

IoT Based Traffic Prediction and Traffic Signal Control System for Smart City

Neelakandan S (✉ snkresearch07@gmail.com)

Jeppiaar Institute of Technology <https://orcid.org/0000-0001-8583-0019>

Berlin M A

RMD Engineering College

Sandesh Tripathi

Birla Institute of Applied Sciences

Brindha Devi V

Sri Sairam Institute of Technology

Indu Bhardwaj

Galgotia University: Galgotias University

Arulkumar N

Christ University

Research Article

Keywords: Optimized Weight Elman Neural Network (OWENN), Improved Beetle Swarm Optimization (IBSO), Intel 80286 Microprocessor, Internet of Things (IoT), and Smart City

Posted Date: April 19th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-306500/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

IoT BASED TRAFFIC PREDICTION AND TRAFFIC SIGNAL CONTROL SYSTEM FOR SMART CITY

S.Neelakandan¹, Dr.M.A.Berlin², Dr.Sandesh Tripathi³,Dr.V.Brindha Devi⁴,
Indu Bhardwaj⁵,Arulkumar N⁶

¹Assistant Professor, Department of IT,Jeppiaar Institute of Technology, Chennai,
snkresearch07@gmail.com

²Professor, Department of CSE, R.M.D Engineering College, berlinstanly@gmail.com

³Assistant Professor, Birla Institute of Applied Sciences,sandeshtripathi@gmail.com

⁴Associate Professor, Department of Information Technology,

Sri Sai Ram Institute of Technology,Chennai,brindha.renju@gmail.com

⁵Assistant Professor, Galgotias University, indubhardwaj2011@gmail.com

⁶Assistant Professor, Department of CS,

CHRIST (Deemed to be University), Bangalore,itsprofarul@gmail.com

Abstract: Because of the population increasing so high, and traffic density remaining the same, traffic prediction has become a great challenge today. Creating a higher degree of communication in automobiles results in the time wastage, fuel wastage, environmental damage, and even death caused by citizens being trapped in the middle of traffic. Only a few researchers work in traffic congestion prediction and control systems, but it may provide less accuracy. So, this paper proposed an efficient IoT based traffic prediction using OWENN algorithm and traffic signal control system using Intel 80286 microprocessor for a smart city. The proposed system consists of '5' phases, namely, IoT data collection, feature extraction, classification, optimized traffic IoT values, and traffic signal control system. Initially, the IoT traffic data is collected from the dataset. After that, traffic, weather, and direction information are extracted, and these extracted features are given as input to the OWENN classifier, which classifies which place has more traffic. Suppose one direction of the place has more traffic, it optimizes the IoT values by using IBSO, and finally, the traffic is controlled by using Intel 80286 microprocessor. The experimental results show that the proposed system outperforms state-of-the-art methods.

Key words: *Optimized Weight Elman Neural Network (OWENN), Improved Beetle Swarm Optimization (IBSO), Intel 80286 Microprocessor, Internet of Things (IoT), and Smart City.*

1. INTRODUCTION

The current high level of increase in the number of vehicles without additional supporting transportation infrastructure is a major problem for the smart city growth. The endless amount of car, which is very crowded in urban areas, increase air pollution and sound pollution and other causes, which reduce the fuel efficiency, encourage the intersection traffic, and raise the pace of traffic by major changes in the speed control framework in urban areas [1]. Owing to the pollution and traffic disruptions generated by these mechanisms (signal control), traffic management and logistics are often major problems that must be dealt with. Previously, the traffic signals controls device looked to be utilised as a traffic management system, but now that it is being used as a smart city's traffic management system, it plays an important part in traffic safety [2]. As one of the main inputs for understanding and managing traffic, traffic data collections are quite important. Traffic counting is a common occurrence nowadays, since a traditional state or city government has some way to maintain track of traffic, such as speed guns, radar weapons, microwave sensors, and cameras [3,18]. But, IoT-based data collection is the best way to collect traffic information in the smart city [4] because it collects real-time traffic information. On the other hand, traffic congestion detection (i.e. traffic management) is one of the problems in the efficient traffic signal control system. In certain areas, traffic congestion in main roads has been a big issue with the shortage and restricted provision of public transport. If there is no initiative, this dilemma would be unmanageable [5, 6].

Various alternatives to the traffic jam crisis in the smart cities have been suggested. In order to reduce congestion, certain strategies propose the best individual routes [7], which may exacerbate congestion in other regions [8]. Others have proposed some classification algorithms, say, Fuzzy Neural Network (FNN), Random forest, C4.5, K-Nearest Neighbour (KNN), etc [9]. But, the existing algorithm doesn't overcome the current problems in smart city traffic. If once the system efficiently classifies the congestion or traffic, it could be easily controlled by the system [10]. This paper proposed an efficient approach to classify the traffic and control the traffic efficiently by optimizing the waiting time using a microprocessor for a smart city.

The rest of the paper is organised as follows: Section 2 addresses related works in the traffic forecast and traffic management framework for the smart city. In Section 3, the traffic prediction and traffic control technique is defined. Section 4 deals with findings and ultimately, Section 5 ends work and possible directions for studies planned.

2. RELATED WORK

Haipeng Yao et al.[11] have proposed a traffic classification mechanism for the networking Internet of Things (IoT) capsule for cities with integrated extraction, practical collection and classification model functionality. The technology also removed the manual selection method for traffic and extended in particular to smart city scenarios. In the sense of traffic classification, the capsule networks were included. Experimental findings showed the system's viability and efficacy which resulted in high ranking accuracy, although the system's measurement costs and training time were high.

Mengting Bai *et al.* [12] implemented a forecast of traffic congestion using a relative congestive tensor in smart cities. First, a congestion matrix on regional traffic networks focused on the notion of relative places for the road nodes has been created. The machine then used a long-term memory network approach for forecasting congestion on all places of the road network. Experiments have shown that the device has greatly outperformed baseline models at any point where congestion sometimes exists. The spatial matrix designed therefore had such sparseness, and where there were a few road nodes in the regional road network, the usage of space was inadequate. The road network must then be fairly separated.

MdAshifuddinMondal and ZeenatRehena [13] presented smart classification method of congestion traffic dependent on artificial neural network presented (ANN). The traffic congestion condition designation was performed using the ANN-based method. The ITS framework automatically updates the traffic laws, as adjusts the queue long in traffic signal signals, recommending alternative paths, depending on traffic congestion status. The consequence was that, for various types of road segments of variable lane width, the machine specifically defined the congested status provided the model is thoroughly trained with appropriate traffic data sets. However, the machine has an overhead storage issue and the system's training period was longer.

