ARTICLE
5	Transit Feeds	It provides web feeds for transport data and provides updated information related to transport department of a city or state etc.	http://transitfeeds.com/
6	Department for Transport UK	It provides data for all the A class roads at city level. Data collected from data collection points on roads that fall in the selected city could be accessed from this source	
7	Transport Infrastructure Ireland (TII)	This site also provides traffic data for main roads (highways). It could be useful while dealing with the intercity traffic data. Do not provide enough data to deal with the traffic on minor roads in a city	https://www.nratrafficdata.ie/c2/gmapbasic.asp?sgid=ZvyVmXU8jBt9PJE$c7UXt6
8	Tyne and Wear Region Data	We can access the live traffic data by using the API provided by the "Open Data Service" authority	http://www.gateshead.gov.uk/Parking-roads-and-travel/planning/TADU.aspx

(*continued*)

Table 1. (*continued*)

	Data source	Description	URL
9	The WisTransPortal System	Hourly traffic data index page could be accessed to get a list of counties in the Wisconsin State, USA or county could be selected from the map as well. By selecting the county, it displays all the data available for different roads in that county by their names	https://transportal.cee.wisc.edu/products/hourly-traffic-data/
10	Wisconsin Department of Transport	Provides traffic flow data on weekly and/or annual basis on selected roads (say highways)	http://wisconsindot.gov/Pages/projects/data-plan/traf-counts/default.aspx
11	North East Combined Authority	Provides data for selected areas. It provides data related to special events, roadworks, incidents, journey times for key roads, car parks and CCTV images	https://www.netraveldata.co.uk/
12	Highways England	Provides three types of data: Monthly Summary Data, Journey Time Data, and Traffic Flow Data. HE also provides a conversion table that gives description of traffic data measurement sites	http://tris.highwaysengland.co.uk/
13	Developer.here.com	Provides API to get traffic flow and incidents data	https://developer.here.com/

6.1 Deep Model Setup

In this work, we have used vehicles flow data on minor roads in a city in UK. It includes six different vehicle categories ranging from cars or small personal vehicles to big trucks used for transportation of goods. Data used as input contains 70470 data flow values for all six vehicle categories for the years from 2000 to 2015 and the road names along with the road categories are also given.

We are using a deep regression model to predict the vehicle flow values. We have implemented this model using Keras deep learning library [26]. Our regressing model has four layers including one input, two hidden and one output layer. We have used the annual average flow data to predict the traffic flow in a city. Input dataset is divided in the ratio of 7, 2, 1 for training, testing and prediction purposes respectively. Batch size was set to 10 and number of epoch was set to 1000.

6.2 Dataset Schema

Dataset we have used in this work contains annual average flow data for different types of vehicles. It also provides road names, road category and other information. In Table 2, we have given the schema of input dataset that provides brief description of some important input attributes in this dataset.

Table 2. Schema of dataset used as input in our deep learning model

S.No	Attribute name	Description
1	Road	Gives character code names assigned to a road in the city
2	Road name	Name of the road
3	RCat	Roads have been divided into different categories. RCat gives character codes to define its category in city road network
4	iDir	Traffic direction on a road e.g. heading east or west
5	Year	Year for which AAFD was collected
6	dCount	Day of the year when data was collected. It is in the format dd-mm-yy h:mm
7	Hour	Hour of the day
8	CAR, BUS, LGV, HGVR2, …	A set of different types of vehicles to provide their flow values. For example, car gives the annual average flow value for cars. Similarly, Bus, provides the annual average flow value for buses and so on

6.3 Performance Analysis

In this paper, our focus is mainly on providing details of the deep learning based traffic prediction approach. Details of the overall evacuation method can be found in our earlier work [3–5]. For our deep learning model, we divided the dataset into three parts where 70% data was used for training, 20% data for testing purpose and the rest 10% data is used for prediction purposes. Our deep learning model was executed for 20 times to get results for analysis purpose. Furthermore, for all the 20 models, the batch size for training purpose was 10 and the training procedure was repeated for 2000 times in each execution.

We have used annual average vehicle flow data on different roads in a city to predict flow values on minor roads in a city in UK. We have evaluated the results of all 20 executions of our model to see the variation in the accuracy and error rate. This gives a better idea about the performance of deep learning model and we calculate the average accuracy rate. For performance analysis, we have used mean absolute error (MAE), and mean absolute percentage error (MAPE). MAE is used to shows the closeness between the actual and the predicted values and MAPE shows the relative difference between the actual and the predicted values. MAPE is not suitable to calculate error rate if the input data or actual values contain zeros because in this case it suffers from the division by zero error. MAE and MAPE values are calculated using the following equations.

$$Mean\,Absolute\,Error\,(MAE) = \frac{1}{N}\sum_{i=1}^{N} V_i - P_i$$

$$Mean\,Absolute\,Percentage\,Error\,(MAPE) = \frac{1}{N}\sum_{i=1}^{N} \frac{V_i - P_i}{V_i}$$

Here N is the size (number of values predicted by the model) of dataset used for prediction purpose, V is the set of actual values used as labels, and P is the set of values predicted by our deep learning model.

