Chapter 7
A Smart Disaster Management System for Future Cities Using Deep Learning, GPUs, and In-Memory Computing

Muhammad Aqib, Rashid Mehmood, Ahmed Alzahrani, and Iyad Katib

7.1 Introduction

Smart cities appear as "the next stage of urbanization, subsequent to the knowledge-based economy, digital economy, and intelligent economy" [1, 2]. Smart cities aim to "not only exploit physical and digital infrastructure for urban development but also the intellectual and social capital as its core ingredient for urbanization"[1, 2]. Smart cities are driven by, or involve, integration of multiple city systems, such as transport, healthcare, and operations, and hence are considered a major driver for the transformation of many industries [2, 3]. Smart society is an extension of the smart cities concept, "a digitally-enabled, knowledge-based society, aware of and working towards social, environmental and economic sustainability" [2]. A recent book has covered a number of topics related to smart cities and societies [4].

Smart cities rely on dynamic monitoring and management of city assets and systems and this generates data [5, 6] of diverse characteristics, known as big data. Formally, big data refers to the "emerging technologies that are designed to extract value from data having four Vs characteristics; volume, variety, velocity and veracity" [7]. Big data leverages distributed and high performance computing (HPC) technologies to manage and analyze data. These two technologies (big data and HPC) are converging to address their individual limitations and exploit their synergies [8].

M. Aqib · A. Alzahrani · I. Katib
Department of Computer Science, FCIT, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: mpervez@stu.kau.edu.sa; asalzahrani@kau.edu.sa; iakatib@kau.edu.sa

R. Mehmood (✉)
High Performance Computing Center, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: RMehmood@kau.edu.sa

R. Mehmood et al. (eds.), *Smart Infrastructure and Applications*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-13705-2_7

Smart cities must be equipped with disaster and emergency management systems to manage manmade and natural calamities such as floods, hurricanes, earthquakes, fires, and terrorist attacks. Disasters not only result in loss of human lives but could also damage the economy. For example, the June/July 2018 floods in Japan left around 200 people dead and many injured. Millions of people were ordered to evacuate the affected areas, and thousands were transferred to temporary shelters [9, 10]. Rescue teams comprised of workers from civil defense and many other organizations. They worked round-the-clock to overcome the disastrous situation in the affected areas. The cost of flood rebuilding was estimated to be $2bn [11]. The Barcelona terrorist attack of August 2017 resulted in 24 deaths and 152 injured people. Earlier, the cost of 2011 Japan earthquake and tsunami disaster alone was in excess of 200 billion USD in addition to the irrecoverable loss of over 18 thousand lives [12].

The advent of many new technologies has improved our ability to manage disaster situations. Many governments around the world are applying these new techniques and technologies to minimize the effects of these disasters. Plans are made to respond not only during the disaster situation, but also after the disaster, and more importantly, how to prevent or minimize the effects of disasters before its occurrence.

Mobility plays a key role in effectively managing disaster situations [13]. Smart mobility requires smart transportation infrastructures [14, 15]. Many approaches have been developed to improve transportation. These include, for example, social media based approaches [16–18], big data based techniques [15, 19, 20], HPC based techniques [15, 19, 21, 22], vehicular networks (VANETs) and systems [23–26], modeling and simulations [27, 28], methods to improve urban logistics [15, 19, 21, 29, 30], and solutions based on autonomous vehicles and autonomic mobility systems [31–34].

Smart mobility allows smooth evacuation of people from the affected areas by dynamically monitoring the disaster-affected areas as well as the other adjacent areas to avoid road congestion, blockages, and chaos. Traffic data is collected from various static and mobile sensors including those deployed under and over the road networks. These could include inductive sensors, motorway incident detection and automatic signaling (MIDAS) loops, GPS sensors, VANETs, cameras, image processing systems, and many more. The collected data is analyzed to monitor traffic flow and other metrics, and is used to devise navigation strategies to provide emergency services and smooth evacuation from the affected areas, avoiding congestion, minimizing risks to public safety, and economic losses.

Our research focuses on using emerging technologies to develop cutting edge solutions for disaster management. We have proposed a cloud computing based disaster management system along with its implementation in [13]. The work was extended in [35] leveraging VANETs to sense traffic related information and propagate navigation instructions. These works were based on macroscopic modeling. Further improvements to the disaster management system were proposed in [12] where microscopic modeling was used to improve and validate the earlier results. Moreover, different evacuation strategies were used to investigate the performance of evacuation operations on the proposed disaster management system. Further extensions of the proposed system were reported in [36, 37] using various evacuation strategies.

The availability of various data related to smart environments, generated, for instance, by the internet of things (IoT), and the advancements in artificial intelligence (AI) has provided new opportunities for data-driven studies (see, e.g., [2, 3, 38]). Deep learning has emerged as a promising AI technology with reportedly higher prediction accuracy, albeit higher computational costs [39]. In [22], we extended our work by using deep learning to predict traffic plans for evacuation in disaster situations. We had used in-memory computations and graphics processing units (GPUs) to address intensive and timely computational demands in disaster situations.

This paper extends our earlier work and provides extended analysis and results of the proposed system. A system architecture based on the in-memory big data management and GPU-based deep learning computations is proposed. The background technologies have been elaborated. An extended literature review is provided. We have used road traffic data made publicly available by the UK Department for Transport (DfT). The results show the effectiveness of the deep learning approach in predicting traffic behavior in disaster and city evacuation situations. To the best of our knowledge, this is the first proposal where deep learning, in-memory data-driven computations, and GPUs are brought together for timely, compute intensive, predictions of road traffic in disaster situations.

The rest of the paper is organized as follows: Sect. 7.2 provides background material introducing the tools and technologies used in our work. The related work is discussed in Sect. 7.3. Our proposed framework is introduced in Sect. 7.4. In order to find suitable city data, we have examined a number of datasets and their details are given in Sect. 7.5. These could be useful for the researchers working in related areas. Performance evaluation and analysis of the proposed system is given in Sect. 7.6. Finally, in Sect. 7.7, we conclude the paper with directions for the future work.

7.2 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.

7.2.1 Graphical Processing Units

In this section, we will give an overview of the GPU architecture. A GPU chip contains multiple multi-processors (MPs) and each MP contains many stream-processors (SPs). 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 execute 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 are also part of the discussion.