HyunjinJoo *et al.* [14] recommended reinforcement learning centred traffic signal management for smart cities. The device primarily maximises the amount of cars that reach a crossroads and balances road signals with Q-learning (QL). The configuration of the framework was adaptive and could be changed to suit improvements in the initial intersection structure. In comparison to other QL-based systems, the framework has done well as shorter queue duration and waiting time with a normal deviation in queue length. However, a traffic signal may not be operated more accurately by the machine.

Albert Rego et al. [15] introduced a framework for the effective handling of emergencies in smart towns, with a software-determined network based control systems. This design has been developed upon a series of IoT networks consisting of illumination, traffic cameras and an algorithm. The algorithm tracked resource demand and route shift to promote the deployment of emergency services vehicles[17]. Experiments also shown that when the Algorithm applied and the device could be scaled the emergency traffic delay increased by 33 percent. But, the system is not suitable for another kind of smart city real-world scenarios. The data collected using this system are uploaded to the cloud using a mobile application and then transferred to the field-programmable gate array (FPGA) analysis system. The raw data are computed and processed by the FPGA system, and pathological conditions are displayed on the patient's wearable IoT device[16].

3. PROPOSED METHODOLOGY

Today, traffic management has developed into one of the metropolitan city's main issues. The steady rise in vehicle numbers contributes to a repetitive traffic control crisis. The introduction and applicability of the Internet of Things in Smart Cities provides a great forum to cope with traffic challenges and thus build intelligent traffic management systems (ITMS). This paper proposed an efficient OWENN algorithm for traffic prediction and traffic signal control using a Intel 80286 microprocessor for a smart city. Figure 1 presents the proposed traffic prediction and traffic signal control system. This model mainly consists of three phases, namely, IoT data collection, traffic prediction, feature selection, and traffic signal control. Each phase is briefly explained in the below subsections.

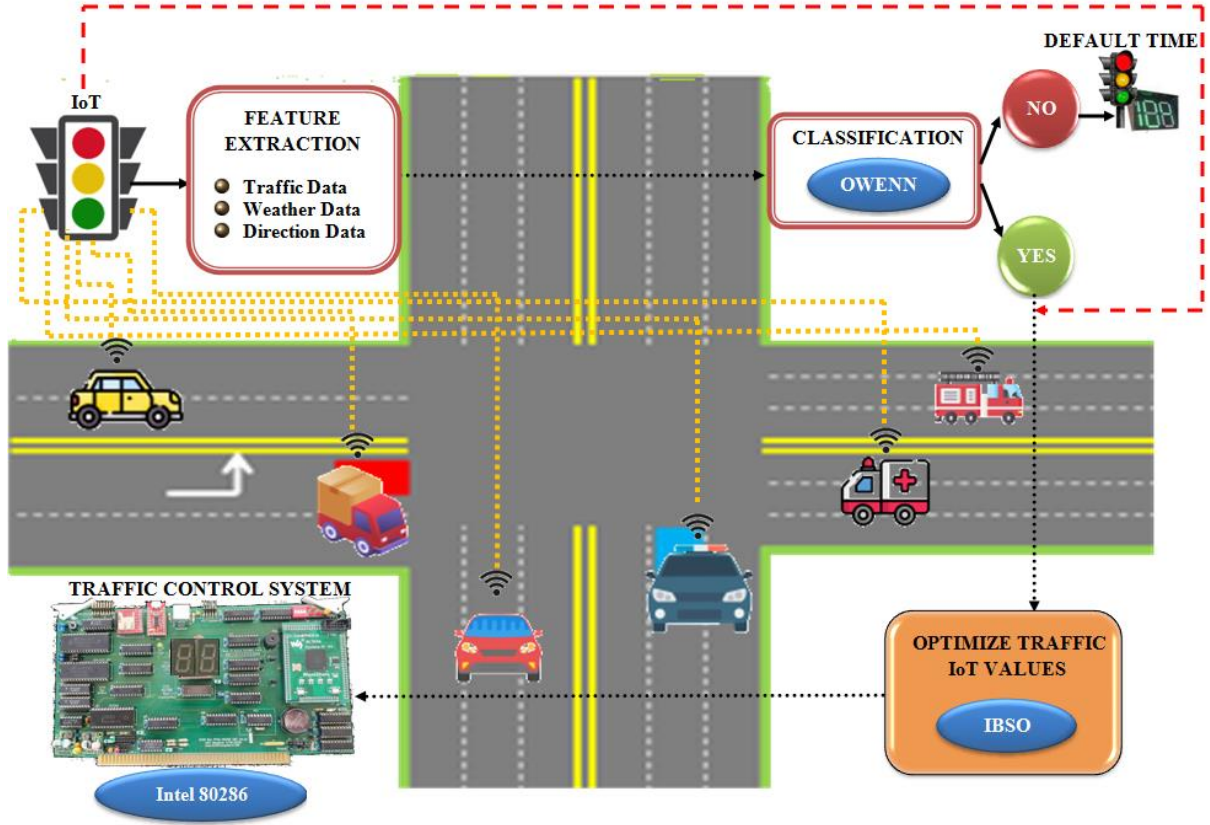


Figure 1: The system model

3.1 IoT Data Collection

The initial process will be to collect the data from the public repository. The traffic IoT data, such as traffic-related information, weather-related information, and direction-related information are extracted and get saved in the database for further process. The dataset is expressed as follows,

$$\bar{A}_{ID}'' = \{\bar{a}_1'', \bar{a}_2'', \bar{a}_3'', \dots, \bar{a}_n''\} \quad (1)$$

Where, \bar{A}_{ID}'' represents the traffic IoT dataset for further processing and \bar{a}_n'' indicates the n -number of traffic related information in the dataset.

3.2 Feature Extraction

In this phase, the feature extraction process is carried out. Here, the system extracts traffic information, weather information, and direction information features. These features are explained briefly in the below section.

3.2.1 Traffic information

Traffic information gives the basis for traffic signal control systems. It covers data, policies, rules, and temporary measures. The important traffic information is extracted from the database are speed, the total number of vehicles, travel time, position, and street lighting features. These characteristics are used to define the service quality, the congestion quantity and the steps to be implemented to mitigate circumstances. It is evaluated as follow:

$$\overline{B}_i'' = \{\overline{b}_1'', \overline{b}_2'', \overline{b}_3'', \dots, \overline{b}_n''\} \quad (2)$$

Where, \overline{B}_i'' indicates the extracted traffic information features and \overline{b}_n'' signifies the n - number of features.

