Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques

Muhammad Saleem a, Sagheer Abbas a, Taher M. Ghazal b,c, Muhammad Adnan Khan d,e,f, Nizar Sahawneh f, Munir Ahmad a

a School of Computer Science, National College of Business Administration and Economics, Lahore 54000, Pakistan
b Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia
c School of Information Technology, Skyline University College, University City Sharjah, 1797 Sharjah, UAE
d Riphah School of Computing & Innovation, Faculty of Computing, Riphah International University, Lahore Campus, Lahore 54000, Pakistan
e School of Business, Skyline University College, University City Sharjah, 1797 Sharjah, UAE
f Pattern Recognition and Machine Learning Lab, Department of Software, Gachon University, Seongnam 13557, South Korea

A R T I C L E   I N   F O
Article history:
Received 12 February 2022
Revised 11 March 2022
Accepted 17 March 2022
Available online xxxx

Keywords:
Vehicular networks
Smart city
Machine learning
Fusion

A B S T R A C T

Smart cities have been developed over the past decade, and reducing traffic congestion has been the top concern in smart city development. Short delays in communication between vehicles and Roadside Units (RSUs), smooth traffic flow, and road safety are the key challenges of Intelligent Transportation Systems (ITSs). The rapid upsurge in the number of road vehicles has increased traffic congestion and the number of road accidents. To fix this issue, Vehicular Networks (VNs) have developed many new ideas, including vehicular communications, navigation, and traffic control. Machine Learning (ML) is an efficient approach to finding hidden insights into ITS without being programmed explicitly by learning from data. This research proposed a fusion-based intelligent traffic congestion control system for VNs (FITCCS-VN) using ML techniques that collect traffic data and route traffic on available routes to alleviate traffic congestion in smart cities. The proposed system provides innovative services to the drivers that enable a view of traffic flow and the volume of vehicles available on the road remotely, intending to avoid traffic jams. The proposed model improves traffic flow and decreases congestion. The proposed system provides an accuracy of 95% and a miss rate of 5%, which is better than previous approaches.

© 2022 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Smart city development is coupled with a significant shift regarding planning and adopting advanced technologies to assemble "smarter" cities to enhance people’s quality of life. The European Commission launched innovative and apparent creativity on smart cities in four areas: electricity, buildings, heating and cooling systems, and transport [1]. An intelligent transportation solution can improve traffic flow in smart cities by monitoring traffic patterns and adjusting traffic signal timing. The aim is to ascertain and assist sustainable forms of transportation, to boost an Intelligent Transportation System (ITS) occupying real-time information, Traffic Management Systems (TMSs) to avoid congestion, safety, and green applications (e.g. to minimize the utilization of fuel, gas, and energy) [2]. ITS leverages novel and emerging technologies to make mobility more pleasant and cost-effective in a smart city.

In recent years, one of the fundamental dilemmas with regard to transportation systems has been traffic congestion, which must be solved to minimize fuel waste, accidents, traffic jams, and driver frustration. The majority of traffic delays in metropolitan areas occur because of the high number of vehicles. Traffic regulation
during rush hours is an important issue [3]. Transportation networks are becoming an integral part of human life due to the shortage of land resources and saturated transportation infrastructure in metropolitan areas. As a result of this overcrowding, numerous traffic-related drawbacks have arisen in urban zones where people must move quickly from one location to another [4].

With the recent advances in Artificial Intelligence (AI), and Machine Learning (ML) ITS, and smart environmental monitoring in smart cities monitor the factors influencing the environment more precisely, with an optimum control of pollution, traffic congestion and other adverse effects. Traffic congestion affects people’s quality of life because it reduces traffic performance and increases severe environmental pollution. Hence, the country’s production, economic development, and human activities are influenced by traffic congestion. The most pressing issue in urban planning is determining how to address traffic congestion effectively [5]. Traffic congestion management is a significant area of study, with many solutions emerging from miscellaneous research projects in the field over the last several decades [6]. As time passes, traffic data collection and intelligent transportation systems have evolved in response to these issues [7].

