sciendo

Journal of Smart Internet of Things (JSIoT)
VOL 2023, No.02 | 81-96 | 2023
DOI: 10.2478/jsiot-2022-0006

Article

Improved Arithmetic Optimization with Deep Learning Driven Traffic Congestion Control for Intelligent Transportation Systems in Smart Cities

Muntather Hassan Almusawy[1]

[1] Department of Smart Grid Control, Ministry of Electricity, Baghdad, Iraq

*Corresponding author: Muntather Hassan Almusawy (email: muntatheralmusawi@gmail.com)

Received 10 November 2022; Accepted 01 December 2022; Issue Published 15 December 2022

Abstract: In the last few years, some progress had been made in smart cities, and reduction in traffic congestion was the topmost concern in the development of smart cities. Shorter delays in transmission between Roadside Units (RSUs) and vehicles, road safety, and smooth traffic flow are the major difficulties of Intelligent Transportation Systems (ITS). The rapid improvement in automobiles occurs which increased the number of road accidents and traffic congestion. Machine Learning (ML) was an advanced technique to find hidden insights into ITSs without being explicitly programmed by learning from datasets. This article introduces an Improved Arithmetic Optimization with Deep Learning Driven Traffic Congestion Control (IAOADL-TCC) for ITS in Smart Cities. The presented IAOADL-TCC model enables traffic data collection and route traffic on existing routes for avoiding traffic congestion in smart cities. To accomplish this, the IAOADL-TCC model employs hybrid convolution neural network attention long short-term memory (HCNN-ALSTM) method for traffic congestion control. In addition, IAOA based hyperparameter tuning strategy is derived to optimally modify the hyperparameter values of the HCNN-ALSTM model. The presented IAOADL-TCC model effectively enhances the flow of traffic and reduces congestion. The experimental result study of the IAOADL-TCC method can be tested by making use of road traffic dataset from Kaggle repository. The experimental outcome stated the better performance of the IAOADL-TCC model over other DL methods.

Keywords: Smart cities; intelligent transportation systems; traffic prediction; traffic congestion; deep learning

Copyright © 2023 Journal of Smart Internet of Things (JSIoT) published by Future Science for Digital Publishing and Sciendo . This is an open access article license CC BY (https://creativecommons.org/licenses/by/4.0/)

1. Introduction

Due to the growth of urbanization and the popularity of vehicles, transportation issues become

very challenging: traffic flow (TF) can be congested, accidents occur frequently, and traffic atmosphere is worsening [1]. One effective way for enhancing the traffic atmosphere is establishing an effectual and precise transport system, which could aid in better arrangement of transport resources, diffuse TF before it was overloaded, and further offer copious on road entertainment. The Intelligent Transportation System (ITS) was one such popular among such mechanism [2, 3]. ITS becomes a complicated mechanism that can be combined with various advanced technologies, for example, transportation communication mechanisms. At the same time, using the benefit of the advancement of 5G transmission system, on-road sensors, and so on. ITS could enhance traffic efficacy, reduce traffic accidents, ease traffic congestion, reduce environmental pollution, and increase capacity of roads [3]. With ITS, an efficient and accurate road traffic prediction mechanism could offer continual and accurate road status data on the basis of past road situations. That data is helpful for the applications indulged in ITS, namely vehicular cloud (VC), traffic congestion control, traffic light control, and much more [4].

Traffic congestion is an excess automobile on a portion of roadway at particular time that leads to longer trip times and slower speeds and it turns out to be main challenge in transportation planning and traffic management [5]. Diminishing traffic congestion would grant safety to people, minimize road accidents, dimmish fuel utilization, aids in air pollution control, lessens waiting hours and comprehend smooth motion of cars in the transportation routes, and help in offering the data which is needed for future road analysis and planning [6]. There occur various causes of traffic congestion, namely inadequate capacity, uncontrolled demands, huge red-light delay, and difficulties on the road like double parking, accidents, road work, road narrowing down, and random vehicle stops. Traffic produces vast volumes of data which were collected from distinct kinds of devices, like intelligent sensors and cameras [7]. Hence, there was no problem in gathering such data; the issue is how to process, store, analyze manage and handle the raised amounts of traffic data to make useful use of it.

The above-mentioned techniques majorly aim at examining vast volume of traffic data for extracting some prospects of traffic data -including but not limited to traffic volume, traffic speed, average waiting time, and vehicle arrival rate [8]. Data mining was the process of discovering, analyzing, and predicting exciting knowledge and hidden paradigms from huge volumes of data which is stored in repositories, like data warehouses and databases [9]. This procedure invokes statistical methods, mathematical techniques, and machine learning (ML) techniques. Utilizing data mining technology in traffic management offers a robust processing function and analysis of mass traffic data and guides drivers and mechanisms in making superior decisions [10]. Knowledge mining and discovery were an evolving one in traffic management systems aimed at analyzing and using huge quantity of traffic data that is utilized for traffic control, route programming, or route guidance.