3.2.2 Weather information

The weather information at this intersection was installed to warn drivers of conditions and reduce accidents. Here, the system extracts the weather information, such as snow, summer, and rainy reduce roadway capacity. Hither, the system extracts these common features from the dataset. Mathematically, it is expressed as follow:

$$\overline{C}_{wi}'' = \{\overline{c}_1'', \overline{c}_2'', \overline{c}_3'', \dots, \overline{c}_k''\} \quad (3)$$

Where, \overline{C}_{wi}'' represents the extracted weather information feature and \overline{c}_k'' indicates the k -number of feature.

3.2.3 Direction information

The direction-based information is used to find the traffic and control the traffic signal effectively. Here, the system extracts the direction-based information as north, east, west, and south. These features are described mathematically as,

$$\overline{D}_{di}'' = \{\overline{d}_1'', \overline{d}_2'', \overline{d}_3'', \dots, \overline{d}_m''\} \quad (4)$$

Where, \overline{D}_{di}'' denotes the extracted direction information feature and \overline{d}_m'' indicates the m -number of feature. These aforementioned features are extracted from the database; it is commonly denoted as \overline{E}_{fs}'' , which is used for further processing.

3.3 Classification

After extracting the features, the classification is performed in this step. Hereabout, the classification is done by using the Optimized weight Elman Neural Network (OWENN) algorithm that classifies which places have more traffic. Elman Neural Network (ENN) is an effective algorithm to classify objects. In a normal ENN, only one hidden layer is built, but, hither more number of hidden layers is utilized and each classification algorithm has the weight initialization problem, which also affects the system fastness and accuracy. So, here, a new weight initialization method is used, which will give a better result. So, the term is called an OWENN algorithm. The steps involved in the OWENN algorithm are explained below:

First, the extracted feature set $\bar{E}_{fs}'' = \{\bar{e}_1'', \bar{e}_2'', \bar{e}_3'', \dots, \bar{e}_k''\}$ is given as input to the OWENN algorithm. The weight is calculated for the input value and that is given to the hidden layer. The hidden layer is mathematically described as follows:

$$\bar{F}_{hl}''(l) = g\left(\kappa_1 \bar{H}_{cu}''(l) + \kappa_2 \bar{E}_{fs}''(l-1)\right) \quad (5)$$

Where, $\bar{F}_{hl}''(l)$ indicates the l^{th} output of the hidden unit, $\bar{H}_{cu}''(l)$ signifies the context unit that is described below,

$$\bar{H}_{cu}''(l) = \bar{F}_{hl}''(l-1) \quad (6)$$

Where, $g(.)$ indicates the transfer function of the hidden layer, κ_1 and κ_2 indicates the connection weight of the input layer to the hidden layer and the context unit to the hidden layer, thus, context unit is also useful for weight calculation but, hither the *Glorot* weight-based weight value is calculated, which is stated as follows,

$$\kappa_m \sim U_d \left[-\frac{1}{\sqrt{k}}, \frac{1}{\sqrt{k}} \right] \quad (7)$$

Where, κ_m signifies the weight value ($m=1,2,3,\dots,n$), U_d indicates the uniform distribution and k indicates the size of the previous layer. Thus, the hidden layer output is given as the input to the output layer. The output calculation is formulated as follows,

$$I(l) = \varphi(\kappa_3 \bar{F}_{hl}''(l)) \quad (8)$$

Where, $I(l)$ signifies the output layer, $\varphi(\cdot)$ signifies the transfer function of the output layer, and κ_3 represents the weight of the hidden layer to the output layer. The pseudo-code for the proposed algorithm is elucidated in the below figure,

