EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

AIoT Enabled Traffic Congestion Control System Using Deep Neural Network

Shahan Yamin Siddiqui[1,2], Inzmam Ahmad[1], Muhammad Adnan Khan[3,*], Bilal Shoaib Khan[1], Muhammad Nadeem Ali[4], Iftikhar Naseer[5], Kausar Parveen[6], Hafiz Muhammad Usama[1]

[1]School of Computer Science, Minhaj University, Lahore, Pakistan.
[2]School of Computer Science, National College of Business Administration & Economics, Lahore, Pakistan.
[3]Riphah School of Computing and Innovation, Faculty of Computing, Riphah International University, Lahore Campus, Lahore, Pakistan.
[4]Department of Computer Science, Lahore Garrison University, Lahore, Pakistan.
[5]Department of Computer Science and Information Technology, Superior University, Lahore, Pakistan.
[6]Department of Computer Science, University of Engineering and Technology, Lahore 54890, Pakistan.

Abstract

With rapid population growth in cities, to allow full use of modern technology, transportation networks need to be developed efficiently and sustainability. A significant problem in the traffic motion barrier is dynamic traffic flow. To manage traffic congestion problems, this paper provides a method for forecasting traffic congestion with the aid of a Deep neural network that minimizes blockage and plays a vital role in traffic smoothing. In the proposed model, data is collected and received by using smart Internet of things enabled devices. With the help of this model, data of the previous junction of signals will send to another junction and update after that next layer named as intelligence prediction for the congestion layer will receive data from sensors and the cloud which is used to find out the congestion point. The proposed TC2S- DNN model achieved the accuracy of 98.03 percent and miss rate of 1.97 percent which is better then previous published approaches.

Keywords: Deep neural network (DNN), Traffic congestion control system, AIoT, Smart city, Machine Learning .

Received on 20 April 2021, accepted on 21 September 2021, published on 28 September 2021

doi: 10.4108/eai.28-9-2021.171170

1. Introduction

A smart system is an innovative concept of using modern technological advents to design a smarter and better lifestyle. The intelligent structures include neural networks and networking technologies to improve the use of public resources, improve quality, and reduce the operating costs of services to make them accessible for the residents. Smart city vision provides people easiness, integrated, and economic access to public services. This idea has brought many advantages to plan automatic public services as well as availability, reliability, and openness to local government services, such as safety and surveillance, public property, and the protection of cultural patrimony, residue management, advanced traffic monitoring system, etc. The demand for safe cities brings together main industries such as intelligent governance, smart construction, smart utilities, intelligent environment, highly intelligent transport, and so on [1]. As we have known intelligent transport is a part of the smart city project and traffic jamming is one of the major problems with the public transport system in recent times. Traffic congestion impacts human activities negatively because it slows down a country's productivity, development, and overall growth. As a result, over the past few decades, several researchers have focused on the issue of traffic congestion in the public transport system [2].

To solve traffic jamming problems several countries are working also to maintain their current transport systems to enhance flexibility, security, traffic movements, and reduce

* Corresponding author. Email: Mnadeemali@lgu.edu.pk

car use demand. According to the U.S. statistical investigation of 2007 by improving public transit, road control services, upgrading traffic signals, and handling collisions [2]. The transportation department was found to have attributed half of the congestion generated to repetitive congestion. These traffic complications are frequently congestion problems owing to the poor direction of the travelers. The other half of the congestion is due to the non-recurring congestion caused by incidents of traffic, areas of work, or on special occasions. Non-recurring incidents affectedly reduce the available capacity and efficiency of the overall transport system [2].

In Colorado, the Denver metropolitan council of governments frequently tracks road congestion and reports on road bottlenecks and also establishes a congestion reduction strategy [3] and the Beijing expressway network based on speed level data collected from the Beijing traffic management bureau (BTMB) [1]. It is now known that often traffic pollution is attributed to inadequate signal control in metropolitan areas. As the signals include all congested roads and non-congested roads with a static timing. Congested highways also take more time to need traffic hazards. The cars are queued on the road as time rises. Sometimes this large traffic queue also affects the other connecting roads which are directly linked to the congested road. Such traffic queues often impact emergency services, such as mobile cops, fire service, an ambulance, etc. [4]

Nowadays this topic is commonly debated in smart city planning and earlier studies suggest that traffic congestion relies not just on road construction but also road maintenance. Building underpasses or overhead bridges to widen the path doesn't fix the overarching issue of traffic congestion. Often, conventional approaches like static traffic signals and traffic police are not enough to manage congestion as combined with the increase in road traffic. The studies further indicate that the network still requires careful maintenance for traffic. Past study summaries show that other factors such as speed, density, and distance also play an important role in traffic control. [5].