Traffic congestion management is critical for making it easier to drive on highways. Most traffic signals are preprogrammed. This does not support real-time conditions and causes traffic congestion. There is an increased need for density-driven traffic movement that is based on existing traffic conditions. Traffic junction signals play a crucial role in reducing congestion [8]. Another reason for traffic congestion is a lack of road infrastructure. The driver’s waiting time has risen as a result. This is primarily due to the inefficient operation of traffic signals. Approaches focused on Vehicle-to-Vehicle (V2V) communication are unable to reliably measure traffic congestion volume. Additionally, traffic signalling systems with a fixed operating period cannot handle shifting traffic volumes, resulting in long traffic queues at road crossings [9].

ITS includes intelligent traffic signal controls, highway management, and emergency services management. These systems capture real-time traffic data and proceed with required steps to prevent road traffic congestion. Traffic congestion on the roads of cities can be predicted precisely by smartphone applications such as Google and Apple Maps based on sensor data collected from highways and city road monitoring devices [10]. Real-time traffic alerts on exit roads may help the driver choose the best route from his present location. Drivers like to be informed about the state of congestion at upcoming intersections in order to save time.

The proposed system is based on VNs coupled with fused ML techniques, measures the density of traffic congestion, and provides the user with an alternative path for travelling by using a mobile application that can save time.

## 2. Literature review

In the past several years [2-7], a number of studies have focused on the topic of traffic congestion. Using Information and Communication Technology (ICT) [11] and Internet of Things (IoT) [12,13] applications, multiple types of research to monitor road traffic congestion and traffic control have been suggested in the literature to increase the efficiency of the current TCCS. Using a real-time dynamic traffic control scheme to route vehicles will improve traffic movement on roads, allowing facilities to be used more effectively and creating an atmosphere conducive to resolving city gridlock.

The Internet of Vehicles (IoV) is a network of vehicles equipped with sensors that may play a prime role in TMS by linking physical devices over the internet to provide more precise, quick, and accurate results. In IoV, every database is stored on a computer via the internet. Remote access to IoV components reduces human involvement [14].

VN is a specific case of wireless communication technology that is hampered by rapid topological changes due to the fast movement of vehicle nodes [15]. Given the rising quantity of vehicles prepared by wireless and communication devices, intravehicular communication has become an auspicious field of research. VNs empower many applications by providing wireless communication technology to vehicle nodes. This can involve accident prevention, dynamic route planning, and real-time traffic status monitoring. VN is a new mobile network (MN) that includes vehicles that manage themselves as mobile nodes. VN has been proposed to enhance safety and comfort with the aid of V2V and Vehicle-To-Roadside (V2R) communications. VNs have been created to improve driver safety and comfort with the assistance of V2V and V2R communications in vehicular environments [16,17]. VNs are now considered an infrastructure for ITSs with the growing number of autonomous vehicles in smart cities. The deployment of a VN is a solution to communicate between vehicles [18]. The architecture of a VN is presented in Fig. 1.

ML is an application of artificial intelligence that permits systems to automatically learn from data and make decisions without human assistance. ML enables systems to learn from experience and recover without being explicitly programmed. ML can mechanize and increase the efficiency of intelligent traffic congestion control systems (TCCSs) while reducing travelling costs more efficiently and accurately in a reliable way [19]. Data fusion is a process of combining data from one and multiple sources with insufficient raw data to gather precise, comprehensive, and unified entity information. At the decision level, fusion is used to produce a single decision after combining decisions from multiple sources to create a more intelligent decision on action. By taking the data patterns of various algorithms, the fusion of decision-making data and ML can help make better choices [20].

In [22], researchers suggested a real-time TMS composed of a small system of roadside units, junction units, and mobile units that determine the time of traffic lights to avoid gridlock development. This system also includes a web-based application for drivers of vehicles that uses data from real-time traffic monitoring to show the current traffic flow so that approaching vehicles take alternate routes to help reduce the gridlock. The limitations of this technique are a lack of performance and a secure mechanism for communication between nodes.

Reference [23] used IoT and sensing technology to create a framework for real-time traffic monitoring. Ultrasonic sensors...
were utilized to monitor traffic status in lanes. The controller collects this information from sensors and processes it. The processed data are then sent to the server via a Wi-Fi module. The traffic signal control system, which is based on perceiving traffic status in lanes, controls traffic. If a road has a significant amount of traffic, then it receives the highest priority, which means it takes a long time for vehicles to travel, and it will be given a long green-signal time. This proposed system is reliable, easy, and inexpensive. This presented system has limited calculation capacity and needs the proper execution of instructions by highly skilled programmers.