7.2.2 In-Memory Computing

For computation purposes, data is normally stored on disks that provide the facility to store a large amount of data. In addition to disks, other memory storage components are also used for this purpose that include registers, cache, and main memory. These storage components differ in size and also in performance. Registers are the smallest ones in terms of capacity to store data but these are the most efficient in terms of speed. Then there are caches and main memory in this hierarchy, which are much smaller than disks but provide higher speed to access data. In recent years, the main memory size has also been increased and its cost is also decreasing that make it possible to use large amount of main memory to perform compute intensive tasks. Due to increased size, it is capable to hold a large amount of data as well, thus making it easy for the programs to access that data on low I/O costs. In-memory computing also supports the technique to store the data required for processing on the main memory instead of storing it on the disks. This idea was introduced a long ago but high price and availability of low storage capacities were the constraints in using in-memory technique to deal with large amount of data. Now using in-memory, a large amount of data could be stored into the main memory for processing. In case of big data, where data size is large enough so that it could not be stored in main memory of a single computer, it is normally distributed among multiple nodes in a cluster of compute nodes and each node is assigned a block of data according to its capacity. Many frameworks exist that distribute the data to all the nodes in the cluster to be stored on the main memory and then processed. The results generated by the individual nodes are then combined to generate unique output.

Due to increase in cost of energy and increase in its demand as compared to the production rate, researchers are now finding the ways to optimize the existing systems or methods to develop the new energy efficient systems. The authors in [40] have studied the role of database software in order to improve the efficiency of a server. According to them, among the nodes in a scale-out architecture, the highest performing one is considered as the most energy efficient configuration. Also the power consumed by different operators like joins, sorts, etc., varies and also the CPU power consumption and its utilization are not linearly related to each other. In a survey of the energy efficiency techniques [41] have focused on the characteristics of the two main power management technologies: static power management (SPM) systems and dynamic power management systems (DPM). The article presents a brief discussion on the techniques proposed by researchers to reduce the power consumption in cluster computing systems. The pros and cons of both the methodologies have been discussed in detail. Non-volatile memory has great importance in main memory data management systems. But it has many issues as well and energy consumption is one of them. A technique has been proposed in [42] that deals with the high energy consumption rate during write operations in phase change memory (PCM). A solution based on out of position PCM write operations has been proposed that reduces power consumption, however, degrades the system performance. PDRAM [43] is another approach for in-memory data management systems based on the phase change random access memory (PRAM) and DRAM. The authors have proposed an approach that deals with the low read and standby power and DRAM has low write power by providing a hybrid hardware software solution. Some other techniques that do not suggest the storage of whole data in main memory also propose a mechanism to store the data needed for computation in main memory. Such a technique [44] proposes the bulk copy and initialization completely in the DRAM, which in return reduces the data transfer over the memory channels and thus saves energy. The proposed technique is named as RowClone and it copies the complete row of data from source to a row buffer and then from the buffer to the destination. As part of semi-structured data processing, SAP HANA provides the facility to process graphs data.

7.2.3 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 supervised learning and unsupervised learning techniques. Artificial neural network, clustering, genetic algorithms, and deep learning 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 [45]. In this work, we are using CNNs for training and testing purposes. So, we will discuss them in detail in the following paragraph.

Convolutional Neural Networks (CNNs) In the CNNs, multiple layers including convolutional, pooling, and connected layers are used for training purpose in a robust manner. The authors in [45] have defined a general architecture of CNN for image classifications. The whole process is divided into two main phases: forward phase that includes convolutional and pooling layers and backward phase where fully connected layers are used to produce the output.

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 subsampling layers are realized and how the nets are trained [46].

7.2.4 Microscopic Models and Tools

In this section, we will discuss the microscopic model that is used in traffic management works. Although, instead of using these models or any other related simulation tools, we are using deep learning to forecast traffic plans in disaster situations but here we are giving a brief introduction about other techniques to give an overview of these models to the readers.

Lighthill–Whitham–Richards (LWR) model [47, 48] is a macroscopic model that could be used to analyze the traffic behavior in roads. It uses some traffic data characteristics like speed, flow, and density. This model could be derived from the following equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0 \tag{7.1}$$

Here ρ is the traffic density, x is the distance, t is the time, and u is the speed to travel x distance in t time. Now using the Greenshields' model [49], the relation between the density (ρ) and speed (u) could be defined as follows:

$$u(\rho) = u_{max} = \left(1 - \frac{\rho}{\rho_{max}}\right) \tag{7.2}$$

Here u_{max} is the maximum speed, and ρ_{max} is the maximum density. So, by using this model, the relationship between flow, density, and speed could be given as

$$\text{flow} = \text{density} \times \text{speed} \tag{7.3}$$

To model these microscopic models, a number of simulators are available for this purpose. In the following paragraphs, we will discuss about two of them which are used by researchers to carry out their research work. MITSIM [50] is a microscopic traffic simulator that uses the information related to road network, surveillance system, traffic signs and signals, etc. It classifies the lanes according to its speed limit, regulations, and also simulates loop detectors, lane use signals, etc. As an input, an origin–destination table, traffic control, and route guidance logic are used. Here vehicles are considered to move between their origin and destination and it collects the sensor readings that include traffic count, occupancy, and speed of vehicles at given intervals of time. To simulate the vehicle movement in a network, two models are used that are: acceleration model and lane changing model [51].

Simulation of urban mobility (SUMO) [52] is another microscopic and continuous road traffic simulation package that deals with the large road networks. It provides the users the facility to define their own network through the use of data configuration files. Normally input data is given in the form of XML files where different nodes having different parameter values define different configuration values. It also provides the facility to generate real world scenario by selecting the area on a map in a browser by running a program included in package.

7.3 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, both are useful for us and therefore we are presenting some approaches for better understanding of the work done in this area.