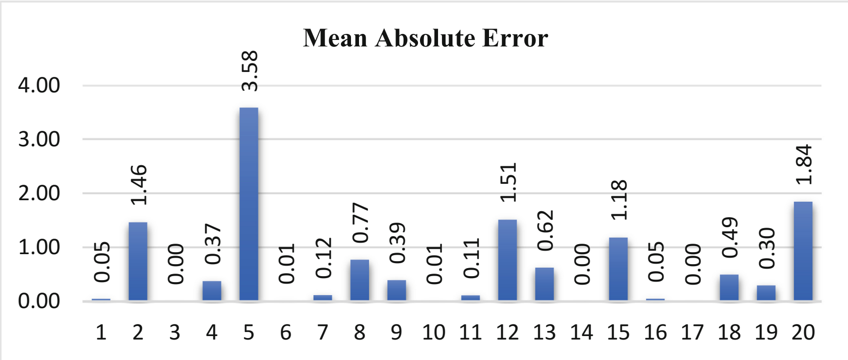

Fig. 4. Mean absolute error

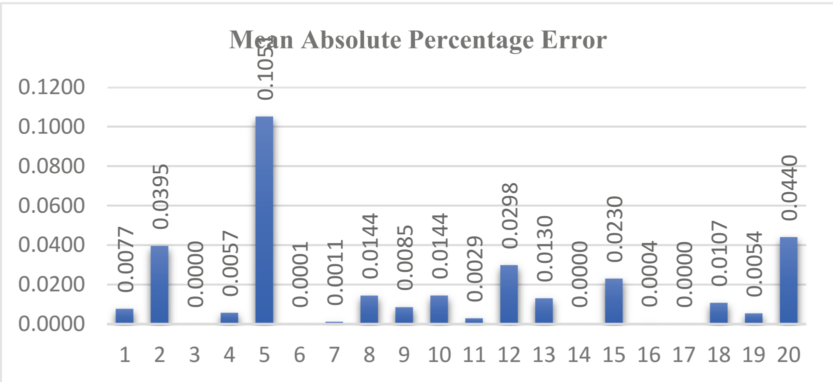

Fig. 5. Mean absolute percentage error

In Fig. 4, we have shown the results obtained by executing our deep model 20 times. In this graph, *x-axis* shows the number of model and it ranges from 1 to 20, and *y-axis* shows the MAE values calculated by using the given equation. Graph shows that error rate was very low because the maximum error value calculated was for model 5 and it was 3.58, and in some cases, it was as low as zero. Here zero does not mean that prediction was exactly the same, but it shows that the values were very close and there was not a big difference between the original and the predicted values.

In Fig. 5, we have shown the results calculated by using the mean absolute percentage error. Same as MAE, we have calculated MAPE for all 20 executions and prediction results of our deep learning model. Maximum MAPE value is 0.105 for 5th execution of our model with the same configurations. MAPE is considered a best measure to the data where there are no extremes and our data also contains a relatively balanced set of flow values. Therefore, our MAPE values describe that the predicted results have very low error rate and predicted values are very close to the original flow values.

In addition to the graphs showing error rates using MAE and MAPE, we have plotted the actual and predicted flow values to show the difference between patterns as well. Our MAE and MAPE values shows that the actual and predicted values are very close. If this is true, then the graphs of both plotted values should show the similar trends. In Fig. 6, we have plotted the first 100 actual and the predicted flow values. In this graph, *y-axis* shows the flow values. As both, actual and predicted values are very close, so graph is drawn by doubling the predicted values to avoid the overlapping of both curves. Both the curves show that these are not same but follow a similar trend. This shows that the predicted values are following the same trend that was followed by the input flow data with slight differences.

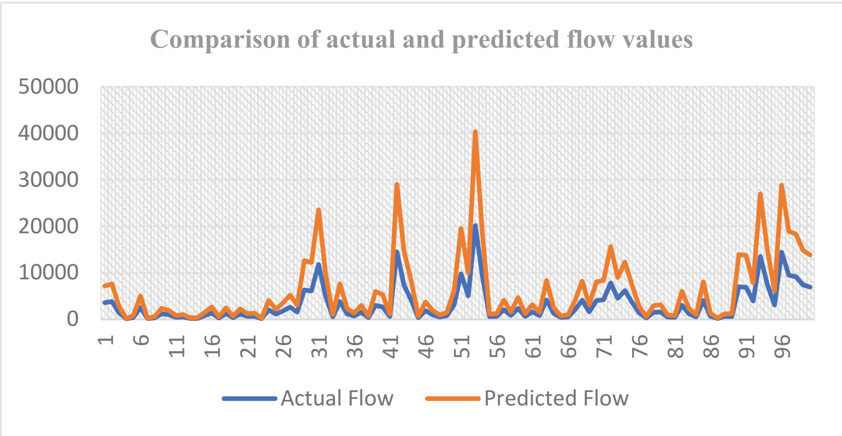

Fig. 6. Comparison of first 100 actual and predicted flow values.