In [24], researchers developed a platform focused on Wireless Sensor Networks (WSNs) to acquire, fuse, and store city traffic data. The extended city intelligent transportation system is more versatile and efficient than other current city transportation systems. It is impossible to use WSNs for high-speed communication because they are designed for low-speed applications and are too expensive to build. In [25], the authors developed a vehicle traffic control mechanism via Usage-based Insurance (UBI) and smartphone-based measurement methods. This architecture is designed to model, perceive, and monitor traffic flow. This system has seven layers, starting with physical smartphones and servers and ending with the overall business strategy at the top.

In [26], researchers developed a model using a Random Forest Classification (RFC) algorithm of ML to construct a model for traffic congestion state perception. The RFC has high robustness, high efficiency, and a predictive accuracy of 87.5%. In addition, the generalization error is short and can be foreseen efficiently. The limitations of this research are that other machine learning techniques may provide more robust and accurate results. In [27], the authors developed a method for identifying road traffic congestion using GPS, a webcam, and an opinion poll. The sliding window technique extracted the vehicle’s movement patterns and fed them into an Artificial Neural Network (ANN) and J48. The J48 model performs better with 91.29% accuracy, which is lower than that of the proposed FITCCS-VN using machine learning techniques.

In [28], the authors developed an Unmanned Aerial Vehicle (UAV)-based traffic monitoring system using a Convolutional Neural Network (CNN). The camera on the UAV captured traffic images, and the system had a 91.67% accuracy based on the traffic situation. In [29], the authors developed a data-fusion-based TCCS based on a CNN and Long Short-Term Memory (LSTM) frameworks. The CNN was used to classify spatial data. LSTM for historical data had a 92.3% and 7.7% accuracy and miss rate, respectively, whereas the J48 model performs better with 91.29% accuracy, which is lower than that of the proposed FITCCS-VN using machine learning techniques.

Comparison of Previous Published Works with Proposed FITCCS-VN.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Preprocessing Layer</th>
<th>Use of VN</th>
<th>Decision-Making</th>
<th>Fused ML techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Saikar et al., 2017</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>V.S. Nagmode et al., 2017</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Y. Liu et al., 2017</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>T. Thaninwiet et al., 2010</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>L. Jian et al. 2010</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>S. Khan et al., 2021</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Proposed FITCCS-VN</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

3. Limitations of previous work

This paper presents a fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques. Traffic congestion and the number of traffic accidents have increased rapidly. Proposed model solved the problem of short delays in communication between vehicles and roadside units, smooth traffic flow, and road safety by intelligent transportation systems. The aim of this study is to provide innovative services to drivers that enable a view of traffic flow and the volume of vehicles available on the road remotely, intending to avoid traffic jams. This study develops a fusion-based intelligent traffic congestion control system for vehicular networks using machine learning technology to alleviate traffic congestion. This contribution offers pathways to improve traffic flow and decrease congestion.

Table 1 shows that in the research mentioned above, there are several limitations including a lack of performance, the need for a secure communication mechanism between nodes [22], limited computational capacity [23], and lower accuracy [26,27,28,29].

The proposed techniques play a vital role in providing a secure communication mechanism between nodes by using a VN architecture, improving performance through a preprocessing layer to mitigate noisy data and achieving higher accuracy, higher execution capacity, and more robust decision-making by including fused ML techniques.

4. Proposed fusion-based intelligent traffic congestion control system

In this research, an intelligent TCCS is used for smart cities to monitor and control traffic congestion using machine learning techniques. This research proposed a FITCCS-VN using ML techniques. Figs. 2 and 3 present a complete picture of the proposed FITCCS-VN in which data are collected via IoT-enabled devices. This system allows signals from one junction to send and update data to another junction. Subsequently, the sensory layer receives data from sensors, and then these sensing values are passed through the preprocessing, training, performance, and validation phases.

Fig. 2 shows that VN has become a prominent network model with the development of communication technology. VN enables vehicles to broadcast with V2V and exchanges information with the Vehicle to Infrastructure (V2I). The collection of sensory data from wireless communication vehicles instead of costly cellular communication with the help of VN thus becomes possible. First, data can be assembled from roadside base stations via V2V or V2I communication. Then, the base station (server) can transfer the collected data directly to the data centre to further monitor and route traffic congestion.