The smart city focuses on these factors and provides a solution to the base of these factors. Over the past several designs and models have been developed to solve the conditions of pollution. Such massive sensors and methods are being planted [3] to adjust the traditional traffic signals network for the dynamic or self-learning framework, but sensing the congestion and monitoring the junction signal at one stage will not solve the congestion issue entirely. If one junction does not have the previous one's congestion status, so the junctions are unconnected. For the seamless movement of traffic, the roadside signals have to be coordinated with each other. And the traffic flow will be regulated at all junctions to prevent congestion. Accurate and on-the-spot traffic flow data is critical for the efficient implementation of smart transport systems. Traffic data has risen in recent years and we are now in the era of large data travel [6].

Existing approaches to traffic flow prediction using primarily superficial traffic prediction models and are still unsatisfactory for many real-world implementations. This situation urges us to re-examine the issue of modeling traffic flow focused on broad traffic information and deep development models. The Deep neural networks (DNN) is a multi-layered and latest algorithm which has more accuracy rates as compared to old algorithms [7, 8].

The proposed Artificial intelligent internet of things enabled Traffic congestion control system uses the deep neural network (TC2S-DNN) system model used to analyze the congestion of traffic in the environment of a smart city. The proposed model comprises a machine learning-enabled IoT-based road traffic congestion control system in which congestion occurrence is notified on a specific point. The model is proposed for a smart traffic congestion control system to collect data from different sensor devices and then analyze it to predict the congestion of traffic. A deep neural network is used for the prediction of congestion for traffic control systems.

2. Literature Review

Several studies have been conducted for congestion control in urban areas and different literature uses different mythology to provide a solution. This paper focuses on the substantial advantages the smart city market has not elevated yet because of political, technical, and financial hurdles and barriers [9].

Ata suggests a traffic congestion prediction based on the Artificial intelligence (AI) system. The suggested algorithm is called Modelling smart road traffic congestion regulation utilizing neural network artificial backpropagation (MSR2C-ABPNN). The program shows the message on the vehicle LCD after forecasting the congestion, and also offers the alternate route by utilizing the Google map [1]. Tamimi et al. presented the artificial neural network (ANNs) approach to estimate traffic delay with fuzzy logic. The proposed model was based on time, distance, traffic flow, wind speed, temperature, and humidity variables. The simulation results revealed the feasibility of the suggested method and confirmed that expert functions for the set of data provided to learning by a well-trained network [7].

Li proposed to forecast traffic incidents based on deep machine learning. Data obtained from California on traffic incidents using eight metrics such as descriptions, incident frequency, upstream station speed (km / h), upstream occupancy (percent), upstream strength (vehicle per lane and hour), downstream station speed (km / h), downstream occupancy (percent) and downstream occupancy (vehicle per lane and hour). The Extreme learning machine (ELM) was implemented and compared with other algorithms such as Naïve Bayes (NB) and Support vector machine (SVM) [10]. Ma proposed a transport congestion prediction system using a deep learning method. A Global position system (GPS) is used to collect data on transport network congestion. GPS tracks the location, time, and pace of the connections. This reported data is then supplied to the neural network as feedback, and this network generates the congestion forecast. The prediction is based on RNN-RBM neural network algorithms [11].

Siddiqui suggested low-cost vehicle speed measurement methodology by categorizing acoustic wave patterns, reported with a sole acoustic sensor on the roadside. This uses engine noise, tire exhaust noise, and air friction as vehicle vibration, Doppler change whose frequencies are used to measure vehicle speed. Nevertheless, the prevalence of extremely loud transportation and a wide range of acoustic vehicle signs will restrict the applicability of such approaches in emerging regions [12].

Sakran studied an infrastructure that combines IoT with agent technology into a common network where agent technology manages efficient communication and interfaces within IoT between a large number of heterogeneous, dynamically distributed, and autonomous devices. The networks included the usage of active RFID, wireless sensor technologies, ad-hoc networking of objects, and Internet-based information systems through which tagged traffic artifacts can be dynamically sensed, monitored, and queried across a network. The analysis presented a summary of a distributed traffic simulation model for the IoT traffic management system using Net Logo, an agent-based environment, mobile agent technologies [13].