A deep learning approach to predict traffic flow for short intervals on road networks is proposed in [53]. A traffic prediction method based on long short-term memory (LSTM) has been used by the authors for prediction purpose. An origin–destination correlation (ODC) matrix has been used as input to the training algorithm. Dataset used for this process is collected from the Beijing Traffic Management Bureau and it is collected from more than 500 observation stations or sensors containing around 26 million records. A 5-min interval data from Jan 1, 2015, to June 30, 2015, has been collected where the data for the first 5 months has been used for training and the rest of the data is used for testing purposes. For evaluation of proposed model, mean absolute error (MAE), mean square error (MSE), and mean relative error (MRE) have been calculated. Input data has been used to predict the flow in 15, 30, 45, and 60 min time intervals. The authors in this work have selected three observation points with high, medium, and low flow rates to compare the actual flow and predicted flow values on those observation points. MRE values for a 15-min interval flow prediction reported in this work are

6.41, 6.05, and 6.21%. They have compared the result with the other approaches including RNN, ARIMA, SVM, RBF, etc., and concluded that for time interval less than 15 min, RNN is relatively accurate, but with big time intervals, error increases, but overall it performs better than other old machine learning models. Therefore, it is concluded that LSTM is an appropriate choice for long time intervals.

Yu et al. in [54] also have proposed an approach that uses deep learning for vehicles' speed prediction on highways in peak hours and post-accident conditions. In this work, the authors have used the long short-term memory (LSTM) recurrent neural networks for prediction purposes. In addition to LSTM, they also have used autoencoders whose output is also used in their deep LSTM model. This model is named as "mixture deep LSTM". For this purpose, they have used the data from the 2018 loop detectors (sensors) in Los Angeles County during the period starting from May 19, 2012, to June 30, 2012. This provides data collected from around 5400 miles long highways cumulatively. Also, as they are predicting the speed after accidents as well, so accidents data, for this purpose, has been collected from various sources including California Highway Patrol, California Transportation Agencies, etc. Normalized data including 5 min aggregated speed values, day time, and weekdays has been used in this work. To deal with the missing values, data collected from the sensors with more than 20% missing values is excluded from the datasets. Also, no criteria is defined to deal with the missing values and the estimated speed values for missing input values have been excluded from the predicted output datasets and have not been considered for evaluation. For analysis purposes, mean absolute percentage error (MAPE) has been used. Performance of the proposed is compared with other methods like ARIMA, random walk, and historical average, etc. Speed in peak hours has been predicted using four different time intervals of 5, 15, 30, and 60 min. The results show that the highest accuracy is achieved for small time interval, i.e., 5 min. It is around 6 in peak hours and around 5.5 in off-peak hours. Error rate increases with the increase in time interval but in all the four intervals, deep LSTM performs much better than other techniques.

In another work, Jia et al. have used a deep learning approach called deep belief networks (DBN) to predict the vehicles' speed on a road network in [55]. In this work, they have used restricted Boltzmann machines (RBMs) for unsupervised learning and then have used the labeled data for fine tuning. Dataset used in this purpose is obtained from Beijing Traffic Management Bureau (BTMB). Three months data (June–August 2013) has been used that provided 2-min interval data collected from the detectors installed on a specified segment of road in Beijing, China. Around 11-week data is used for training purpose, whereas the remaining last week's data is used for testing purpose. This provides 2-min interval speed, flow, and occupancy values and by using this 2-min interval data, the authors have predicted speed for intervals of 2, 10, and 30 min. Furthermore, for performance analysis, three performance metrics have been used: mean absolute percentage error (MAPE), root mean squared error (RMSE), and normalized root mean squared error (RMSN). No mechanism is mentioned by authors to deal with the erroneous or missing data values and also no big data technology is used for data management. Also, no specific information about data, e.g., number of detectors, etc., is given to know about the size of data. For deep model configurations, they have executed the model with different configuration and based on the MAPE values, best configurations have been selected. With best selected configurations, MAPE value for 2-min interval is 5.81, 7.33 for 10 min, and its value is 8.48 for 30-min interval. This shows that it performs better for short time intervals and cannot cope with the stochastic fluctuations in long time intervals. Although results are quite good for speed prediction, but it still need to investigate how it behaves when some other information are included, e.g., we do not know whether data from multiple detectors has been used or separate data for each detector is used because in the former case we get more fluctuations in data as compared to the latter case. Also, the size of data could also change the results.

The authors in [56] 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, which 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 [52] 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 [16] and autonomic transportation [31]. Internet of things (IoT) could be considered as the back bone of future smart cities [38]. Mehmood et al. [2] propose 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 [1] is collected for traffic management. There are many studies that use IoT and IoV to propose a traffic management plan as in [57, 58]. In addition to this, a lot of work has been done in the area of autonomic transport management in smart cities [33]. The work in [30] also shows the importance of fog and other cloud technologies in dealing with emergency situations in smart cities. In [59] a parallel transportation management and control system for smart cities has been presented that not only uses the artificial intelligence technologies but also uses massive traffic data and big data technologies or frameworks like MapReduce. This shows the importance of these technologies in traffic management in smart cities.

A traffic flow prediction approach has been proposed in [60]. The 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) [61]. 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 the same freeway were treated as different freeway. Support vector machines (SVM) have been used for comparison purpose.

The authors in [62] 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. The 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 [61]. In addition to this, the 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 2 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 show 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 [63] to model the traffic flow. In this work, the 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 autoregressive model for predictors selection. These predictors are later used to build a deep learning model. Stochastic gradient descent (SGC) method is used to know the structure and weights of parameters. They also have applied three filtering techniques (exponential smoothing, median filter, and loess filter) on traffic data to filter noisy data from the sensors. Data for this purpose is collected from 21 loop detectors on 5-min 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 the 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.

The authors in [64] 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, the authors have used support vector machines (SVMs) and found that prediction accuracy was increased by at least 17%.

7.4 Disaster Management System

In this section, we will discuss the proposed deep learning based disaster management system in detail. Figure 7.1 depicts the architecture of our proposed system. The proposed framework consists of three main layers: input layer, data processing layer, and prediction layer. A general framework was given in our previous work [22] as shown in Fig. 7.2. In this work, we have presented the complete architecture that gives details about each layer and the components in each layer. In the following sub-sections, we will discuss these layers in detail.