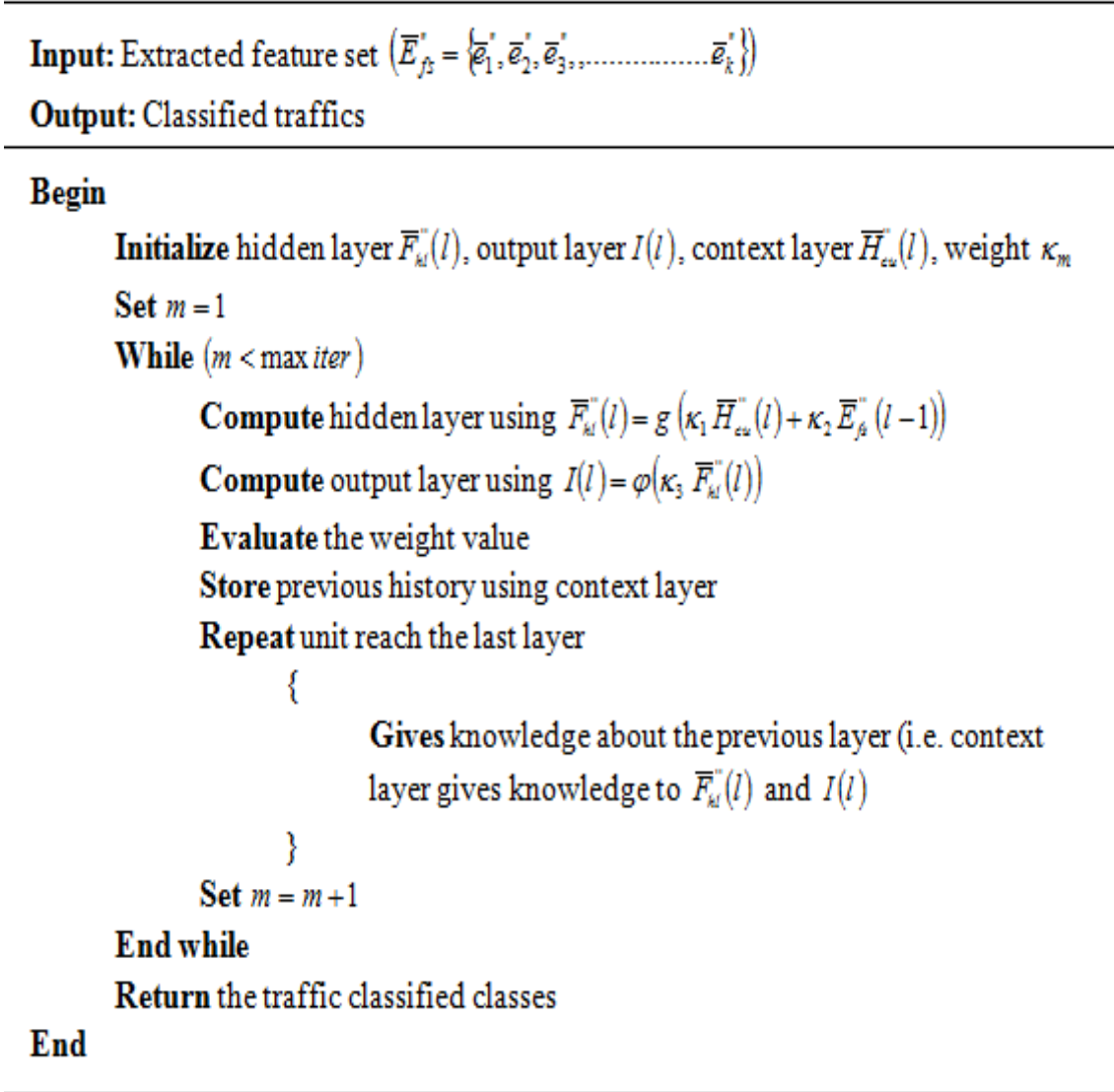


Figure 2: Pseudocode for the OWENN algorithm

It identifies which places have more traffic accurately. Once it finds one direction of the place that has more traffic, it optimizes the traffic IoT values, and it increases the waiting time of the vehicle by using the Intel 80286 microprocessor; otherwise, it sets the default time of the traffic signal.

3.4 Optimize Traffic IoT Values

In this section, the IoT traffic values (\bar{A}_{ID}'') are given as input to the Improved Beetle Swarm Optimization (IBSO) algorithm. The Beetle Swarm Algorithm (BSO) is a powerful algorithm that is suggested by improving the efficiency of swarm optimization by implementing the concepts of beetle forage. But, the conventional BSO algorithm has a convergence problem and less optimization accuracy. So, this paper proposed an IBSO algorithm that selects the high fitness value from the gathered IoT data. The procedures included in the IBSO algorithm are explained below:

Initialize the traffic IoT values as $\bar{A}_{ID}'' = \{\bar{a}_1'', \bar{a}_2'', \bar{a}_3'', \dots, \bar{a}_n''\}$. Then, the fitness value of each beetle position (i.e. vehicle) is calculated. The speed of the m^{th} beetle is described as $\bar{J}_{sb}'' = \{\bar{j}_1'', \bar{j}_2'', \bar{j}_3'', \dots, \bar{j}_m''\}$. The individual extremity of the beetle is described as $\bar{K}_{eb}'' = \{\bar{k}_1'', \bar{k}_2'', \bar{k}_3'', \dots, \bar{k}_m''\}$ and the group extreme value of the population is expressed as $\bar{L}_{ge}'' = \{\bar{l}_1'', \bar{l}_2'', \bar{l}_3'', \dots, \bar{l}_m''\}$. The mathematical model for simulating its behaviour is as follow:

$$\bar{M}_m^{z+1} = \bar{M}_m^z + \mu \bar{j}_m'' + (1 - \mu) \rho_m'' \quad (9)$$

Where, $m = 1, 2, \dots, n$, z indicates the current number of iterations, \bar{j}_m'' indicates the speed of the beetles, μ denotes the positive constants, and ρ_m'' describes the increase in beetle position movement. Thus, the speed formula is written as follows:

$$(\bar{J}_{sb}'')^{z+1} = \phi(\bar{J}_{sb}'')^z + \alpha_1 \beta_1 ((\bar{K}_{eb}'')^z - (\bar{M}_m'')^z) + \alpha_2 \beta_2 ((\bar{L}_{ge}'')^z - (\bar{A}_{ID}'')^z) \quad (10)$$

$$\phi(\bar{J}_{sb}'')^z \sim \eta = z^{-1-\bar{J}_{sb}''}, \quad 0 \leq \bar{J}_{sb}'' \leq 2 \quad (11)$$

Where, α_1 and α_2 denotes two positive constants, β_1 and β_2 indicates two random functions in the range $[0, 1]$, ϕ signifies the levy flight to select optimal weight, η indicates the standard normal distribution, and z indicates the current number of iterations. After that, update the incremental function by using,

$$\rho_m'' = \chi^z * (\bar{J}_{sb}'')^z * (f(N_{lm}^z) - (N_{rm}^z)) \quad (12)$$

$$N_{lm}^{z+1} = N_{lm}^z + (\bar{J}_{sb}'') * d/2 \quad (13)$$

$$N_{m}^{z+1} = N_{m}^z - (\bar{J}_{sb}^n) * d / 2 \quad (14)$$

Where, N_{lm}^{z+1} and N_{rm}^{z+1} indicates the search behaviour of position of the vehicle (i.e. direction of the vehicle) respectively, d signifies the distance between the vehicles and $f(.)$ signifies the fitness value. High fitness value is taken as optimal values, which are used for further processing.

3.5 Traffic Signal Control System

After optimizing the traffic IoT values, traffic signals are controlled by using the *Intel 80286* Microprocessor in this phase. It controls the traffic signal automatically (i.e. increase the waiting time of the vehicle) based on the IoT values. The Intel 80286 is an advanced microprocessor, and it is a 16-bit microprocessor. It provides special operations to allow effective operating system deployment and execution. For example a command may stop the output of a task, save its status, turn to another task, load its state, and start the performance of the new task. It also helps virtual memory structures by offering an exception and restartable guidance for segments not current. It consists of '4' sectional blocks, namely, address unit, bus unit, instruction unit, and an execution unit. The working principles of these blocks are separately explained below:

Address Unit: Initially, The address unit is computed for the physical address where the data or directions are to be fetched (i.e. optimised Traffic IoT values). The determined address shall then be passed to the bus unit after measurement of the physical address.

Bus Unit: The data was collected from the memory via the data bus. The bus unit can fetch the instructions from the memory in advance and save them in the queue for the smoother implementation of the instruction. It then sends them on the command unit.

Instruction Unit: The instruction unit now decodes the instructions. When instructions are extracted from a queue, the decoder constantly pays attention to the instructions and stores the instructions into a properly decoded list. In this instruction unit, the proposed system sets the waiting time of the vehicle based on the IoT values. For example, for 100 IoT values, it sets one waiting time, for 200 IoT values, it sets another one waiting time, etc.

Execution Unit: The instructions from the decoded instruction queue are fed to the execution unit. Here, suppose, one direction of the place has more traffic, then it increases the waiting

time based on the optimized traffic IoT values to provide sophisticated control and coordination to confirm that traffic moves as smoothly and safely as possible.