Mishu worked on real-time observing and guiding road traffic using IoT. Cloud for IoT is used where different facilities are provided such as storage server and application. RF transmitter is used for emergency traffic control and also used load cells to compute the time required for the flawless volume of traffic. When vehicles passing on the load cells beneath the road it converts the loading action into electrical signals [14].

Li examined the approach based on the A* algorithm that was developed to identify the shortest paths for any time interval on a road network with speed trends. For example, for an individual who wants to go to work sometime between 7:00 and 7:45 AM, it can be the shortest path. Every day belongs to precisely one division of their proposed method; workday or non-workday and a path line have the same duration at the same time of the day for the days in the same group. We believed, though, that speed is partially constant on each street line [15].

Ding studied a time-dependent shortest path problematic with an edge-delay feature travel time function connected with each edge of a graph or road network. To find the least total travel time, they proposed a new Dijkstra-based algorithm [16]. Akinboro proposed a smart road traffic monitoring network that involves weighted cameras, global positioning systems, VSAT, control rooms, and handheld computers. To get the details about the car, the weighted sensors are mounted on the roadside. This information is then forwarded to the control room. The control room monitors the traffic figures and shows the traffic condition on cell phones on the Google chart. This device sends a text alert message regarding the congested road and proposes an alternate route for road users. Sadhukhan offers an IoT-based device composed of two units: A Traffic density management module (TDMM) and a Traffic management module (TMM). The device monitors vehicle numbers and regulates the signals according to the traffic level. The TDMM is a sensor that is mounted on the roadside to calculate traffic intensity and this knowledge is then transmitted to TMM to adjust traffic signals according to the density measured at the road crossing. Thatsanavipas et al., the comparison, image processing techniques were used to determine traffic density and to control traffic properly. Image processing is used for edge detection [17].

The Peripheral interface controller (PIC) microcontroller is used to calculate the number of vehicles in real-time mode. The registered information is transmitted to the central station by the microcontroller. The controller accesses traffic situations on any available transportation lights and also adjacent roads to reduce traffic jamming. Malhi et al. also experiment carried out in the use of an Infrared (IR) sensor in prioritizing emergency vehicles where microcontrollers used to obstruct through sending a red gesture to all sides of the road but one with an emergency vehicle. Work also performed using the IR sensor utilizing fuzzy logic to identify the position of emergency vehicles as well. Evolutionary computation is playing a vital role in multimodal optimization [18] for multisolution problems [19]. By using internet of things, smoothnes of traffic can be increased in efficient manners [20, 21].

The major contribution of the proposed TC2S- DNN model helps to increase the efficiency of traffic management system.

3. Proposed TC2S-DNN System Model

The researcher proposed a new Cloud and IoT-based intelligent prediction for Traffic congestion control system using a deep neural network (TC2S- DNN) for an intelligent traffic system. The whole process of the proposed TC2S- DNN system is shown in Fig. 1. In the proposed CI2P-TC-DLNN model, data were obtained through smart Internet of things (IoT) enabled devices. The proposed model is categorized into two phases training phase and the validation phase. There are four layers in the training phase which are sensor layer, object layer, pre-processing layer, and application layer. The sensor layer collects data and sends data to the object layer the data is raw so it may contain noisy and missing values. Then the pre-processing layer receives data from the object layer, it mitigates the noisy values and predicts missing values using different functions. The application layer is further divided into two sub-layers prediction layer and the performance evaluation layer. The prediction layer predicts traffic congestion using DNN. Then the performance layer evaluates the performance of the system using different parameters such as miss rate, accuracy, etc. When the application layer is completed then the system is decided based on the output of the application layer. If required learning criteria did not meet then retrain the system and again evaluate the performance of the system and soon. Once it achieves the required learning rate then the trained model is stored on the cloud.

After that, the second part of the model validation phase is activated which contains two layers named as sensor layer and IPTC. The sensor layer collect data with the help of IoT devices. After that, the IPTC layer receives and combines data from the sensor layer and cloud. IPTC layer predicts that congestion is found or not at the point. If congestion is not found then the IPTC layer again collects data from the sensor layer and again predicts the output with the help of import trained model from the cloud. Again check congestion found or not if congestion found then select an alternate path with the help of Google maps. These congestion points will forecasts by our proposed TC2S- DNN system model with the help of DEML. Condition of the road also become worse if adverse weather condition occurs that will directly affect the traffic congestion too. Different factors like rain, humidity, temperature, etc. may be affected by adverse weather.