7.4.1 Input Layer

Input layer manages the traffic data that is used for 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. The role of input layer is to collect data from the source and to forward it to processing layer. In case of

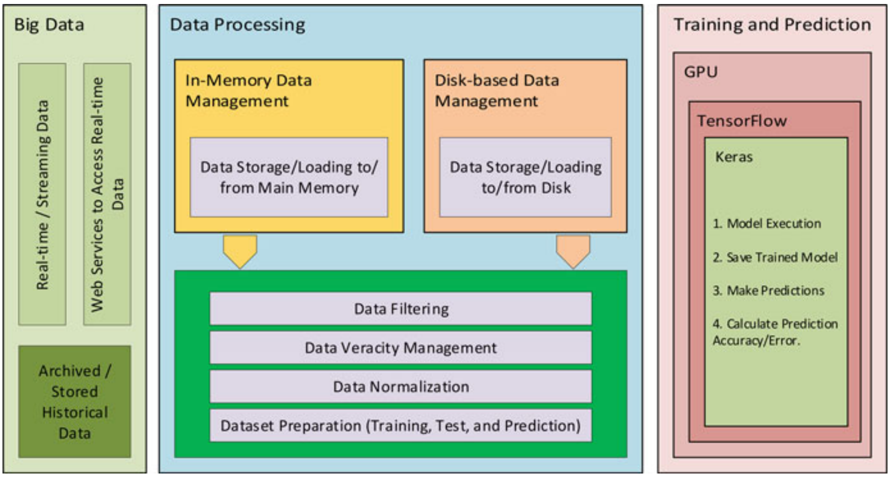

Fig. 7.1 System architecture for prediction of traffic plan using deep learning

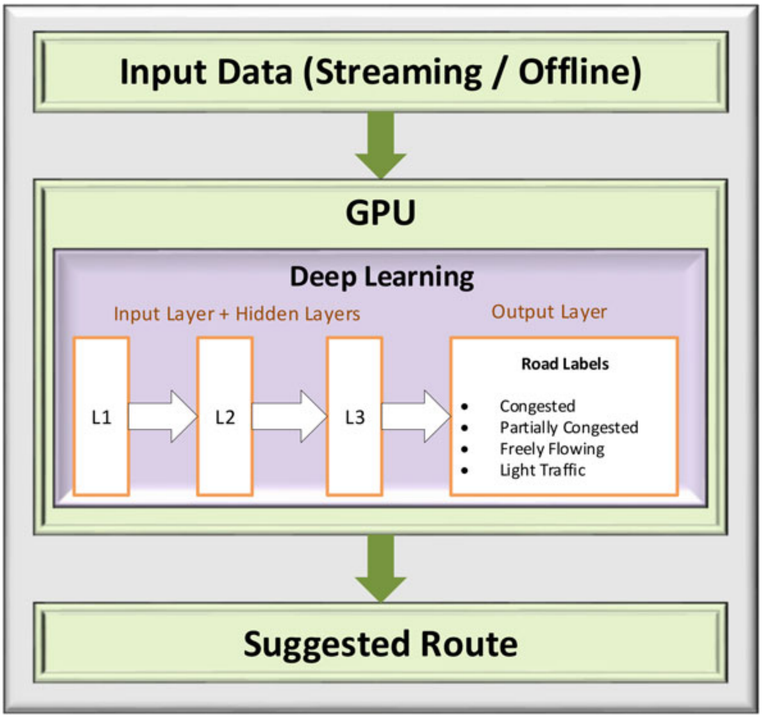

Fig. 7.2 The proposed disaster management framework

offline or historical data, the data is collected from the source and then stored on a disk drive so that it could be forwarded for processing layer. 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, by using the APIs provided by the data generating source or web services, and forwarded it to the processing layer in real-time for further data formatting.

7.4.2 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 processes 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 attributes could be useful in training process, e.g., we can get peak hours, and can separate the data based on weekends, etc. Different big data related issues like dealing with data veracity are also resolved in this layer. Because the data collected from the sensors or other devices is not guaranteed to be free of veracity issues. For example, due to malfunctioning in recording device, the recorded values may be incorrect, or missing, etc. So in this layer, we have a mechanism to deal with the erroneous data. For this purpose, well-known techniques are applied to ensure the correctness of data. Furthermore, we may need to normalize the input data for our regression model. So, data normalization is also performed in this layer.

7.4.3 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 7.3 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

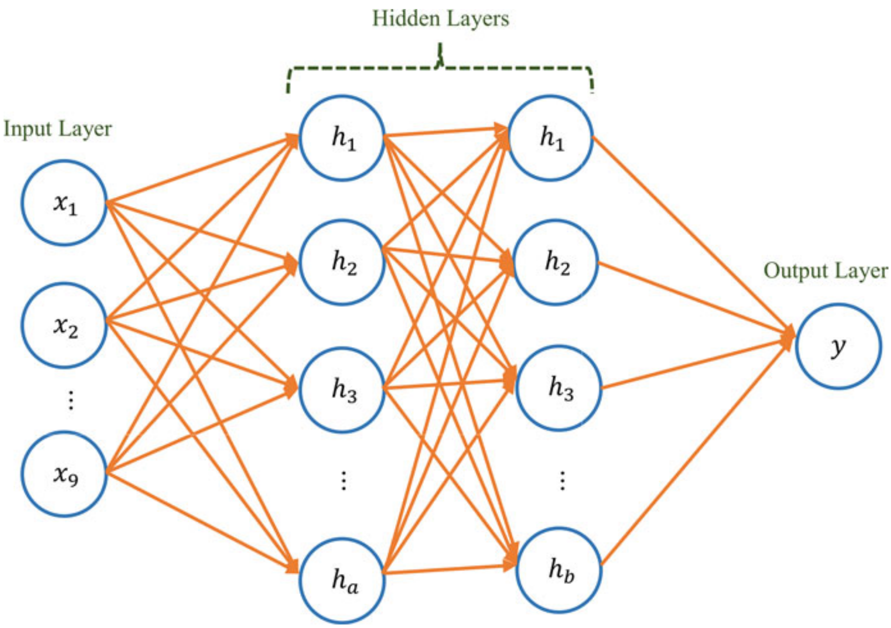

Fig. 7.3 Our deep neural network with two hidden layers

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.

7.5 Datasets

In this work, we are mainly working on the UK traffic data. So, we have explored a variety of traffic data available through multiple sources in the UK that could be used for different purposes to work on traffic management plans. Some data sources of same kind outside the 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 7.1, we have given some data sources that provide traffic data. Short data description and URLs to access the data are also given.

7.6 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.

7.6.1 Deep Model Setup

In this work, we have used vehicles flow data on minor roads in a city in the 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 70,470 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 [65] which uses TensorFlow library [66] at the backend. 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, and 1 for training, testing, and prediction purposes, respectively. Batch size was set to 10 and number of epoch was set to 1000.