4. RESULTS AND DISCUSSION

In this section, the performance of the proposed IoT based traffic prediction and traffic signal control system for a smart city is analyzed. The proposed system collects the data from the Kaggle dataset, which is the publically available dataset. The system is implemented in the working platform of JAVA. The performance of the proposed system is validated through the performance analysis section.

4.1 Performance Analysis

In this performance analysis, the performance of the proposed OWENN algorithm is compared with some traditional techniques, namely, Elman Neural Network (ENN), Convolutional Neural Network (CNN), Neural Network (NN), and Adaptive Neuro-Fuzzy Inference System (ANFIS) techniques. The scheme utilises four estimation metrics used for classification problems to assess the efficiency of the proposed procedure, i.e. precision, f-measurement, mean absolute error (MAE) and Root mean square error (RMSE). These analyses are shown in the figure below,

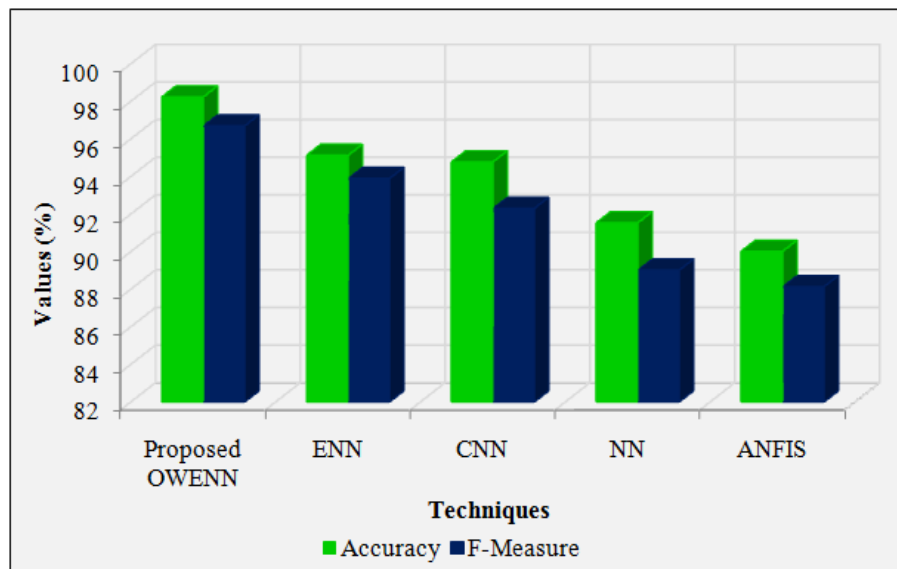
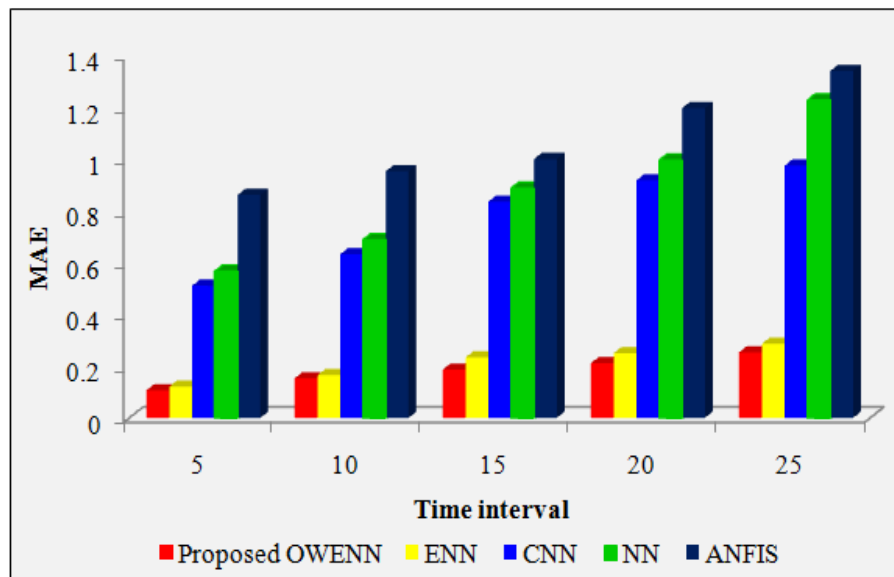


Figure 3: Accuracy and F-measure analysis

Discussion: The above figure indicates the performance of the proposed OWENN algorithm with conventional ENN, CNN, NN, and ANFIS methods in terms of accuracy and f-measure

metrics. Concerning the accuracy metric, the proposed one attains 98.23 % accuracy, but the existent ENN, CNN, NN, and ANFIS proffer accuracy of 95.12 %, 94.77 %, 91.53 %, and 90.01 % respectively, which is very lower when compared to the proposed one. Concerning the f-measure metric, the existing ANFIS proffers lower performance than the proposed one. Also, the existing ENN, CNN, and NN offer low-level performance than the proposed one. But, the proposed OWENN proffer f-measure of 96.69 %, which is higher than all the existing methods. Thus, the discussion concludes that the proposed one attains better results.



(a)