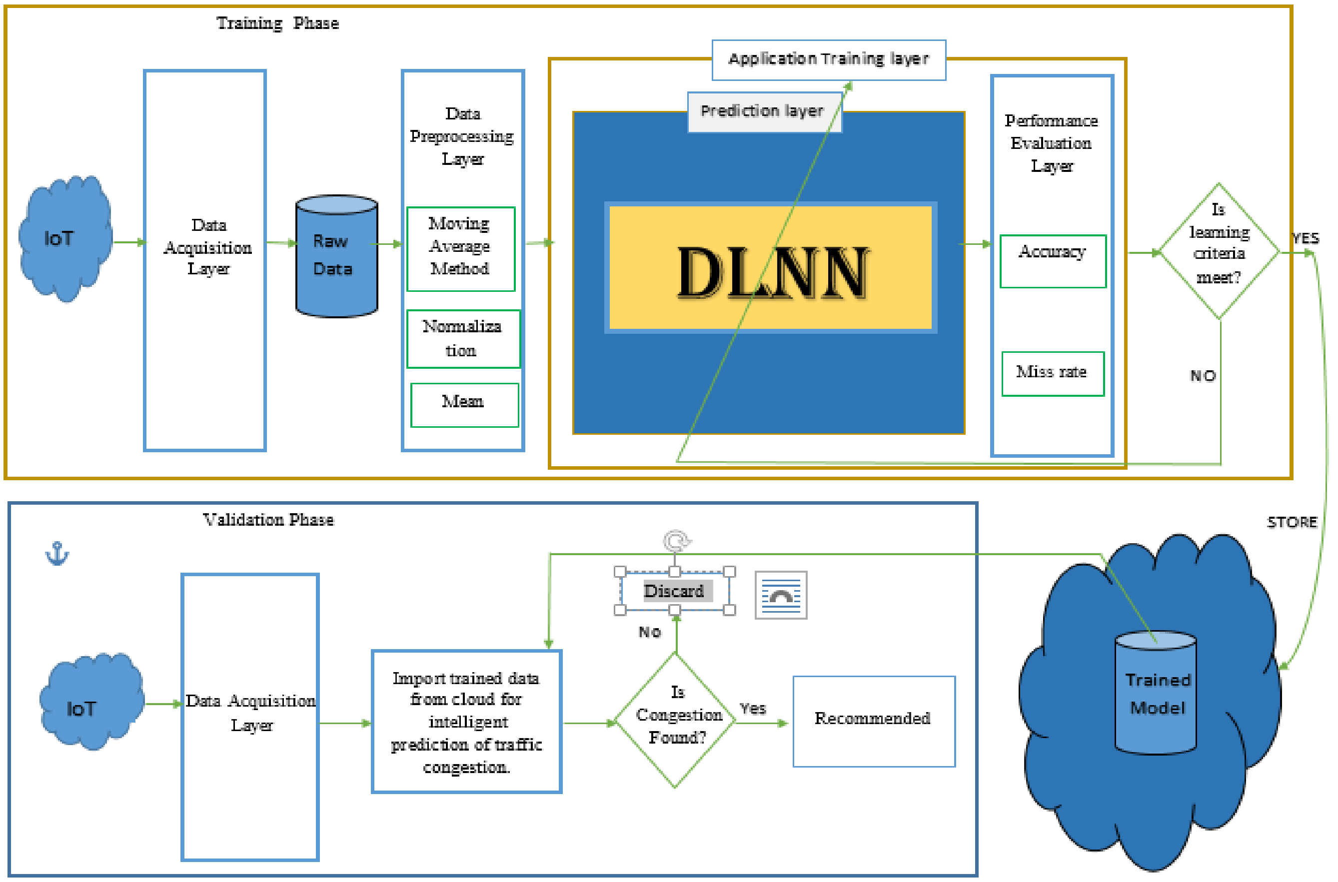

Figure 1. Proposed TC2S- DNN system model

Another work on the traffic congestion control system by using the same data set as uses in the proposed model. The data set contains nine attributes and 1786 instances [1]. The dataset has eight input attributes and one output attribute. The input attributes are time, speed, flow, headway, air temp, relative humidity, average wind speed, and freeze temperature and the output attribute is occupancy.

The input layer, hidden layer, and output layer are used in the back-propagation process of the artificial neural network. There are different steps in the backpropagation process, such as weight initialization, feedforward, backpropagation of error, and weight and bias updating. Every neuron in the hidden layer has the activation function f(x) = Sigmoid(x). The sigmoid function for first layer input is written as in Eq. (1), and the first layer output is written as in Eq. (3), and the suggested TC2S- DNN hidden layer of the sigmoid function is written as in Eq. (3) activation feature for the output layer as seen in Eq. (4).