Fig. 2 also depicts a proposed model in which the traffic congestion data sensed from the VN are received from the sensory layer and passed through preprocessing training and performance lay-
ers. After the performance layer, the output is forwarded to edge computing. In the validation phase, the learned data are imported from edge computing to further predict whether there is traffic congestion.

As shown in Fig. 3, training and validation are the two phases of the proposed FITCCS-VN using ML techniques. Each step is further divided into stages. The 1st phase collects the dataset from various traffic control sensors installed on multiple VNs. To implement the proposed research model, a prelabelled VN dataset is selected. This dataset contains 2282 instances and 515 features, of which 514 attributes are independent and one, the output class, is dependent. The next layer is preprocessing, which mitigates the noisy data using moving averages and normalization. Then, the preprocessed data are divided into 70% training and 30% testing datasets. After this process, the training data are sent to the training layer, whereas the testing dataset is stored in edge computing. A classification process is performed in the training layer to predict the traffic congestion using both ML techniques [ANN and support Vector Machine (SVM)]. Each neuron has a function of activation, such as \( f(x) = \text{Sigmoid}(x) \) in the hidden layer. The sigmoid function for \( i \)th input and the hidden layer of the proposed FITCCS-VN can appear as:

\[
\begin{align*}
\sigma_i &= \frac{1}{1 + e^{-p_2 \sum_{j=1}^{n} (w_{ij} + t_{ij} \sum_{k=1}^{n} (y_{ik} p_3))}} \\
\text{where} & \\
& j = 1, 2, 3 \ldots n \\
\end{align*}
\]

The minimum mean square error can be calculated as below:

\[
A = \frac{1}{2} \sum_{x} (\hat{y}_x - y_x)^2 
\]

where \( \hat{y}_x \) shows the estimated output, and \( y_x \) is a deliberate output. Both layers’ weights in change can be calculated as below:

\[
\Delta W = - \frac{\partial A}{\partial W} \\
\Delta w_{jk} = - \frac{\partial A}{\partial w_{jk}}
\]

In the above equation, \( r_i \), \( w_{ij} \), \( p_1 \), \( t_{ij} \), and \( p_2 \) represent the input features, weights amongst the \( i^{th} \) input and \( j^{th} \) hidden layer neurons, bias of hidden layers, weights between the \( j^{th} \) hidden layer and \( k^{th} \) output layer neurons, and the bias of output layer, respectively. These are listed in Table 2.

The minimum mean square error can be calculated as below:
Eq. (3) can be written as.

$$\Delta l_{b,i} = -\epsilon \frac{\partial A}{\partial b_i} - \frac{\partial b_k}{\partial \psi_i} \cdot \frac{\partial \psi_i}{\partial A}$$  \hspace{1cm} (4)$$

The above equation after simplification can be written as.

$$\Delta a_{k,j} = \epsilon \left[ \sum_{i} (x_i - b_n) \cdot x_i \cdot (1 - b_n) \cdot (1 - b_n) \right] \times b_j(1 - b_j) \times \hat{r}_i$$

$$\Delta a_{k,i} = \epsilon \left[ \sum_{i} (x_i - b_n) \cdot x_i \cdot (1 - b_n) \cdot (1 - b_n) \right] \times b_j(1 - b_j)$$

$$= \epsilon \left[ \sum_{i} \xi_i(v_j) \right] \times b_j(1 - b_j) \times \hat{r}_i$$

$$\Delta a_{k,j} = \epsilon \bar{\xi}_i \hat{r}_i$$  \hspace{1cm} (5)$$

where

$$\bar{\xi}_i = \left[ \sum_{i} \xi_i(v_j) \right] \times b_j(1 - b_j)$$

The hidden and output layers are presented in Eq. (6), which updates the weight and bias.

$$v_{j,k}^+ = v_{j,k} + \lambda \Delta a_{k,j}$$  \hspace{1cm} (6)$$

Updating the weight and bias among the input layer and hidden layer is presented in Eq. (7):

$$\omega_{ij} = \alpha_{ij} + \lambda \Delta a_{ij}$$  \hspace{1cm} (7)$$

$$\lambda$$ denotes the learning rate of FITCCS-VN using ML techniques. The convergence of FITCCS-VN using ML techniques depends upon the careful selection of $\lambda$.