Table 7.1 The UK traffic data sources

S.No	Data source	Description
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 UK government agencies could be accessed from this portal. Its transport data section provides many options to explore traffic data. 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. http://data.dft.gov.uk/
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
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
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	Website Developer.here.com	Provides API to get traffic flow and incidents data. https://developer.here.com/

Table 7.2 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

7.6.2 Input 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 7.2, we have given the schema of input dataset that provides brief description of some important input attributes in this dataset.

7.6.3 Performance Metrics

For performance analysis, we have used mean absolute error (MAE) and mean absolute percentage error (MAPE). MAE is used to show 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 by using Eqs. (7.4) and (7.5), respectively.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|V_i - P_i| \tag{7.4}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|V_i - P_i|}{V_i} \tag{7.5}$$

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.

7.6.4 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 [12, 13, 22, 35]. We have executed our deep model with different configurations and with different input dataset sizes. Furthermore, we have divided the analysis process in different phases where we have used different model configurations and different dataset distribution sizes to compare the results for analysis purpose.

In the first phase of analysis process, 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. We have reserved the data for prediction purpose, because, after running the model for training and purpose, we saved the model and the specified amount of data was used as input to the saved model to predict the output. This enabled us to compare the results produced by analyzing the testing dataset and the results calculated by us by analyzing the values produced by the saved model by using the prediction dataset. In addition to this, our deep learning model with one configuration setup was executed for 20 times to get results for analysis purpose. Furthermore, for all the 20 models with the same configurations, 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 the 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.

In Fig. 7.4, 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

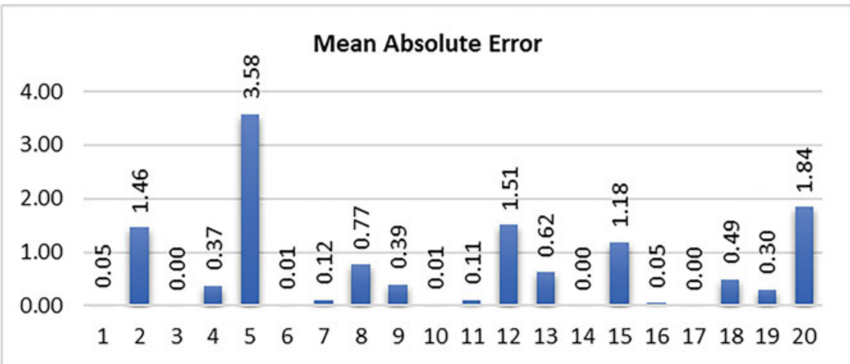

Fig. 7.4 Mean absolute error

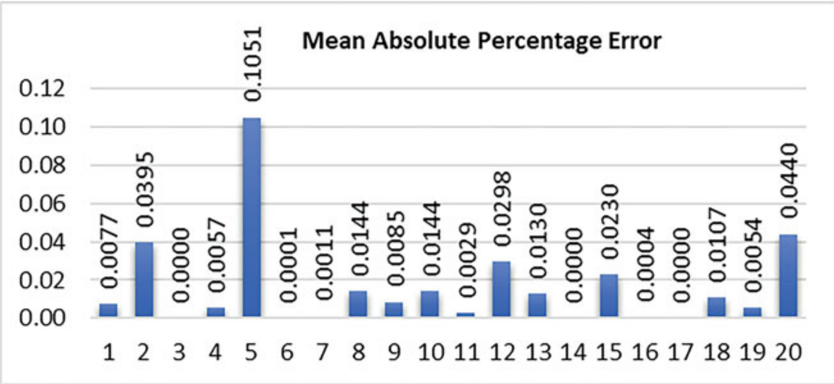

Fig. 7.5 Mean absolute percentage error

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. 7.5, 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 show 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. 7.6, 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, 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.

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 1–500. 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. 7.7. Again, the predicted values are doubled to avoid overlapping of both curves representing the flow values. This graph also shows

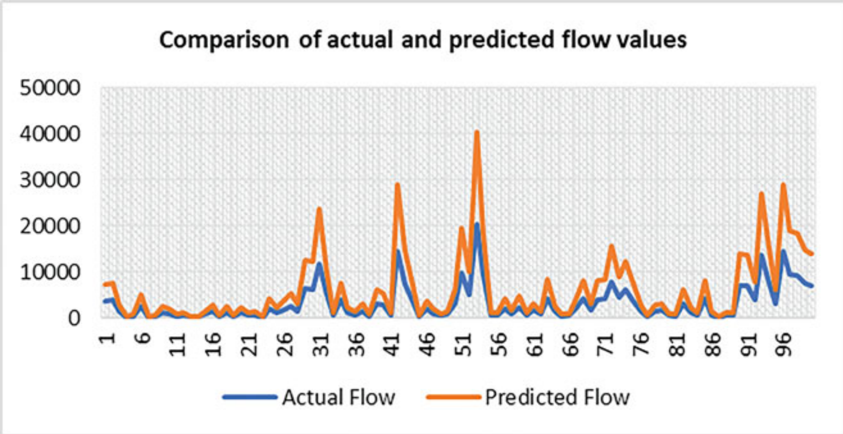

Fig. 7.6 Comparison of first 100 actual and predicted flow values

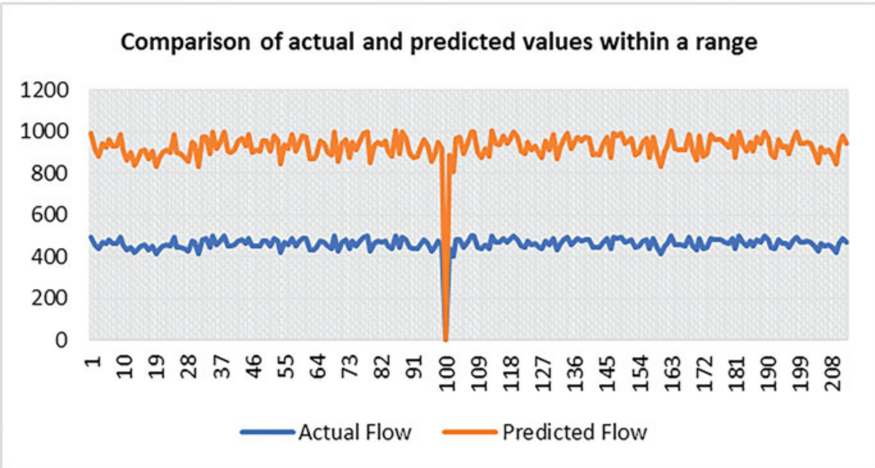

Fig. 7.7 Comparison of actual and predicted values when flow is less than 500

similar graph for both, actual and predicted flow values with not big differences. In this graph, we have selected values within a range; therefore, 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.