3.1 Mathematical Model

Eq. (1) shows the feed-forward propagation for the input layer and the output of the first layer is written in Eq. (2).

$$\chi_j = b_1 + \sum_{i=1}^{m} (\psi_{ij} * o_i) \tag{1}$$

Where b_1=bias, ψ_{ij}= weight, $o_i = input$

$$\upsilon_j = \frac{1}{1+e^{-\chi_j}} \; where\ j = 1,2,3,\dots,o \tag{2}$$

Where υ_j=output of hidden neurons, χ_j= net of hidden neurons

Eq. (3) shows the feed-forward propagation for the hidden layers to the output layer.

$$\chi_{k^l} = b^l + \sum_{j=1}^{n}(z_{jk^{l=1}} * \upsilon_j^{l=1}) \quad (3)$$

Where χ_{k^l} = net of output neurons, $\upsilon_j^{l=1}$ is represents the weights between hidden and output neurons, $z_{jk^{l=1}}$ is represent the out values of hidden neurons

Eq. (4) indicates the activation feature for the output layer.

$$\upsilon_{k^l} = \frac{1}{1+e^{-\chi_k^{l-1}}} \ where\ k = 1,2,3,\dots,o \quad (4)$$

$$\chi_{k^l} = b^l + \sum_{j=1}^{n}(z_{jk^l} * \upsilon_j^l) \ where\ l = 1,2,,\dots,6 \quad (5)$$

Error in backpropagation is written as in Eq. (6)

$$Err = \frac{1}{2}\sum_k(Target_k - \upsilon_k^{l=6})^2 \quad (6)$$

Where $\boldsymbol{Target_k}$ shows the required output.

$$\Delta\boldsymbol{\psi} \propto -\frac{\partial \boldsymbol{Err}}{\partial \boldsymbol{\psi}}$$

$$\Delta\boldsymbol{\upsilon}_{\boldsymbol{j,k^{l=6}}} = -\boldsymbol{\epsilon}\frac{\partial \boldsymbol{Err}}{\partial \boldsymbol{\upsilon}_{\boldsymbol{j,k}}^{\boldsymbol{l=6}}} \quad (7)$$

In Eq. (8), applying the chain rule

$$\Delta\boldsymbol{\upsilon}_{\boldsymbol{j,k^{l=6}}} = -\boldsymbol{\epsilon}\frac{\partial \boldsymbol{Err}}{\partial \boldsymbol{\upsilon_k}^{\boldsymbol{l}}} \times \frac{\partial \boldsymbol{\upsilon_k}^{\boldsymbol{l}}}{\partial \boldsymbol{\chi_{k^l}}} \times \frac{\partial \boldsymbol{\chi_{k^l}}}{\partial \boldsymbol{\upsilon_{j,k^l}}} \quad (8)$$

After applying the chain rule, the above equation can be written as:

$$\Delta\boldsymbol{\upsilon}_{\boldsymbol{j,k^{l=6}}} = \boldsymbol{\epsilon}(\boldsymbol{Target_k} - \boldsymbol{\upsilon_k^l}) \times \boldsymbol{\upsilon_{k^l}}(\boldsymbol{1} - \boldsymbol{\upsilon_{k^l}}) \times (\boldsymbol{\upsilon_j^l})$$

$$\Delta\boldsymbol{\upsilon}_{\boldsymbol{j,k^l}} = \boldsymbol{\epsilon}\,\boldsymbol{\xi_{k^l}}\boldsymbol{\upsilon_j^l}$$

Where

$$\xi_{k^l} = (Target_k - \upsilon_{k^l}) \times \upsilon_{k^l}(1 - \upsilon_{k^l})$$

$$\Delta\psi_{i,j^l} \propto -\left[\sum_k \frac{\partial Err}{\partial \upsilon_{k^l}} \times \frac{\partial \upsilon_k^{\ l}}{\partial \chi_{k^l}} \times \frac{\partial \chi_{k^l}}{\partial \upsilon_{j^l}}\right] \times \frac{\partial \upsilon_{j^l}}{\partial \chi_{j^l}} \times \frac{\partial \chi_{j^l}}{\partial \psi_{i,j^l}}$$