As we know, during SVM the line equation is.

$$x = m_1y + c$$  \hspace{1cm} (8)$$

In Eq. (8), 'r' represents the line slope and 'c' the intercept. Hence,

$$m_1 = x + c$$

Let $\tilde{t} = (u, x)^T$ and $\bar{y} = (u, -1)$. Then, the equation becomes.

$$\bar{y} \cdot \tilde{t} + c = 0$$  \hspace{1cm} (9)$$

This equation comes from two-dimensional vectors. However, Eq. (9), defined as the hyperplane, performs for any number of dimensions. The direction of a vector $\tilde{t} = (u, x)^T$ is $\bar{y}$ and is distinct as.

$$\bar{y} = \frac{u}{||t||} + \frac{x}{||t||}$$  \hspace{1cm} (10)$$

where

$$||t|| = \sqrt{u^2 + x^2}$$

As we know that

$$cos(\theta) = \frac{u}{||t||} \text{ and } cos(\mu) = \frac{x}{||t||}$$

Eq. (10) can also be written as.

$$\bar{y} \cdot \tilde{t} = ||t|| ||\bar{y}|| \cos(\theta)$$

$$\cos(\theta) = \cos \left( \frac{\mu}{\bar{y}} - \mu \right) = \cos \left( \frac{\mu}{\bar{y}} \right) + \sin \left( \frac{\mu}{\bar{y}} \right) \sin(\mu)$$

$$= \frac{\mu}{\bar{y}} \frac{\mu}{\bar{y}} + \frac{\mu}{\bar{y}} \frac{\mu}{\bar{y}}$$

$$\bar{y} \cdot \tilde{t} = \sum_{i=1}^{n} y_i t_i$$  \hspace{1cm} (11)$$

The dot product can be compared using Eq. (11) for $\bar{y}$ dimensional vectors:

Let

$$B = M (\bar{y} \cdot \tilde{t} + \zeta)$$

If sign (B) > 0, then this is appropriately classified; and if sign (B) < 0, then it is imperfectly classified.

Calculate $\tilde{t}$ on a training dataset by dataset $\Pi$:

$$B_i = M_i (\bar{y} \cdot \tilde{t} + \zeta)$$

$p$ is the functional margin of the dataset.

$$p = \min_{i=1}^{n} B_i$$

When comparing hyperplanes, one through the largest $p$ will be chosen. $p$ is the geometric margin of the dataset. The goal is to discover an optimal hyperplane, which means finding the optimal hyperplane values of $\bar{y}$ and $B$. Lagrangian function:

$$\Lambda (\bar{y}, \zeta, \mu) = \frac{1}{2} \bar{y} \cdot \bar{y} - \sum_{i=1}^{n} \mu_i [M : (\bar{y} \cdot \tilde{t} + \zeta) - 1]$$

$$\nabla_y \Lambda (\bar{y}, \zeta, \mu) = \bar{y} - \sum_{i=1}^{n} \mu_i M_i t_i = 0$$  \hspace{1cm} (12)$$

$$\nabla_z \Lambda (\bar{y}, \zeta, \mu) = - \sum_{i=1}^{n} \mu_i M_i = 0$$  \hspace{1cm} (13)$$

From Eqs. (16) and (17), we get.

$$\bar{y} = \sum_{i=1}^{n} \mu_i M_i t_i$$ and $\sum_{i=1}^{n} \mu_i M_i = 0$  \hspace{1cm} (14)$$

while substituting the Lagrangian function $\Lambda$:
In the validation phase, the test data stored in edge computing and the learned patterns are imported from the edge database and referred to the ML techniques to predict whether traffic congestion is found. If the answer is ‘No’, then the process is discarded; and if the answer is ‘Yes’, then the message indicates that traffic congestion has been found.

The fusion approach using ML techniques develops and applies fuzzy logic to optimized classification algorithms. ML techniques (ANN and SVM) produce logic using fuzzy rules. The conditional statements to create fuzzy logic are given below.

1. If ANN is No and SVM is No, then FITCCS-VN is No
2. If ANN is No and SVM is Yes, then FITCCS-VN is Yes
3. If ANN is Yes and SVM is No, then FITCCS-VN is Yes
4. If ANN is Yes and SVM is Yes, then FITCCS-VN is Yes

Fig. 4 illustrates that if the value of SVM lies between 60 and 100 and ANN lies between 60 and 100, then FITCCS-VN is good (yellow). If SVM lies between 40 and 60 and ANN lies between 40 and 60, then FITCCS-VN is satisfactory (green). If SVM lies between 0 and 40 and ANN lies between 0 and 40, then FITCCS-VN is satisfactory (blue).