Similarly, to analyze the pattern in depth, we have selected a range of actual flow values from 1 to 500, i.e. we have selected only those results where actual flow values are in the range of 1 to 500. The purpose of selecting this range is to see the trends when flow values were uniform and thus input data values were very close. This is shown in Fig. 7. Again, the predicted values are doubled to avoid overlapping of both curves representing the flow values. This graph also shows similar graph for both,

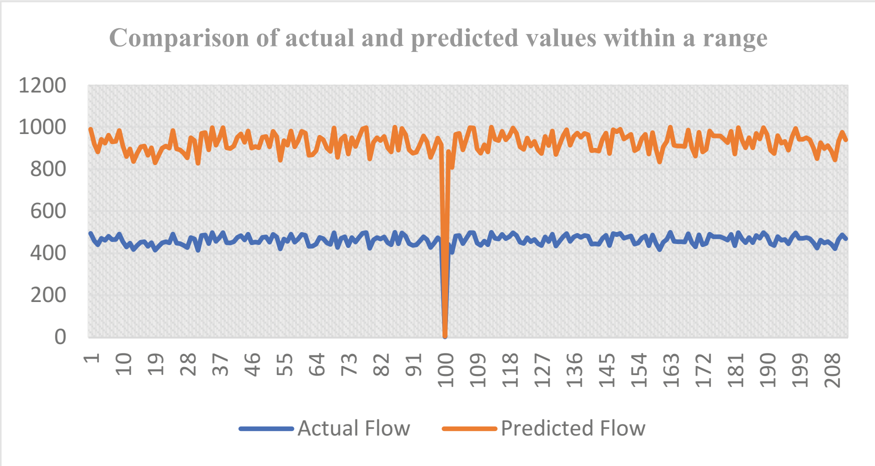

Fig. 7. Comparison actual and predicted values when flow is less than 500.

actual and predicted flow values with not big differences. In this graph, we have selected values within a range, therefor it is expected for good prediction results that the output values should also be in a specific range as shown in this graph. So, we can say that predicted values have followed the trend that was present in the input dataset. Therefore, the accuracy rate is high and low MSE and MAPE rates are reported.

7 Conclusion and Future Work

In this work we have used deep learning approach to manage traffic flow in smart cities for disaster management. Deep learning requires a large amount of data for training purpose that could easily be accessed from the traffic departments in smart cities. In this work we have used historic traffic data to predict the traffic flow and its behaviour in disaster. Results shows very high accuracy rate but this is because of the high correlation between the input data and the output values. Results may differ when same deep learning model is applied on a different type of data. We have plotted MSE and MAPE results for all 20 executions of our model with the same specification. Results shows that a specific accuracy rate was maintained in all 20 executions of our model and thus we can say that its output is consistent to a certain extent. In addition to error rates, we have plotted the original and predicted flow values to visualize the difference between the graph trends followed by actual and predicted values graphs. Graphs also show similar trends and proves that there are not big differences between the actual and the predicted values. As mentioned earlier, we mainly have focused in this paper on providing details of the deep learning based traffic prediction approach. Details of the overall evacuation method can be found in our earlier work [3–5].

Although, we have shown excellent results in this work, but this is not guaranteed while working with other traffic data with same or other deep learning models. This could be the result of high uniformity in input data that was used for training and testing purposes and therefore the same performance of deep model could not be guaranteed for other datasets. Therefore, we aim to work on different data with many other features including incidents data etc. to see its impact. This may also help us in predicting the people and other stakeholders behavior in emergency situations and we may model them collectively to present a model to not only to manage traffic by flow values but also by including other important factors in that environment as well. We can also use real-time traffic and other data to present an effective traffic management plan in the effected areas and can also use big data technologies to deal with real-time data.

Acknowledgments. The authors acknowledge with thanks the technical and financial support from the Deanship of Scientific Research (DSR) at the King Abdulaziz University (KAU), Jeddah, Saudi Arabia, under the grant number G-673-793-38. The work carried out in this paper is supported by the High Performance Computing Center at the King Abdulaziz University, Jeddah.