$$\Delta\psi_{i,j^l} = -\epsilon\left[\sum_k \frac{\partial Err}{\partial \upsilon_{k^l}} \times \frac{\partial \upsilon_{k^l}}{\partial \chi_{k^l}} \times \frac{\partial \chi_{k^l}}{\partial \upsilon_{j^l}}\right] \times \frac{\partial \upsilon_{j^l}}{\partial \chi_{j^l}} \times \frac{\partial \chi_{j^l}}{\partial \psi_{i,j^l}}$$

$$\Delta\psi_{i,j^l} = \epsilon\left[\sum_k (Target_k - \upsilon_k^{\ l}) \times \upsilon_{k^l}(1 - \upsilon_{k^l}) \times (\upsilon_{j^l})\right] \times \upsilon_{k^l}(1 - \upsilon_{k^l}) \times q_i$$

$$\Delta\psi_{i,j^l} = \epsilon\left[\sum_k \xi_{k^l}(z_{j,k^l})\right] \times \upsilon_{j^l}(1 - \upsilon_{j^l}) \times q_i$$

$$\Delta\psi_{i,j^l} = \epsilon\xi_{j^l}\,q_i$$

Where,

$$\xi_{j^l} = \left[\sum_k \xi_{k^l}(z_{j,k^l})\right] \times \upsilon_{j^l}(1 - \upsilon_{j^l})$$

Eq. (10) is used for updating the weights between the hidden and output layer.

$$\boldsymbol{z}_{\boldsymbol{j,k^{l=6}}}^{+} = \boldsymbol{z}_{\boldsymbol{j,k^{l=6}}} + \boldsymbol{\lambda}_{\boldsymbol{c^{l=6}}}\Delta\boldsymbol{\upsilon}_{\boldsymbol{j,k^{l=6}}} \quad (10)$$

Eq. (11) is used for updating the weights between the remaining hidden layers

$$\boldsymbol{\psi}_{\boldsymbol{i,j^l}}^{+} = \boldsymbol{\psi}_{\boldsymbol{i,j^l}} + \boldsymbol{\lambda}_{\boldsymbol{c^l}}\Delta\boldsymbol{\psi}_{\boldsymbol{i,j^l}} \quad (11)$$

$\boldsymbol{\lambda_c}$ are the learning rate of the CI2P-TC-DLNN and the value $\boldsymbol{\lambda_c}$ is between 0 and 1. The convergence of CI2P-TC-DLNN depends upon the careful selection of the value$\boldsymbol{\lambda_c}$.

4. Result and Discussions

In this article, the MATLAB 2019a tool is used and the dataset was adopted from the internet that displays M1 junction 37 England's weather forecast and traffic speed at 10 minutes intervals. To forecast the network's behavior, dynamically distribute resources [7], the experiments show that this approach allows for better network usage. DNN was used to train and fit 1786 instances of data. The data set is divided into two parts one is 80% for training (1430 samples), second is 20% (356 samples) for validation. To measure the performance of the proposed TC2S- DNN model, various statistical parameters like miss rate, accuracy, sensitivity, specificity, etc. are used to predict COVID-19.

$$Misrate = \frac{O1/C0 + O0/C1}{C0 + C1} * 100\%$$

$$Accuracy = \frac{O0/C0 + O1/C1}{C0 + C1} * 100\%$$

$$Sensitivity = \frac{O0/C0}{O1/C0 + O1/C1} * 100\%$$

$$Specificity = \frac{O1/C1}{O1/C0 + O1/C1} * 100\%$$

$$FPV = [100 - Specificity]\%$$

$$FNR = [100 - Sensitivity]\%$$

$$NPV = \frac{O1/C1}{C1} * 100\%$$

$$PPV = \frac{O0/C0}{C0} * 100\%$$

Table 1. Training accuracy of proposed TC2S- DNN model

N=1430	Performance of DNN o/p(O)

(80%) i/p(C)	O0 Congestion not found	O1 Congestion found
C0=900 No congestion found	896	4
C1=530 Congestion found	11	519

Table 2. Validation of the accuracy of the proposed TC2S- DNN model

N=356 (20%) i/p(C)	Performance of DLNN o/p(O)	
	O0 Congestion not found	O1 Congestion found
C0=224 No congestion found	222	2
C1=132 Congestion found	5	127

Table 3. Training and validation accuracy of the proposed TC2S- DNN model

	Accuracy	Miss Rate
Training	98.95%	1.05%
Validation	98.03%	1.97%

DNN is used for the traffic data set, and the simulation findings are shown in Tab. 1 and Tab. 2. The testing performance of the proposed TC2S- DNN method with different hidden layers during traffic congestion forecasting in preparation. The dataset is divided into two parts, 80% (1430 samples) for training and 20% (356 samples) for validation purposes showed in Tab. 1. C0 indicates that no congestion occurs and C1 indicates that congestion occurs. In Tab. 1, it seems that the number of 1430 samples with 900 and 530 for no congestion and congestion respectively. In no congestion found case, the proposed system correctly predicted 896 and 4 wrongly predicted. In the congestion case, the proposed system model predicted 519 correctly and 11 wrongly.