To show the accuracy of our predict results, we also have compared the actual and predicted values. We have calculated the maximum difference between the actual and the predicted values. The main purpose to calculate the maximum difference between the actual and the predicted values in each model execution is that it clearly

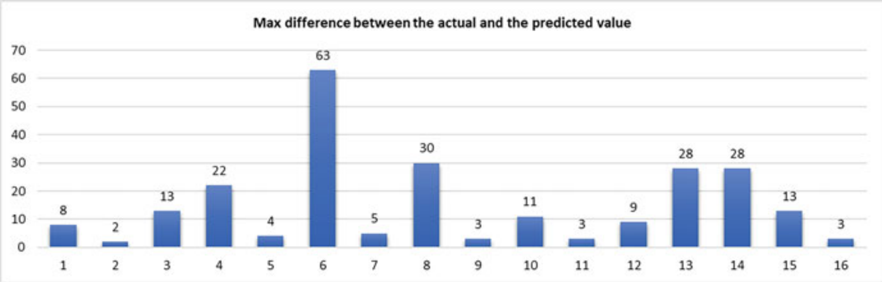

Fig. 7.8 Maximum difference between the actual and predicted vehicles flow values (phase 1)

shows whether the predicted values predict the number of vehicles that match the ground reality or it is far away from the actual values. Maximum difference between the actual and predicted vehicles flow values in each model execution is shown in Fig. 7.8.

In this figure, we have compared the available predicted values for the 16 executions of same deep model on the same input dataset. From this graph, it is clear that the minimum value for the maximum difference is 2, which shows that results obtained in this model execution were very close to the original values and we can say that it can be used to represent the actual data. On the other hand, the maximum value while calculating the maximum difference is 63, which can be used to represent the actual values if the actual vehicles flow value was very big, e.g., say 1000, but if in actual, there were only 100 vehicles on the road, then the difference of 63 between the actual and the predicted values represents the inaccuracy of predicted results that cannot be used to represent the actual values.

In this phase the distribution of the dataset for training and testing processes was changed to 60% and 30%, respectively, whereas the rest 10% was used for prediction purpose. Batch size in training process was also same, i.e., 10 but now number of iterations to repeat the training process was reduced from 2000 to 1000.

As the same procedure was repeated with different dataset sizes and iterations in training process, we have measured the same attributes for comparison purpose as we have done before. To compare the results with the previously used model configurations and the dataset distribution for training, testing, and prediction, we are again comparing the maximum difference between the actual and the predicted flow values as shown in Fig. 7.9.

From Fig. 7.9, we can see that the minimum maximum difference value is 0, and the maximum value for maximum difference between the predicted and the actual value is 38. To see whether there is overall improvement in the prediction or not, we have calculated the average maximum difference in both the cases. For first phase (Fig. 7.8), the average difference value is approximately 15, whereas it is 11.5 in phase 2 (Fig. 7.9). So, we can say that in phase 2, the accuracy as compared to the model configurations in phase 1 has improved. In addition to maximum

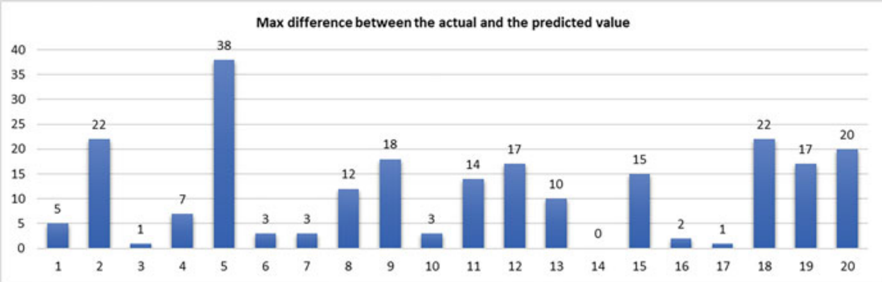

Fig. 7.9 Maximum difference between the actual and predicted vehicles flow values (phase 2)

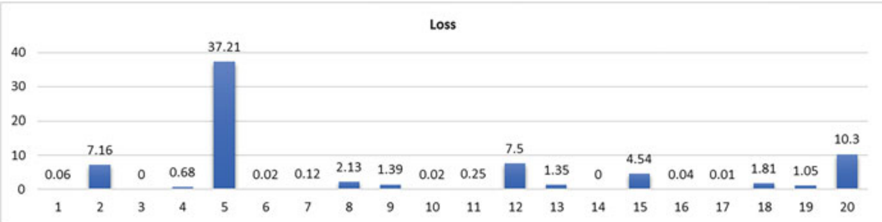

Fig. 7.10 Loss values when predicting vehicles flow in phase 2

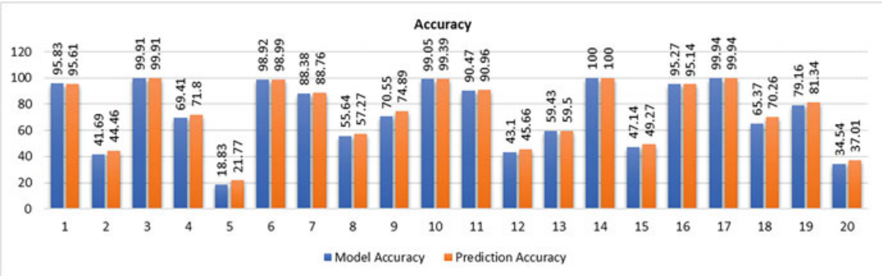

Fig. 7.11 Comparison of testing and prediction accuracy in phase 2

difference values, we have calculated system generated loss values which are shown in Fig. 7.10.