References

1. Gharaibeh, A., Salahuddin, M.A., Hussini, S.J., Khreishah, A., Khalil, I., Guizani, M., Al-Fuqaha, A.: Smart cities: a survey on data management, security, and enabling technologies. IEEE Commun. Surv. Tutor. **19**(4), 2456–2501 (2017)
2. Su, K., Li, J., Fu, H.: Smart city and the applications. In: 2011 International Conference on Electronics, Communications and Control (ICECC), pp. 1028–1031. IEEE (2011)
3. Alazawi, Z., Altowaijri, S., Mehmood, R., Abdljabar, M.B.: Intelligent disaster management system based on cloud-enabled vehicular networks. In: 2011 11th International Conference on ITS Telecommunications, pp. 361–368. IEEE (2011)
4. Alazawi, Z., Abdljabar, M.B., Altowaijri, S., Vegni, A.M., Mehmood, R.: ICDMS: an intelligent cloud based disaster management system for vehicular networks. In: Vinel, A., Mehmood, R., Berbineau, M., Garcia, C.R., Huang, C.-M., Chilamkurti, N. (eds.) Nets4Cars/Nets4Trains 2012. LNCS, vol. 7266, pp. 40–56. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29667-3_4
5. Alazawi, Z., Alani, O., Abdljabar, M.B., Altowaijri, S., Mehmood, R.: A smart disaster management system for future cities. In: Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities - WiMobCity 2014, pp. 1–10. ACM Press, New York (2014)
6. Alam, F., Mehmood, R., Katib, I., Albeshri, A.: Analysis of eight data mining algorithms for smarter Internet of Things (IoT). Procedia Comput. Sci. **98**, 437–442 (2016)
7. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: a review. Neurocomputing **187**, 27–48 (2016)
8. Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two, pp. 1237–1242 (2011)
9. Djahel, S., Salehie, M., Tal, I., Jamshidi, P.: Adaptive traffic management for secure and efficient emergency services in smart cities. In: 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 340–343. IEEE (2013)
10. SUMO | Simulation of Urban MObility. http://sumo.dlr.de/wiki/Main_Page

11. Suma, S., Mehmood, R., Albugami, N., Katib, I., Albeshri, A.: Enabling next generation logistics and planning for smarter societies. Procedia Comput. Sci. **109**, 1122–1127 (2017)
12. Schlingensiepen, J., Nemtanu, F., Mehmood, R., McCluskey, L.: Autonomic transport management systems—enabler for smart cities, personalized medicine, participation and industry grid/industry 4.0. In: Sładkowski, A., Pamuła, W. (eds.) Intelligent Transportation Systems – Problems and Perspectives. SSDC, vol. 32, pp. 3–35. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-19150-8_1
13. Alam, F., Mehmood, R., Katib, I., Albogami, N.N., Albeshri, A.: Data fusion and IoT for smart ubiquitous environments: a survey. IEEE Access **5**, 9533–9554 (2017)
14. Mehmood, R., Alam, F., Albogami, N.N., Katib, I., Albeshri, A., Altowaijri, S.: UTiLearn: a personalised ubiquitous teaching and learning system for smart societies. IEEE Access **5**, 2615–2635 (2017)
15. Tawalbeh, L., Basalamah, A., Mehmood, R., Tawalbeh, H.: Greener and smarter phones for future cities: characterizing the impact of GPS signal strength on power consumption. IEEE Access **4**, 858–868 (2016)
16. Dandala, T.T., Krishnamurthy, V., Alwan, R.: Internet of Vehicles (IoV) for traffic management. In: 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), pp. 1–4. IEEE (2017)
17. Rizwan, P., Suresh, K., Babu, M.R.: Real-time smart traffic management system for smart cities by using Internet of Things and big data. In: 2016 International Conference on Emerging Technological Trends (ICETT), pp. 1–7. IEEE (2016)
18. Schlingensiepen, J., Mehmood, R., Nemtanu, F.C.: Framework for an autonomic transport system in smart cities. Cybern. Inf. Technol. **15**, 50–62 (2015)
19. Arfat, Y., Aqib, M., Mehmood, R., Albeshri, A., Katib, I., Albogami, N., Alzahrani, A.: Enabling smarter societies through mobile big data fogs and clouds. Procedia Comput. Sci. **109**, 1128–1133 (2017)
20. Zhu, F., Li, Z., Chen, S., Xiong, G.: Parallel transportation management and control system and its applications in building smart cities. IEEE Trans. Intell. Transp. Syst. **17**, 1576–1585 (2016)
21. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.-Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
22. Caltrans PeMS. http://pems.dot.ca.gov/
23. Polson, N.G., Sokolov, V.O.: Deep learning for short-term traffic flow prediction (2017)
24. Polson, N., Sokolov, V.: Deep learning predictors for traffic flows (2016)
25. Ma, X., Yu, H., Wang, Y., Wang, Y.: Large-scale transportation network congestion evolution prediction using deep learning theory. PLoS ONE **10**, e0119044 (2015)
26. Keras Documentation. https://keras.io/