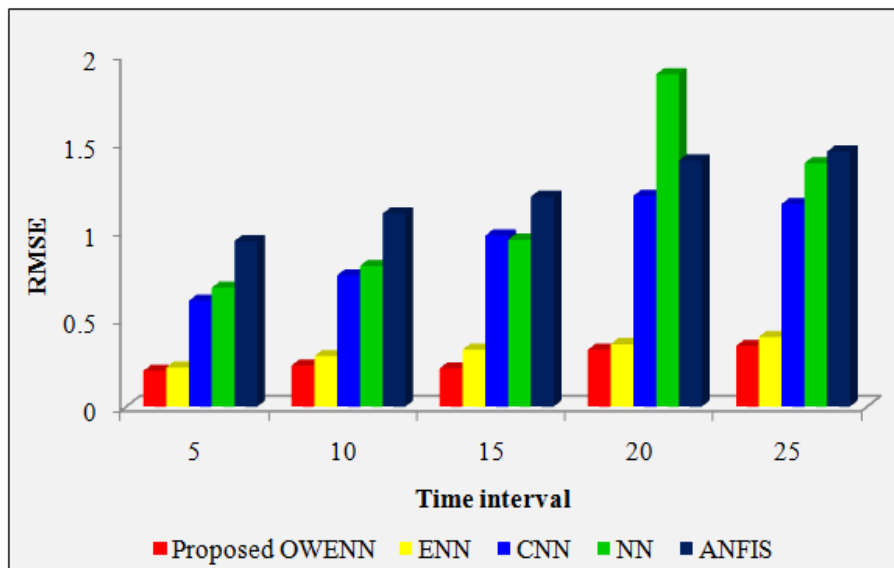


Figure 4: (a) MAE and (b) RMSE analysis with different time intervals

Discussion: The above figure evinces the performance of the proposed OWENN algorithm with the prevailing techniques, say, ENN, CNN, NN, and ANFIS techniques in terms of (a) MAE and (b) RMSE performance metrics. The MAE and RMSE are two important metrics to measure accuracy for continuous variables. The performance is centered on the number of time intervals (5min-25 min). From figure 4 (a), for 5 min interval, the proposed one has an MAE value of just 0.107, which is lower than the conventional ENN, CNN, NN, and ANFIS methods. For, the remaining time interval also, the proposed one attains better performance than the existing methods. From figure 4 (b), the existing ANFIS proffers high RMSE. Also, the existing ENN, CNN, and NN proffer high RMSE, but the proposed one offer RMSE of 5min-25min time interval are 0.203, 0.234, 0.218, 0.324, and 0.345 respectively, which is very lower than all the existing methods. Overall, the discussion shows that the performance of the proposed system gradually decreases while the time interval increases, and also, the proposed one attain better performance than the conventional methods.

5. CONCLUSIONS

There are also drawbacks to handle existing traffic efficiently in the conventional framework. This paper introduced an effective OWENN algorithm to effectively forecast traffic and a proposal to enhance and improve performance in Intel 80286 Microprocessor traffic management. The proposed system comprises of '5' phases, namely, IoT data collection, feature extraction, classification, optimized traffic IoT values, and traffic signal control system. The performance of the proposed system is compared with the existent ENN, CNN, NN, and ANFIS methods in terms of accuracy, f-measure, MAE, and RMSE metrics. Here, the proposed one attains 98.23 % accuracy and 96.69 % measure. And also, the proposed one offers an average MAE and RMSE of 0.254 and 0.345 respectively, which is lower when contrasted to the existing techniques. Overall, the test findings revealed the high degree of efficiency of the suggested relative to traditional approaches. The proposed system uses WiFi to communicate among users, but its energy usage and recharging solutions are considered in future work.

Author contributions

S.Neelakandan ,M.A Berlin contributed to technical and conceptual content and architectural design and Sandesh Tripathi,V.Brindha Devi contributed to guidance and counselling on the writing of the paper. Indu Bhardwaj and Arulkumar N contributed for Testing and validation of outcomes.

Compliance with ethical standards

Conflict of interest

The authors declare that they have no conflict of interest.

REFERENCES

1. Neetesh Kumar, Syed Shameerur Rahman, and NavinDhakad, “Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system”, IEEE Transactions on Intelligent Transportation Systems, 2020,10.1109/TITS.2020.2984033.
2. JunchenJin, ,Xiaoliang Ma, and IisakkiKosonen, “An intelligent control system for traffic lights with simulation-based evaluation”, Control engineering practice, vol. 58, pp. 24-33, 2017, 10.1016/j.conengprac.2016.09.009.
3. Guy M.Lingani, Danda B. Rawat, and Moses Garuba, “Smart traffic management system using deep learning for smart city applications”, In IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0101-0106, 2019,10.1109/CCWC.2019.8666539
4. Mohammed Saleh Ali Muthanna, Yuliy T. Lyachek, Abdulfattah Mohammed ObadiMusaeed, Yaqoob Ahmed HazzaaEsmail, and Abuzar BM Adam, “Smart system of a real-time pedestrian detection for smart city”, In IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EConRus), pp. 45-50, 2020. 10.1109/EConRus49466.2020.9039333.
5. Willy Carlos Tchuitcheu, Christophe Bobda, and MdJubaer Hossain Pantho, “Internet of Smart-Cameras for Traffic Lights Optimization in Smart Cities”, Internet of Things, pp. 100207, 2020, 10.1016/j.iot.2020.100207.
6. VishwajeetPattanaik, Mayank Singh, P. K. Gupta, and S. K. Singh, “Smart real-time traffic congestion estimation and clustering technique for urban vehicular roads”, In IEEE region 10 conference (TENCON), pp. 3420-3423, 2016, 10.1109/TENCON.2016.7848689.

7. Ronan Doolan and Gabriel-Miro Muntean, "EcoTrec—A novel VANET-based approach to reducing vehicle emissions", *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 608-620, 2016.
8. Neelakandan, S & Paulraj, D 2020, 'An Automated Exploring And Learning Model For Data Prediction Using Balanced CA-Svm', *Journal of Ambient Intelligence and Humanized Computing*, Springer, pp. 1-12, April,2020. ISSN 1868-5137
9. Shilin Zhao, Kejiang Ye, and Cheng-Zhong Xu, "Traffic classification and application identification based on machine learning in large-scale supercomputing center", In *IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 2299-2304, 2019, 10.1109/HPCC/SmartCity/DSS.2019.00319.
10. Arjun Dutta Abhisikta Chakraborty, Amit Kumar, Anirup Roy, Swarnabha Roy, Debarpita Chakraborty, HimadriNathSaha et al, "Intelligent traffic control system: towards smart city", In *IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1124-1129, 2019, 10.1109/IEMCON.2019.8936188
11. Haipeng Yao, Pengcheng Gao, Jingjing Wang, Peiying Zhang, Chunxiao Jiang, and Zhu Han, "Capsule network assisted IoT traffic classification mechanism for smart cities", *IEEE Internet of Things Journal*, vol. 6, no. 5, pp.7515-7525, 2019.
12. Mengting Bai, Yangxin Lin, Meng Ma, Ping Wang, and LihuaDuan, "PrePCT: Traffic congestion prediction in smart cities with relative position congestion tensor", *Neurocomputing*, 2020, 10.1016/j.neucom.2020.08.075.
13. AshifuddinMondalMd, and ZeenatRehena, "Intelligent traffic congestion classification system using artificial neural network", In *Companion Proceedings of The World Wide Web Conference*, pp. 110-116. 2019,10.1145/3308560.3317053.
14. HyunjinJoo, Syed Hassan Ahmed, and Yujin Lim, "Traffic signal control for smart cities using reinforcement learning", *Computer Communications*, 2020, 10.1016/j.comcom.2020.03.005.