In the validation phase, a total number of 356 samples were used for the validation of the proposed model. A total number of 222 for no congestion found case and 132 for congestion case. In no congestion found case, the proposed system correctly predicted 222 and 2 wrongly predicted. In the congestion case, the proposed system model predicted 127 correctly and 5 wrongly.

Table 4. Training and validation of the statistical factors of proposed TC2S- DNN model

	Sensitivity	Specificity	FPV
Training	98.79%	99.24%	0.76%
Validation	97.79%	98.45%	1.55%
	FNR	**NPV**	**PPV**
Training	1.21%	97.92%	99.55%
Validation	2.20%	96.21%	99.12%

Table 5. Comparison with previous work with the proposed TC2S- DNN model

Comparison	Training		Validation	
	Accuracy	Miss Rate	Accuracy	Miss Rate
Tamimi and Zahoor (2010) [7]	78.12%	21.88 %	76.1%	23.9 %
Pushpi and Dilip Kumar (2018) [8]	91.265%	8.732 %	90.6%	9.4%
Atta et al. (2019) [1] Using fitting modelling	96.20%	3.80 %	95.84%	4.16 %
Atta et al. (2019) [1] Using time-series	98.15%	1.85 %	97.56%	2.44 %
Proposed AIoT-TCC-DEML model	98.95%	1.05 %	98.03%	1.97 %

Tab. 3 indicates the accuracy of the proposed TC2S-DNN model during testing and validation. Accuracy and error rates are estimated to be 98.95 percent and 1.05 percent respectively. The accuracy and error rate for validation was 98.03 % and 1.97 % respectively. As in Tab. 4 training results also reveal that the sensitivity, specificity, False positive value (FPV), False negative rate (FNR), Negative predicted value (NPV), and Positive predicted value (PPV), are 98.79 %, 99.24 %, 0.76 %, 1.21 %, 97.92 %, and 99.55 % respectively. Validation results indicate that the sensitivity, specificity, FPV, FNR, NPV, PPV are 97.797 %, 98.45 %, 1.55 %, 2.20 %, 96.21 %, and 99.12 % respectively.

A comparison with various existing studies like Tamimi & Zahoor(2010) [7], Pushpi & Dilip Kumar(2018)

[8], and Atta et al. (2019) [1] were shown in Tab. 5, that represented the proposed TC2S- DNN obtained the accuracy rate and the miss rate are 98.95 %, 1.05 % respectively as well as during the validation process the accuracy rate and the miss rate are 98.03 %, 1.97 %, respectively.

5. Conclusion

As a computational solution, this study has used a deep neural network to control traffic flow congestion. The smart transportation system is found to have a negligible effect on the smart city, while the traditional systems do not have the potential to actively track the neighboring signal timers to reduce traffic congestion. Thus, an advanced traffic congestion management system is implemented to track the traffic signal timer continuously using the approach of a deep neural network. Different sensors mounted on different adjacent signals capturing, exchanging, and transmitting traffic data to a controller by using IoT-enabled devices. Time is critical to soft real-time applications such as smart traffic. Thus, if the information obtained in delay or too much noise by the signal sensors can be influenced by the output of the proposed solution. This paper proposed a new version of TC2S- DNN for a smart traffic system for control the data collection. The findings of the simulation revealed that the performance of the suggested system yield better results compared to the previous approaches.

References

[1] A. Ata, M. A. Khan, S. Abbas, G. Ahmad, and A. Fatima, “Modelling smart road traffic congestion control system using machine learning techniques,” Neural Network World, vol. 29, no. 2, pp. 99-110, 2019.

[2] P. K. Mittal and Y. Singh, “Analysis and designing of proposed intelligent road traffic congestion control system with image mosaicking technique,” International Journal of Information and Technology, vol. 2, pp. 27-31, 2013.

[3] A. M. Rao and K. R. Rao, “Measuring urban traffic congestion-a review,” International Journal for Traffic and Transport Engineering, vol. 2, no. 4, pp. 14-26, 2012.