As we are using the different data for testing and prediction dataset, we have calculated the accuracy for both, testing and prediction processes for all 20 executions of our model in phase 2. Testing accuracy in this case has been generated by the system but the prediction accuracy has been calculated manually by comparing the actual and the predicted vehicles flow values. This shows that our model produced accurate results for both, testing and prediction data subsets. Comparison of testing and prediction accuracy values in phase 2 is shown in Fig. 7.11. In this figure, model accuracy represents the accuracy values obtained during the testing process using testing data subset.

7.7 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 behavior in disaster. The results show very high accuracy rate because of the high correlation between the input data and the output values. The results may differ when same deep learning model is applied on a different type of data. We have plotted MAE and MAPE results for all 20 executions of our model with the same specification. The results show 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 prove 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 [12, 13, 22, 35].

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 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 affected areas and can also use big data technologies to deal with real-time data.

Acknowledgements 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. 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)
2. Mehmood, R., Alam, F., Albogami, N.N., Katib, I., Albeshri, A., Altowaijri, S.M.: UTiLearn: a personalised ubiquitous teaching and learning system for smart societies. IEEE Access **5**, 2615–2635 (2017)

3. Muhammed, T., Mehmood, R., Albeshri, A., Katib, I.: UbeHealth: a personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities. IEEE Access **6**, 32258–32285 (2018)
4. Mehmood, R., Bhaduri, B., Katib, I., Chlamtac, I. (eds.): Smart Societies, Infrastructure, Technologies and Applications, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST). Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 224. Springer, Cham (2018)
5. 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. Tutorials **19**(4), 2456–2501 (2017)
6. 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, Piscataway (2011)
7. Mehmood, R., Faisal, M.A., Altowaijri, S.: Future networked healthcare systems: a review and case study. In: Boucadair, M., Jacquenet, C. (eds.): Handbook of Research on Redesigning the Future of Internet Architectures, pp. 531–558. IGI Global, Hershey (2015)
8. Usman, S., Mehmood, R., Katib, I.: Big data and HPC convergence: the cutting edge and outlook. In: International Conference on Smart Cities, Infrastructure, Technologies and Applications (SCITA 2017); Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, vol. 224, pp. 11–26. Springer, Cham (Nov 2018)
9. BBC: Japan Flood: At Least 179 Dead After Worst Weather in Decades (2018)
10. CNN: Japan Floods: Death Toll Rises to 195 as Abe Visits Affected Areas (2018)
11. Nikkei Asian Review: Japan Faces $2bn Price Tag for Flood Rebuilding. https://asia.nikkei.com/Politics/Japan-faces-2bn-price-tag-for-flood-rebuilding (2018). Accessed 20 Nov 2018
12. 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, pp. 1–10. ACM, New York (2014)
13. 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 (ITST), pp. 361–368. IEEE, Piscataway (2011)
14. Büscher, M., Coulton, P., Efstratiou, C., Gellersen, H., Hemment, D., Mehmood, R., Sangiorgi, D.: Intelligent mobility systems: some socio-technical challenges and opportunities. In: International Conference on Communications Infrastructure. Systems and Applications in Europe, pp. 140–152. Springer, Berlin (2009)
15. Mehmood, R., Meriton, R., Graham, G., Hennelly, P., Kumar, M.: Exploring the influence of big data on city transport operations: a Markovian approach. Int. J. Oper. Prod. Manag. **37**(1), 75–104 (2017)
16. Suma, S., Mehmood, R., Albugami, N., Katib, I., Albeshri, A.: Enabling next generation logistics and planning for smarter societies. Proc. Comput. Sci. **109**, 1122–1127 (2017)
17. Suma, S., Mehmood, R., Albeshri, A.: Automatic event detection in smart cities using big data analytics. In: International Conference on Smart Cities, Infrastructure, Technologies and Applications, pp. 111–122. Springer, Berlin (2017)
18. Alomari, E., Mehmood, R.: Analysis of tweets in Arabic language for detection of road traffic conditions. In: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, vol. 224, pp. 98–110. Springer, Cham (Nov 2018)
19. Mehmood, R., Graham, G.: Big data logistics: a health-care transport capacity sharing model. In: Procedia Computer Science, vol. 64, pp. 1107–1114. Elsevier, Amsterdam (2015)
20. Arfat, Y., Mehmood, R., Albeshri, A.: Parallel shortest path graph computations of United States road network data on apache spark. In: International Conference on Smart Cities, Infrastructure, Technologies and Applications, pp. 323–336. Springer, Berlin (2017)
21. Mehmood, R., Lu, J.A.: Computational Markovian analysis of large systems. J. Manuf. Technol. Manag. **22**(6), 804–817 (2011)