Fig. 5 demonstrates that if the value of ANN is no and SVM is no, then the proposed FITCCS-VN is no. Fig. 6 shows that if the value of ANN is yes and SVM is no, then the proposed FITCCS-VN is yes. Fig. 7 demonstrates that if the value of ANN is yes and SVM is yes, then the proposed FITCCS-VN is yes.

5. Simulation results

In this proposed FITCCS-VN using ML techniques, a TCCS is implemented on a dataset [52]. The data were divided randomly into 70% training (1879 samples) and 30% validation (403 samples). The proposed system calculated the output using multiple statistical measures, as described in Equations (23) through (31).

\[
\text{Sensitivity} = \frac{\sum \text{TruePositive}}{\sum \text{ConditionPositive}}
\]  
\[
\text{Specificity} = \frac{\sum \text{TrueNegative}}{\sum \text{ConditionNegative}}
\]  
\[
\text{Accuracy} = \frac{\sum \text{TruePositive} + \sum \text{TrueNegative}}{\sum \text{TotalPopulation}}
\]  
\[
\text{Miss Rate} = 1 - \text{Accuracy}
\]  
\[
\text{Fallout} = \frac{\sum \text{FalsePositive}}{\sum \text{ConditionNegative}}
\]
\[
\text{Likelihood Positive Ratio} = \frac{\sum \text{True Positive Ratio}}{\sum \text{False Positive Ratio}}
\]

(28)

\[
\text{Likelihood Negative Ratio} = \frac{\sum \text{True Positive Ratio}}{\sum \text{False Positive Ratio}}
\]

(29)

\[
\text{Positive Predictive Value} = \frac{\sum \text{True Positive}}{\sum \text{Predicted Condition Positive}}
\]

(30)

\[
\text{Negative Predictive Value} = \frac{\sum \text{True Negative}}{\sum \text{Predicted Condition Negative}}
\]

(31)

Tables 3, 4, 5, and 6 show the training and validation of ML techniques (ANN and SVM) in terms of accuracy and miss rate. In addition to comparisons, the various statistical measures used for performance are deliberate from diverse metrics named as accuracy, sensitivity, specificity, miss-rate, fall-out, Likelihood Positive Ratio (LR+), Likelihood Negative Ration (LR-), Precision, and negative predictive value. By contrast, the True Positive Rate (TPR) is expressed as sensitivity, True Negative Rate (TNR) as specificity, False Negative Rate (FNR) as miss-rate, False-Positive Rate (FPR) as fall-out, and Positive Predictive Value (PPV) as precision.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Training of proposed FITCCS-VN using the ML technique (ANN) when identifying traffic congestion.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proposed Model Training</strong></td>
<td><strong>Input</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Validation of proposed FITCCS-VN using ML technique (ANN) when identifying traffic congestion.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proposed Model Validation</strong></td>
<td><strong>Input</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5
Training of proposed FITCCS-VN using ML technique (SVM) when identifying traffic congestion.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Expected Total samples</th>
<th>Actual</th>
<th>Output</th>
<th>True Positive</th>
<th>False Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>293 Positive</td>
<td>228</td>
<td>230</td>
<td>56</td>
<td>182</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1586 Negative</td>
<td>56</td>
<td>1586</td>
<td>293</td>
<td>1293</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6
Validation of proposed FITCCS-VN using ML technique (SVM) when identifying traffic congestion.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Total samples</th>
<th>True Positive</th>
<th>False Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>53 Positive</td>
<td>18</td>
<td>403</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>350 Negative</td>
<td>21</td>
<td>53350</td>
<td>293</td>
<td>53057</td>
</tr>
</tbody>
</table>

Table 3 shows the proposed FITCCS-VN using ML techniques within the training phase. A total of 1879 samples are used within the training separated into 293 and 1586 positive and negative samples, respectively. It is ascertained that 237 samples are appropriately predicted as positive and that there is no traffic congestion, but 56 records are inaccurately predicted as negative, which means there is traffic congestion. Similarly, a total of 1586 samples are collected, where negative indicates traffic congestion, in which 339 samples are appropriately predicted as negative, meaning there is traffic congestion, and 11 samples are wrongly predicted as positive, which means there is no traffic congestion.