15. Albert Rego, Laura Garcia, Sandra Sendra, and Jaime Lloret, "Software defined Network-based control system for an efficient traffic management for emergency situations in smart cities", *Future Generation Computer Systems*, vol. 88, pp. 243-253, 2018.
16. Satpathy, S., Mohan, P., Das, S. et al. A new healthcare diagnosis system using an IoT-based fuzzy classifier with FPGA. *J Supercomput* 76, 5849–5861 (2020). <https://doi.org/10.1007/s11227-019-03013-2>.
17. Neelakandan, S "Large scale optimization to minimize network traffic using MapReduce in big data applications". *International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, pp. 193-199, April 2016. DOI : 10.1109/ICCPEIC.2016.7557196
18. P. Subbulakshmi, M. Prakash, "Mitigating eavesdropping by using fuzzy based MDPOP-Q learning approach and multilevel Stackelberg game theoretic approach in wireless CRN" *Cognitive Systems Research*, Vol 52, pp 853-861, 2018.

Figures

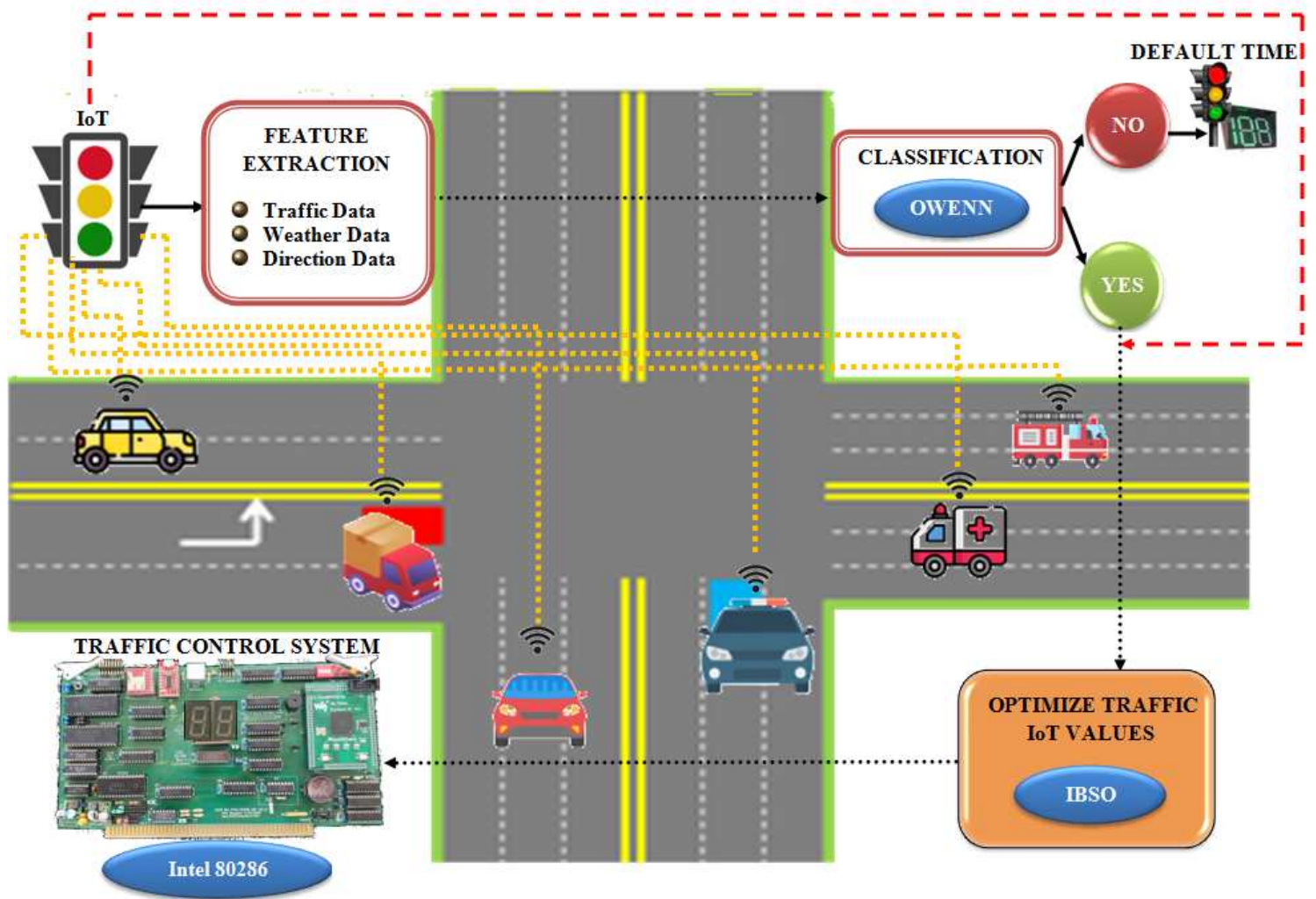


Figure 1

The system model

Input: Extracted feature set ($\bar{E}_{fs}^* = \{\bar{e}_1^*, \bar{e}_2^*, \bar{e}_3^*, \dots, \bar{e}_k^*\}$)

Output: Classified traffics

Begin

Initialize hidden layer $\bar{F}_{hi}^m(l)$, output layer $I(l)$, context layer $\bar{H}_{co}^m(l)$, weight κ_m

Set $m = 1$

While ($m < \max iter$)

Compute hidden layer using $\bar{F}_{hi}^m(l) = g(\kappa_1 \bar{H}_{co}^m(l) + \kappa_2 \bar{E}_{fs}^*(l-1))$

Compute output layer using $I(l) = \varphi(\kappa_3 \bar{F}_{hi}^m(l))$

Evaluate the weight value

Store previous history using context layer

Repeat unit reach the last layer

 {

Gives knowledge about the previous layer (i.e. context layer gives knowledge to $\bar{F}_{hi}^m(l)$ and $I(l)$)