[4] K. Iqbal, M. A. Khan, S. Abbas, Z. Hasan and A. Fatima, "Intelligent Transportation System (ITS) for Smart-Cities using Mamdani Fuzzy Inference System," International Journal of Advanced Computer Science and Applications, vol. 9, no. 2, 2018.

[5] A. Ata, M. A. Khan, S. Abbas, M. S. Khan and G. Ahmad, "Adaptive IoT Empowered Smart Road Traffic Congestion Control System Using Supervised Machine Learning Algorithm," The Computer Journal, pp. 1-18, 2020.

[6] R. Pushpi and K. Shaw, “Artificial neural networks approach induced by fuzzy logic for traffic,” Journal of Engineering Technology, vol. 20, no. 4, pp. 1-15, 2018.

[7] S. Tamimi and M. Zahoor, “Link delay estimation using fuzzy logic,” The 2nd IEEE International Conference on Computer and Automation Engineering, vol. 9, no. 2, pp. 135-140, 2010.

[8] R. Pushpi and K. Shaw, “Artificial neural networks approach induced by fuzzy logic for traffic,” Journal of Engineering Technology, vol. 20, no. 4, pp. 1-15, 2018.

[9] X. Ma, H. Yu, Y. Wang and Y. Wang, Large-scale transportation network congestion evolution prediction using deep learning theory,” PloS One, vol. 10, no. 3, pp. 11-24, 2015.

[10] S. Y. Siddiqui, M. A. Khan, S. Abbas and F. Khan, “Smart occupancy detection for road traffic parking using deep extreme learning machine,” Journal of King Saud University-Computer and Information Sciences, vol. 57, no. 1, pp. 1-7, 2020.

[11] D. Hussain, M. A. Khan, S. Abbas, R. A. Naqvi, M. F. Mushtaq, A. Rehman and A. Nadeem, "Enabling Smart Cities with Cognition Based Intelligent Route Decision in Vehicles Empowered with Deep Extreme Learning Machine," Cmc-Computers Materials & Continua, vol. 66, no. 1, pp. 141-156, 2021.

[12] L. Li, S. Wang and X. Zhou, “Fastest path query answering using time-dependent hop-labeling in the road network,” IEEE Transactions on Knowledge and Data Engineering, vol. 2, no. 3, pp. 10-18, 2020.

[13] S. Abbas, M. A. Khan, A. Athar, S. A. Shan, A. Saeed, and T. Alyas, "Enabling Smart City With Intelligent Congestion Control Using Hops With a Hybrid Computational Approach," The Computer Journal, pp. 1-15, 2020.

[14] P. Sadhukhan and F. Gazi, “An iot based intelligent traffic congestion control system for road crossings,” IEEE International Conference on Communication, Computing and Internet of Things, pp. 371-375, 2018.

[15] K. Thatsanavipas, N. Ponganunchoke, S. Mitatha and C. Vongchumyen, “Wireless traffic light controller,” Procedia Engineering, vol. 24, no. 3, pp. 190-194, 2011.

[16] R. Weil, J. Wootton and A. Garcia-Ortiz, “Traffic incident detection: sensors and algorithms,” Mathematical and Computer Modelling, vol. 27, no. 9, pp. 257-291, 2019.

[17] M. H. Malhi, M. H. Aslam, F. Saeed, O. Javed and M. Fraz, “Vision-based intelligent traffic management system,” IEEE Frontiers of Information Technology, vol. 12, no. 2, pp. 137-141, 2011.

[18] Zhang, Y. H., Gong, Y. J., Gao, Y., Wang, H., & Zhang, J. (2019). Parameter-free Voronoi neighborhood for evolutionary multimodal optimization. IEEE Transactions on Evolutionary Computation, 24(2), 335-349.

[19] Huang, T., Gong, Y. J., Kwong, S., Wang, H., & Zhang, J. (2019). A niching memetic algorithm for multi-solution traveling salesman problem. IEEE Transactions on Evolutionary Computation, 24(3), 508-522.

[20] Liu, W. L., Gong, Y. J., Chen, W. N., Liu, Z., Wang, H., & Zhang, J. (2019). Coordinated charging scheduling of electric vehicles: A mixed-variable differential evolution approach. IEEE Transactions on Intelligent Transportation Systems, 21(12), 5094-5109.

[21] Wang, H., Zhang, Z., & Taleb, T. (2018). Special issue on security and privacy of IoT. World Wide Web, 21(1), 1-6.