22. Aqib, M., Mehmood, R., Albeshri, A., Alzahrani, A.: Disaster management in smart cities by forecasting traffic plan using deep learning and GPUs. In: Mehmood, R., Bhaduri, B., Katib, I., Chlamtac, I. (eds.): International Conference on Smart Cities, Infrastructure, Technologies and Applications (SCITA 2017): Smart Societies, Infrastructure, Technologies and Applications, pp. 139–154. Springer, Cham (2018)
23. Mehmood, R., Nekovee, M.: Vehicular Ad hoc and grid networks: discussion, design and evaluation. In: Proceedings of The 14th World Congress On Intelligent Transport Systems (ITS), Beijing, October 2007. ITS America, Washington (2007)
24. Gillani, S., Shahzad, F., Qayyum, A., Mehmood, R.: A survey on security in vehicular Ad hoc networks. In: International Workshop on Communication Technologies for Vehicles, pp. 59–74. Springer, Berlin (2013)
25. Alvi, A., Greaves, D., Mehmood, R.: Intra-vehicular verification and control: a two-pronged approach. In: 2010 7th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP 2010), pp. 401–405. IEEE, Piscataway (2010)
26. Nabi, Z., Alvi, A., Mehmood, R.: Towards standardization of in-car sensors. In: International Workshop on Communication Technologies for Vehicles, pp. 216–223. Springer, Berlin (2011)
27. Ayres, G., Mehmood, R.: On discovering road traffic information using virtual reality simulations. In: 11th International Conference on Computer Modelling and Simulation, 2009. UKSIM'09, pp. 411–416. IEEE, Piscataway (2009)
28. Mehmood, R.: Towards understanding intercity traffic interdependencies. In: Proceedings of The 14th World Congress On Intelligent Transport Systems (ITS), Beijing, October 2007. ITS America, Washington (2007)
29. Graham, G., Mehmood, R., Coles, E.: Exploring future cityscapes through urban logistics prototyping: a technical viewpoint. Supply Chain Manag. An Int. J. **20**(3), 341–352 (2015)
30. 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)
31. 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: Intelligent Transportation Systems–Problems and Perspectives, pp. 3–35. Springer, Berlin (2016)
32. Schlingensiepen, J., Mehmood, R., Nemtanu, F.C., Niculescu, M.: Increasing sustainability of road transport in European cities and metropolitan areas by facilitating autonomic road transport systems (ARTS). In: Wellnitz, J., Subic, A., Trufin, R. (eds.): Sustainable Automotive Technologies 2013, pp. 201–210. Springer, Ingolstadt (2014)
33. Schlingensiepen, J., Mehmood, R., Nemtanu, F.C.: Framework for an autonomic transport system in smart cities. Cybern. Inf. Technol. **15**(5), 50–62 (2015)
34. Alam, F., Mehmood, R., Katib, I.: D2TFRS: an object recognition method for autonomous vehicles based on RGB and spatial values of pixels. In: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, vol. 224, pp. 155–168. Springer, Cham (Nov 2018)
35. Alazawi, Z., Abdljabar, M.B., Altowaijri, S., Vegni, A.M., Mehmood, R.: ICDMS: an intelligent cloud based disaster management system for vehicular networks. In: International Workshop on Communication Technologies for Vehicles, pp. 40–56. Springer, Berlin (2012)
36. Alazawi, Z., Alani, O., Abdljabar, M.B., Mehmood, R.: An intelligent disaster management system based evacuation strategies. In: 2014 9th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP), pp. 673–678. IEEE, Piscataway (2014)
37. Alazawi, Z., Alani, O., Abdljabar, M.B., Mehmood, R.: Transportation evacuation strategies based on VANET disaster management system. Procedia Econ. Financ. **18**, 352–360 (Jan 2014)
38. 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)
39. 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)

40. Tsirogiannis, D., Harizopoulos, S., Shah, M.A.: Analyzing the energy efficiency of a database server. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 231–242. ACM, New York (2010)
41. Valentini, G.L., Lassonde, W., Khan, S.U., Min-Allah, N., Madani, S.A., Li, J., Zhang, L., Wang, L., Ghani, N., Kolodziej, J., et al.: An overview of energy efficiency techniques in cluster computing systems. Clust. Comput. **16**(1), 3–15 (2013)
42. Chen, J., Chiang, R.C., Huang, H.H., Venkataramani, G.: Energy-aware writes to non-volatile main memory. In: Proceedings of the 4th Workshop on Power-Aware Computing and Systems, p. 6. ACM, New York (2011)
43. Dhiman, G., Ayoub, R., Rosing, T.: PDRAM: a hybrid pram and dram main memory system. In: DAC'09. 46th ACM/IEEE Design Automation Conference, 2009, pp. 664–669. IEEE, Piscataway (2009)
44. Seshadri, V., Kim, Y., Fallin, C., Lee, D., Ausavarungnirun, R., Pekhimenko, G., Luo, Y., Mutlu, O., Gibbons, P.B., Kozuch, M.A., et al.: RowClone: fast and energy-efficient in-dram bulk data copy and initialization. In: Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 185–197. ACM, New York (2013)
45. 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)
46. Ciresan, D.C., Meier, U., Masci, J., Maria Gambardella, L., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: IJCAI Proceedings-International Joint Conference on Artificial Intelligence, vol. 22, p. 1237, Barcelona, Spain. AAAI Press, Palo Alto (2011)
47. Lighthill, M.J., Whitham, G.B.: On kinematic waves II. A theory of traffic flow on long crowded roads. Proc. R. Soc. Lond. A **229**(1178), 317–345 (1955)
48. Richards, P.I.: Shock waves on the highway. Oper. Res. **4**(1), 42–51 (1956)
49. Greenshields, B., Channing, W., Miller, H., et al.: A study of traffic capacity. In: Highway Research Board Proceedings, vol. 1935. National Research Council (USA), Highway Research Board (1935)
50. Yang, Q., Koutsopoulos, H.N.: A microscopic traffic simulator for evaluation of dynamic traffic management systems. Trans. Res. C Emerg. Technol. **4**(3), 113–130 (1996)
51. Ahmed, K.I.: Modeling Drivers' Acceleration and Lane Changing Behavior. PhD thesis, Massachusetts Institute of Technology, Massachusetts (1999)
52. Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent development and applications of SUMO—Simulation of Urban MObility. Int. J. Adv. Syst. Meas. **5**(3&4), 128–138 (December 2012)
53. Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J.: LSTM network: a deep learning approach for short-term traffic forecast. IET Intell. Transp. Syst. **11**(2), 68–75 (2017)
54. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: a generic approach for extreme condition traffic forecasting. In: Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 777–785. SIAM, Philadelphia (2017)
55. Jia, Y., Wu, J., Du, Y.: Traffic speed prediction using deep learning method. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 1217–1222. IEEE, Piscataway (2016)
56. 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, Piscataway (2013)
57. Dandala, T.T., Krishnamurthy, V., Alwan, R.: Internet of vehicles (IoV) for traffic management. In: International Conference on Computer, Communication and Signal Processing (ICCCSP), 2017, pp. 1–4. IEEE, Piscataway (2017)
58. 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: International Conference on Emerging Technological Trends (ICETT), pp. 1–7. IEEE, Piscataway (2016)

59. 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**(6), 1576–1585 (2016)
60. 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)
61. Berkeley, U.: Caltrans (2005) freeway performance measurement system (PeMS) 5.4. pems.eecs.berkeley.edu/Public, Accessed 30 June 2005
62. Polson, N.G., Sokolov, V.O.: Deep learning for short-term traffic flow prediction. Transp. Res. C Emerg. Technol. **79**, 1–17 (2017)
63. Polson, N., Sokolov, V.: Deep learning predictors for traffic flows. arXiv preprint arXiv:1604.04527 (2016)
64. Ma, X., Yu, H., Wang, Y., Wang, Y.: Large-scale transportation network congestion evolution prediction using deep learning theory. PloS One **10**(3), e0119044 (2015)
65. Chollet, F., et al.: Keras (2015). https://keras.io
66. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from https://www.tensorflow.org/