Table 4 shows the proposed FITCCS-VN using ML techniques within the validation phase. A total of 403 samples were used within the validation, divided into 53,350 positive and 53,350 negative samples. It is ascertained that 40 samples are appropriately predicted as positive and that there is no traffic congestion, but 13 records are inaccurately predicted as negative, which indicates traffic congestion. Correspondingly, a total of 350 samples are collected, where negative indicates traffic congestion, in which 339 samples are appropriately predicted as negative, meaning there is traffic congestion, and 11 samples are wrongly predicted as positive, which means there is no traffic congestion.

Table 5 shows the proposed FITCCS-VN using ML techniques within the training phase. A total of 1879 samples were used within the training, separated into 293 and 1586 positive and negative samples, respectively. It is ascertained that 228 samples are correctly predicted as positive and that there is no traffic congestion, but 65 records are incorrectly predicted as negative, which indicates traffic congestion. Similarly, a total of 1586 samples are collected, where negative indicates traffic congestion, in which 1530 samples are appropriately predicted as negative, meaning there is traffic congestion, and 56 samples are wrongly predicted as positive, which means there is no traffic congestion.

Table 6 shows the proposed FITCCS-VN using ML techniques within the validation phase. A total of 403 samples were used within the validation, divided into 53,350 positive and 53,350 negative samples. It is ascertained that 35 samples are appropriately predicted as positive and that there is no traffic congestion, but 18 records are inaccurately predicted as negative, which means there is traffic congestion. Correspondingly, a total of 350 samples are collected, where negative means there is traffic congestion, in which 329 samples are appropriately predicted as negative, which means there is traffic congestion, and 21 samples are wrongly perceived as positive, which means there is no traffic congestion.

Table 7 shows the performance of the proposed FITCCS-VN using ML techniques in terms of accuracy, sensitivity, specificity, miss rate, and precision within the training and validation phases w.r.t. ANN and SVM, respectively. This indicates that the proposed system using the ANN approach within training provides 95.4%, 80.8%, 98.0%, 6.0%, and 96.8% accuracy, sensitivity, specificity, miss rate, and precision. Within validation, the proposed system provides 94.0%, 75.4%, 96.8%, 6.0%, and 78.4% accuracy, sensitivity, specificity, miss rate, and precision, respectively. In addition, more statistical measures of the proposed system can predict values such as fall-out, likelihood positive ratio, likelihood negative ratio, and negative predictive value within training. The results were 0.019, 42.526, 0.047, and 98.5% and validations of 0.031, 24.323, 0.062, and 96.3%, respectively.

Moreover, the simulation results of the fused ML techniques of the proposed FITCCS-VN are listed in Table 8. Twenty random values were deployed using the fusion of ANN and SVM, of which 19 were precise according to human decision-making principles of FITCCS-VN. In contrast, one matter was thought to be low but was indicated as normal per the proposed FITCCS-VN using ML techniques and incorrectness. The probability of correctness or accuracy of the proposed FITCCS-VN using ML techniques is found to be 95%, and the miss rate is 5%. This indicates that the proposed system with the SVM approach within training provides 93.5%, 77.8%, 96.4%, 6.5%, and 80.2% in terms of accuracy, sensitivity, specificity, miss rate, and precision. Within validation, the proposed system provides 90.3%, 66.0%, 94.0%, 9.7%, and 62.5% accuracy, sensitivity, specificity, miss rate, and precision, respectively. In addition, more statistical measures of the proposed system can predict values such as fall-out, likelihood positive ratio, likelihood negative ratio, and negative predictive value within training. These results are 0.035, 22.045, 0.281, and 95.9% and validation of 0.06, 11.005, 0.361, and 94.8%, respectively.

Table 10 shows a performance comparison of the proposed FITCCS-VN using the ML techniques with previous approaches. The accuracies and miss rates of the ANN, SVM, and fusion-based FITCCS-VN are 94%/6%, 90.3%/9.7%, and 95%/5%, respectively.

Table 9 elaborates on the simulation of the proposed FITCCS-VN with fused machine learning techniques. The accuracies and miss rates of the ANN, SVM, and fusion-based FITCCS-VN are 94%/6%, 90.3%/9.7%, and 95%/5%, respectively.