 }

Set $m = m + 1$

End while

Return the traffic classified classes

End

Figure 2

Pseudocode for the OWENN algorithm

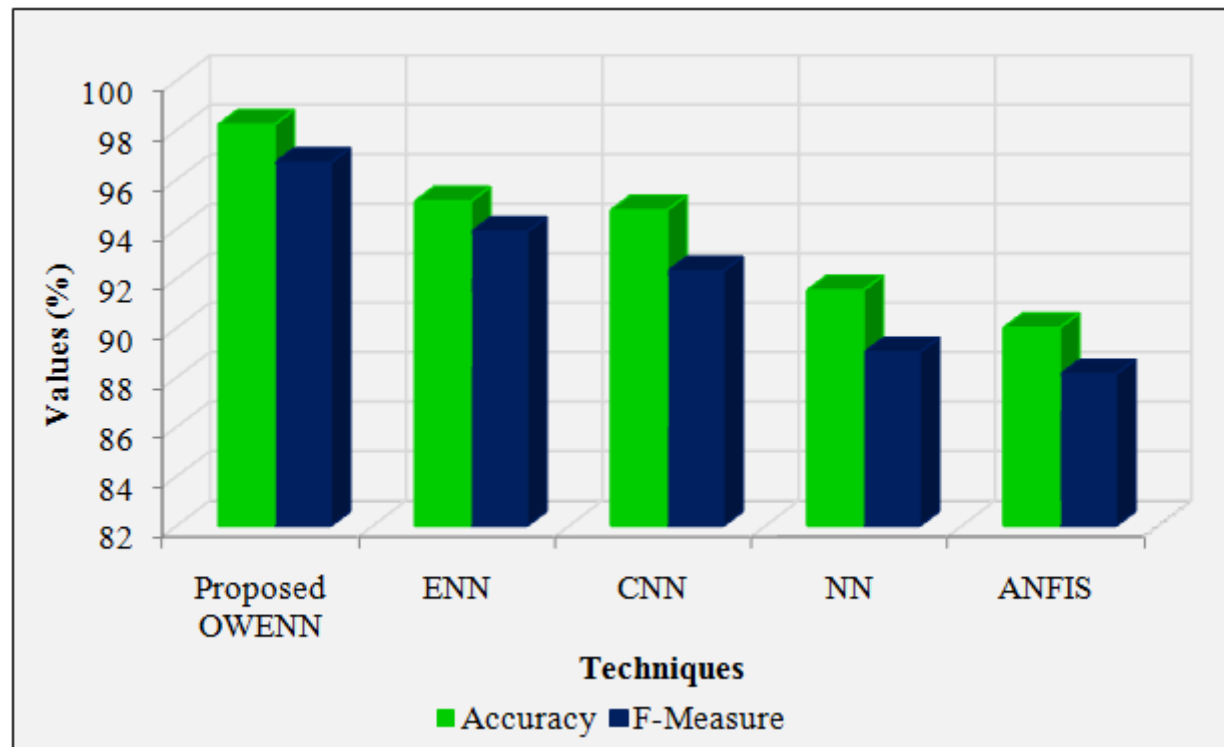
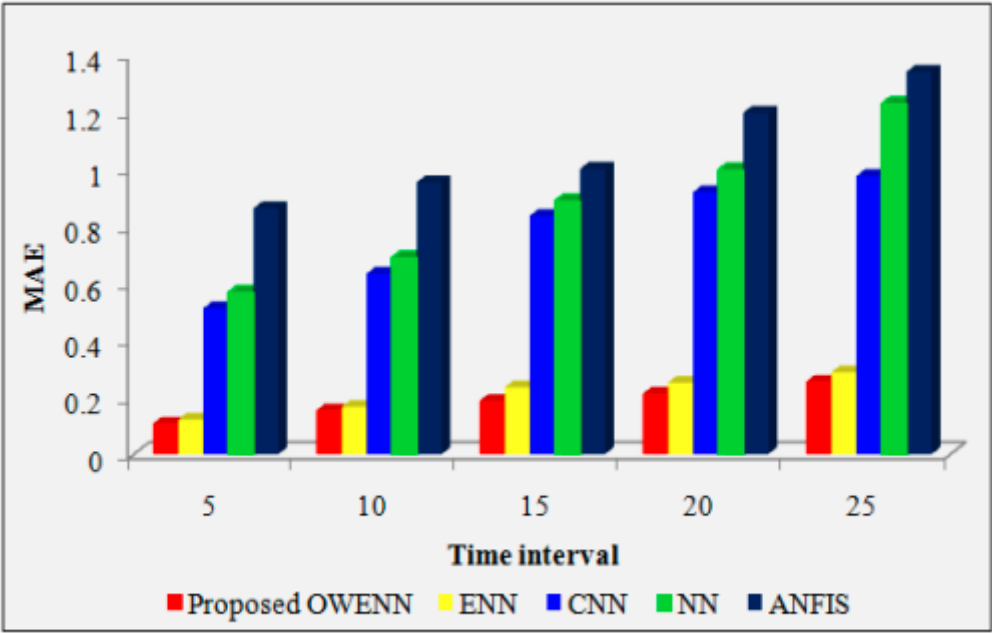


Figure 3

Accuracy and F-measure analysis



(a)

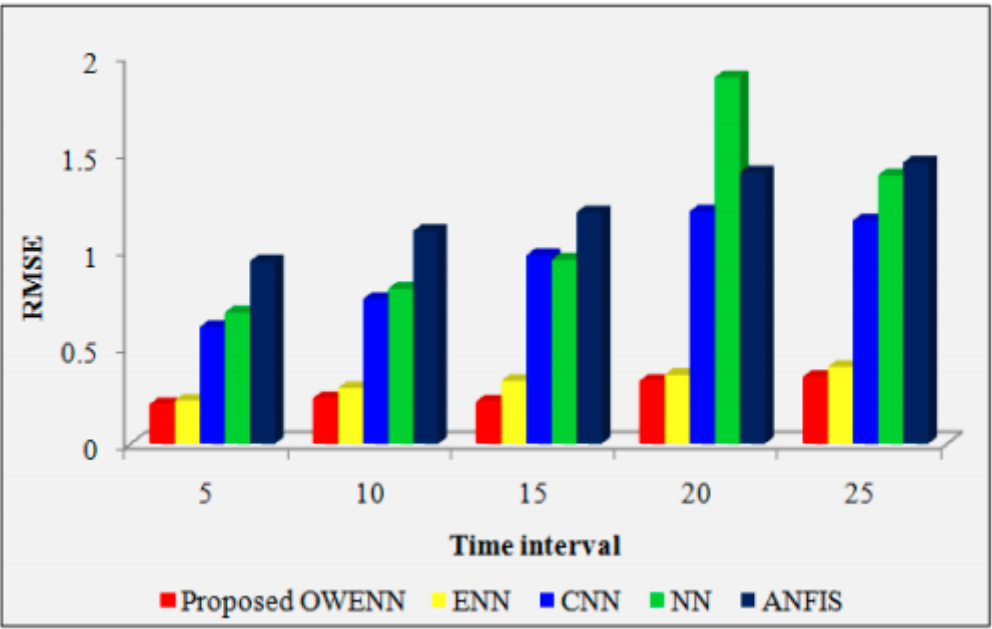


Figure 4

(a) MAE and (b) RMSE analysis with different time intervals