This article introduces an Improved Arithmetic Optimization with Deep Learning Driven Traffic Congestion Control (IAOADL-TCC) for ITS in Smart Cities. The presented IAOADL-TCC model employs hybrid convolution neural network- attention long short-term memory (HCNN-ALSTM) algorithm for traffic congestion control. In addition, IAOA based hyperparameter tuning strategy is derived to optimally modify the hyperparameter values of the HCNN-ALSTM model. The presented IAOADL-TCC model effectively enhances the flow of traffic and reduces congestion. The experimental result analysis of the IAOADL-TCC method can be tested by making use of road traffic dataset from Kaggle repository.

2. Literature Review

In [11], the authors proposed a consortium blockchain-related federated learning architecture to allow secure, decentralized, and reliable federated learning without a central model coordinator.

In this work, the method upgrade from distributed vehicle is proved by miner to avoid undependable model update and was later stored on blockchain. In [12], Restricted Boltzmann Machine (RBM) as the approach for TFP that was a typical method depends on DL architecture. Taking traffic big data aggregation in IoV, multi-media technology offer real sample data sufficiently for training method. RBM generates long run mechanism of polymorphic for chaotic time sequences, by use of phase space reconstruction to identify the information. In [13], a TFP architecture for urban road networks depends on DL method has been introduced. Initially, feature engineering is presented for extracting the feature from a considerable amount of traffic datasets, with the eliminated anomaly nodes. Then, the big traffic data was compressed by spectral clustering compression system. At last, we developed a hybrid TFP system based on Sparse Auto-Encoder (SAE) and LSTM (Long Short Term Memory).

Wei et al. [14] developed a novel TFP methodology named Auto-Encoder LSTM (AE-LSTM) predictive model. In this work, the AE is utilized for obtaining the internal relationships of TF through feature extraction of upstream and downstream traffic datasets. Furthermore, the LSTM network makes use of the obtained historical and the feature dataset for predicting complicated linear traffic dataset. Alam et al. [15] gathered the real-time traffic dataset and employed five regression mechanisms: Sequential Minimal Optimization (SMO) Regression, Linear Regression, M5P model tree, RF, and Multilayer Perceptron to forecast or predict the TF. In [16], a short-term TFP approach of Quantum GA -Learning Vector Quantization (QGA-LVQ) neural network was developed for forecasting the variations of TFs. Unlike BPNN, Learning Vector Quantization (LVQ) neural network was easier implementation, good clustering effect, and simple structure. Using the global optimization capability of QGA, it is integrated into LVQ-NN for overcoming limitations of LVQ-NN involving sensitivity to initial weight and susceptibility to local minimal. In [17], the author presents a new hybrid DL method by utilizing the deep aggregation structure (the sequence to sequence architecture) and Graph Convolution Network (GCN) of Gated Recurrent Unit (GRU). In the meantime, to resolve the real-time predictive problems, that is the online prediction task, we presented a new online predictive approach by utilizing refinement learning.

3. The Proposed Model

In this study, a new IAOADL-TCC algorithm was projected for effective control of traffic congestion in the ITS environment. The presented IAOADL-TCC model initially pre-processed the traffic data. In addition, the HCNN-ALSTM model is utilized to control traffic congestion. Finally, the IAOA based hyperparameter optimizer is derived to optimally tune the hyperparameters related to the HCNN-ALSTM model. Fig. 1 illustrates the overall process of IAOADL-TCC approach.

3.1. Data Pre-processing

In the initial phase, the presented IAOADL-TCC algorithm pre-processed the traffic dataset. The z-score is a classical normalization and standardization methodology that characterizes the quantity of standard deviations,· a raw data point was below or above the population mean. It ranges from [-3, 3]. It regularizes the data to the abovementioned scale to transform each dataset with diverse scales to default scale. For normalizing dataset employing z-score, it is divided by the standard deviation that provides a score ranging from [-3, 3] and it subtracts the mean of population from a raw data point, thus, reflecting how much standard deviation a point was below or above mean as computed by Eq. (1), whereby x characterizes the value of specific sample, μ characterizes mean and σ characterizes standard deviation [16].