Table 10 shows a performance comparison of the proposed FITCCS-VN using the ML techniques with previous approaches.
Table 8
Fusion results of proposed FITCCS-VN using ML techniques (ANN and SVM).

<table>
<thead>
<tr>
<th></th>
<th>ANN</th>
<th>SVM</th>
<th>The proposed FITCCS-VN using ML techniques</th>
<th>The human expert decision of FITCCS-VN</th>
<th>Probability of correctness</th>
<th>Probability of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.4 (N)</td>
<td>20.5 (N)</td>
<td>23.1 (N)</td>
<td>No</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>28.9 (N)</td>
<td>7.73 (N)</td>
<td>23.1 (N)</td>
<td>No</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>17.9 (N)</td>
<td>65 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>10.6 (N)</td>
<td>13.2 (N)</td>
<td>23.1 (N)</td>
<td>No</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>24.3 (N)</td>
<td>22.3 (N)</td>
<td>23.1 (N)</td>
<td>No</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>10.6 (N)</td>
<td>64.1 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>74.8 (Y)</td>
<td>13.2 (N)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>81.2 (Y)</td>
<td>68.6 (N)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>72 (Y)</td>
<td>68.6 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>12.4 (N)</td>
<td>20.5 (N)</td>
<td>23.1 (N)</td>
<td>No</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>24.3 (N)</td>
<td>7.73 (N)</td>
<td>23.1 (N)</td>
<td>No</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>77.5 (Y)</td>
<td>15 (N)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>17.9 (N)</td>
<td>79.5 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>91.3 (Y)</td>
<td>90.5 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>24.3 (N)</td>
<td>66.8 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>83.9 (Y)</td>
<td>65 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>77.5 (Y)</td>
<td>13.2 (N)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>77.5 (Y)</td>
<td>66.8 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>93.1 (Y)</td>
<td>79.5 (Y)</td>
<td>72.1 (Y)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>27.1 (N)</td>
<td>84.1 (Y)</td>
<td>23.1 (N)</td>
<td>Yes</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 9
ANN and SVM results of proposed FITCCS-VN.

<table>
<thead>
<tr>
<th></th>
<th>ANN</th>
<th>SVM</th>
<th>Fusion based FITCCS-VN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>94.0</td>
<td>90.3</td>
<td>95</td>
</tr>
<tr>
<td>Miss Rate (%)</td>
<td>6.0</td>
<td>9.7</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 10
Comparison results of proposed FITCCS-VN using ML techniques with literature.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Miss Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thianniwet T. et al. (2010), [27]</td>
<td>Training 91.29</td>
<td>Validation 90.11</td>
</tr>
<tr>
<td>Khan, Sulaiman et al., (2021) [29]</td>
<td>Training 93.6</td>
<td>Validation 92.3</td>
</tr>
<tr>
<td>Pushpi and Dilip Kumar (2018) [51]</td>
<td>Training 91.2</td>
<td>Validation 90.6</td>
</tr>
<tr>
<td>Proposed system using ML technique</td>
<td>Training 95.4</td>
<td>Validation 90.3</td>
</tr>
<tr>
<td>Proposed system using ML technique (ANN)</td>
<td>Validation 94.0</td>
<td></td>
</tr>
<tr>
<td>Proposed system using ML technique (SVM)</td>
<td>Training 93.5</td>
<td>Validation 90.3</td>
</tr>
<tr>
<td>Proposed FITCCS-VN using ML techniques</td>
<td>Validation 95</td>
<td>5</td>
</tr>
</tbody>
</table>

It is indicated that the proposed techniques provide accurate results compared to previously published approaches.

6. Conclusion

A Vehicular Network (VN) is a self-organized, service-oriented, multipurpose communication network that enables communication between vehicles and roadside infrastructure for message exchange. In a dense traffic scenario, the load generated by the traffic may exceed the road’s capacity, causing traffic congestion. This research proposed a fusion-based FITCCS-VN using ML techniques to assemble data from an IoV-enabled VN, and then evaluated it intelligently to predict and control traffic congestion. Evaluation of the simulation results indicated that the proposed FITCCS-VN using ML techniques exhibits 95% accuracy and a 5% miss rate, which are better than those of previous approaches [27,28,29,51]. In the future, the proposed system accuracy may be improved by using federated learning and Alexnet.

References