$$z_score = \frac{(x-\mu)}{\sigma} \quad (1)$$

3.2 Design of HCNN-ALSTM based Traffic Congestion Control

Next to data pre-processing, the HCNN-ALSTM model is utilized to control traffic congestion. The HCNN-ALSTM element becomes a crucial component of the presented technique to extract the spatiotemporal attribute of data TF. The LSTM contains two LSTM layers and CNN contains two convolution layers correspondingly. The input of HCNN-ALSTM was spatiotemporal data TF matrix X_t^s as mentioned in Eq. (2) infers the old data TF. For extracting the spatial attributes, $1D$ convolution kernel has been implemented onX_t^s data flow at t time step. In $1D$ convolutional kernels, filter has been used for obtaining the local perceptual fields through the sliding filters. The convolutional kernel filtering is defined in the following [17]:

$$Y_t^s = \sigma(W_s * X_t^s + b_s), \qquad (2)$$

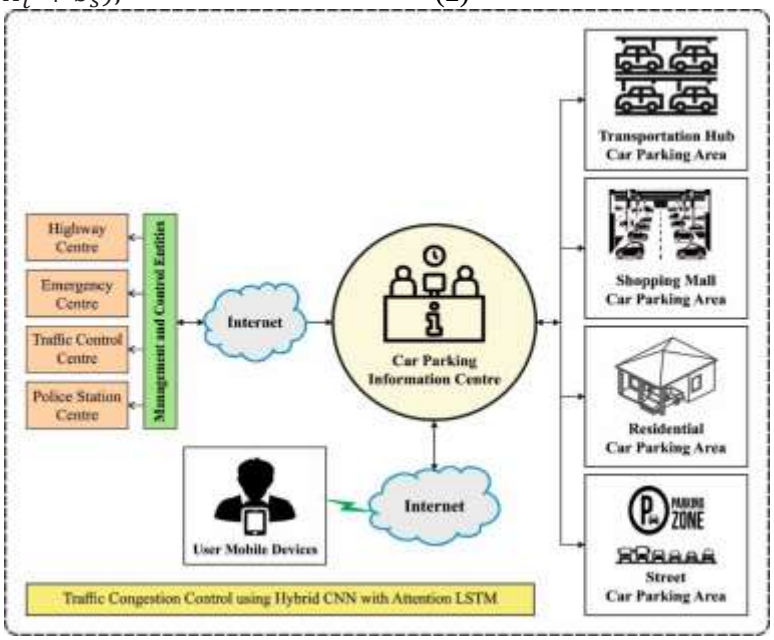


Figure 1: Overall process of IAOADL-TCC approach

In Eq. (2), W_s mentions the weight of filters, b_s refers to bias, X_t^s indicates the input data TFs at t time step, and $*$ symbol indicates the convolution function. The activation function can be represented as σ and the outcome of convolution layer is specified by Y_t^s. These processes can extract the spatial attribute in neighboring observation places. Also, the TF data illustrates temporal relationship from the neighboring time. The LSTM provides an effective technique to learn long time dependency to run sequence datasets. The LSTM architecture contains a cell for storing block cell state and 3 gates: forget, output, and input gates. Fig. 2 demonstrates the framework of LSTM technique.

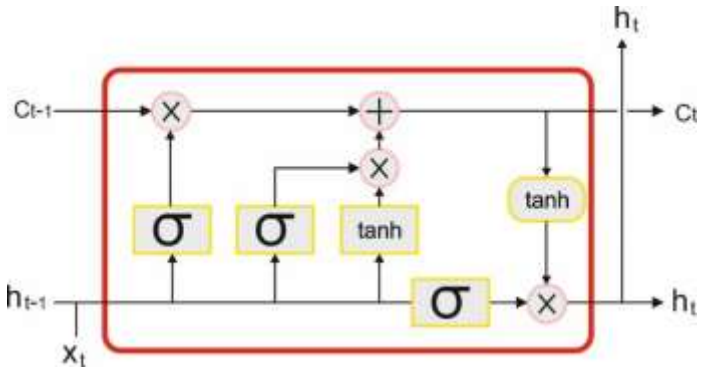


Figure 2: Framework of LSTM

For enhancing the efficiency of DNN approach, the typical method can be used to increase the layer from the method. In such cases, stacked LSTM layer has been modeled to capture better feature levels of the data TF. By using stacked LSTM, every LSTM layer gets the hidden layer of prior state as its input [18]. The LSTM layer1 process the series outcomes in the CNN component $G_t^s = [G_{t-n}^s, \dots, G_{t-1}^s, G_t^s]$ from beginning to the end and calculates the hidden layer every time step $H_{1,t}^s = [H_{1,t-n}^s, \dots, H_{1,t-1}^s, H_{1,t}^s]$. After which, the hidden state series $H_{1,t}^s$ is inputted to LSTM layer2 to calculate the hidden layer $H_{2,t}^s$ to t time step as resultant of entire LSTM network H_t^s and it is formulated by the subsequent formula:

$$i_t = \sigma\left(W_{gi} I_t^s + W_{hi} H_{t-1}^s + W_{ci} \circ C_{t-1} + b_i\right), \quad (3)$$

$$f_t = \sigma\left(W_{gf} I_t^s + W_{hf} H_{r-1}^s + W_{cf} \circ C_{t-1} + b_f\right), \quad (4)$$

$$C_t = f_t \circ C_{t-1} + i_t o \tanh\left(W_{gc} I_t^s + W_{hc} H_{t-1}^s + b_c\right), \quad (5)$$

$$o_t = sigma\left(W_{go} I_t^s + W_{ho} H_{t-1}^s + W_{co} \circ C_t + b_o\right), \quad (6)$$

$$H_t^s = o_t \circ \tanh\left(C_t\right), \quad (7)$$

Wherein I_t^s denotes input of LSTM at t time step, σ indicates the activation function, $i_t, f_t, 0_t$ denotes the output, forget, and input gates at time t, C_t represents the cell state, $\circ$ specifies the element-wise product, W, and b describes the weight and bias. The final resultant of LSTM layer H_t^s can be described by output gate and update cell. The inputs I_t^s of layer1 and layer2 were G_t^s and $H_{1,t}^s$. The resultants H_t^s of layer1 and layer2 corresponding to $H_{1,t}^s$ and $H_{2,t}^s$. Finally, it could achieve the spatiotemporal features H_t^s at t time step.

The attention mechanism has been provided to explore the inherent attributes of dataset and enhances the effectiveness of data process. The common LSTM cannot be identified that was a crucial part of a data TF. In order to address these problems, attention module has been proposed to the HCNN-ALSTM element to exploit different levels of data TF series at different times automatically. The resultant of HCNN-ALSTM at t time steps were computed as weighting sum of outcomes of the LSTM H_t^s:

$$H_t^a = \sum_{k=1}^{n+1} \beta_k H_{t-(k-1)}^s, \quad (8)$$

In Eq. (8), $n+1$ denotes the length of flow series and β_k infers the temporary attention value at time step $t-(k-1)$. The attention value β_k was evaluated by using Eq. (9):

$$\beta_k = \frac{\exp\left(s_k\right)}{\sum_{k=1}^{n+1} \exp\left(s_k\right)}. \quad (9)$$

The score $s = (s, s, \cdots, s_{n+1})^T$ represents the importance of every part of TF data framework that can be obtained by using the following expression:

$$s_t = V_s^T \tanh\left(W_{hs} G_t^s + W_{ls} H_t^s\right), \quad (10)$$

In Eq. (10), V_s, W_{xs} and W_{hs} indicates the learnable variables and H_t^s means the hidden resultant in HCNN-ALSTM system. From Eqs. (9) and (10), it can be apparent that attention value β at t time step depending on G_t^s input and H_t^s hidden parameter at t existing time step and prior n time step. Also, the attention value β is observed as activation of flow selective gates.

3.3 Hyperparameter Tuning Using IAOA

At the last stage, the IAOA based hyperparameter optimizer is derived to optimally tune the hyperparameters related to the HCNN-ALSTM model. The AOA process begins with a collection of solution candidates (X) that is produced in a random fashion, and in every iteration, the optimal solution candidates are taken into account the optimal solution or nearly the optimal [19].

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-l} & x_{1,n} \\ x_{2l} & \cdots & x_{2,\mathrm{j}} & \ldots & x_{2n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \vdots & x_{N-l} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \quad (11)$$

Eq. (12) evaluated Math Optimizer Accelerated (MOA) feature that is exploited in search phase.

$$MOA(C_{Iter}) = \text{Min} + C_{Iter} \times \left(\frac{\text{Max} - \text{Min}}{M_{Iter}}\right) \quad (12)$$

In Eq. (13), $MOA(C_{Iter})$ indicates the function value at $t\text{-}th$ iterations. C_{Iter} illustrates the existing iteration within $[1 \sim M_{Iter}]$. Max and Minindicate the maximal and minimal values of the accelerated function and it is given in the following.

$$x_{ij}(C_{Iter}+1) = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) & r_2 < 0.5 \\ best(x_j) \times MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) & Or\ else \end{cases} \quad (13)$$

Now, $x_i(C_{Iter}+1)$ illustrates the i^{th} solution in the subsequent iteration, $x_{i,j}(C_{Iter})$ defines the j^{th} position of i^{th} solution at the existing iteration, and $best\tau(x_j)$ illustrates the j^{th} position in the optimal accomplished solution. ε indicates a small integer value, UB_j and LB_j denotes the upper and lower values of j^{th} position. μ denotes control parameter for adopting the searching method that is represented as 0.5 depending on the problem that was examined.

$$\text{MOP}(C_{Iter}) = 1 - \frac{C_{Iter}^{\frac{1}{\alpha}}}{M_{Iter}^{\frac{1}{\alpha}}} \quad (14)$$

In Eq. (14), Math Optimizer probability (MOP) indicates a coefficient, MOP (C_{Iter}) shows the function value at t^{th} iterations, C_{Iter} demonstrates the present iteration and (M_{Iter}) designates the maximal iteration count. α indicates sensitive variable and describes exploitation performance on the iteration i.e., fixed corresponding to 5. The process of searching was conditioned by the MOA value for the condition of r_1 is not high when compared to the existing $MOA(C_{Iter})$ values. In AOA, the exploitation operators (Subtraction (S) and Addition (A)) of AOA explores the search region deep in dense region and employs 2 key searching mechanisms for finding a better solution.

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) - MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) r_3 < 0.5 \\ best(x_j) + MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right) Or\ esle \end{cases} \quad (15)$$

The IAOA was derived by exploitation of quasi-oppositional based learning (QOBL). Tizhoosh proposed the idea of OBL that includes opposite numbers that have maximal probability of accomplishing a solution in contrast with an arbitrary number [20]. The incorporation of OBL with AOA method results in improved convergence rate and enhanced performance. The QOBL approach effectually uses quasi-opposite numbers over the opposite number in detecting global optimal outcomes. Given that χ refers to a real number in I dimensional area. Consider the opposite and quasi opposite numbers x^o and x^{qo} (of x) is characterized by:

$$x^0 = a + b - x \quad (16)$$

Where $x \in \mathbb{R}$ and $\in b$].

$$x^{qo} = rand\left(\frac{a+b}{2}, x^0\right) \quad (17)$$

Assume $X(x_1, x_2, \ldots, x_n)$ as a point in n dimensional region. Given that the opposite point,

$X^o(x_1^o, x_2^o, \dots, x_n^0)$ and quasi opposite point, $X^{qo}(x_1^{qo}, x_2^{qo}, \dots, x_n^{qo})$ and are given in the following.

$$x_i^o = a_i + b_i - x_i \quad (18)$$

Where $x_i \in \mathbb{R}$ and $x_i \in [a_i, b_i] \forall i \in 1,2, \dots, n$.

$$x_i^{qo} = rand\left(\frac{a_i+b_i}{2}, x_i^o\right) \quad (19)$$

Algorithm 1: Pseudo-code of AOA

Input: The parameter of AOA includes number of agents (N), overall amount of iterations M_t, and dynamic exploitation parameter (α).

Generate the primary value for the agent $X_i i = 1, \dots N/\cdot$

while $(t < M_t)$ do

Calculate the fitness function for every agent.

Define the optimal agent X_b.

Upgrade the MOA and M_{OP}.

for $i = 1$ to N do

for $j = 1$ to Dim do

Upgrade the value of r_1, r_2, and r_3.

if $r_1 > MOA$ then

Exploration stage

Upgrade the X_i.

else

Exploitation stage

Upgrade the X_i.

end if

end for

end for

$$t = t + 1$$

end while

Display the optimal agent (X_b).

QOBL is employed in the AOA for generation jumping and population initialization. It generates a subset of optimum solutions for the initialized population [20]. The IAOA method derives a fitness function to accomplish better performance of the classification. It determines a positive integer to characterize the improved performance of the solution candidate. In the presented method, the reduction of classifier error rate is taken into account as the fitness function in the following. The optimum solution has the lowest error rate and the worse solution achieves a better error rate.

$$fitness(x_i) = ClassifierErrorRate(x_i)$$
$$= \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} * 100 \quad (20)$$

4 Experimental Validation

The performance validation of the IAOADL-TCC method can be tested using a road traffic

dataset from Kaggle repository [21]. The dataset includes 2282 samples with two class labels as depicted in Tab. 1.

Table 1:Dataset details

Class	No. of Samples
Positive	346
Negative	1936
Total Number of Samples	2282

Fig. 3 establishes the confusion matrices created by the IAOADL-TCC model under varying training (TR) data and testing (TS) data. With 80% of TR data, the IAOADL-TCC algorithm has recognized 242 samples under positive class and 1517 samples under negative class. Also, with 20% of TS data, the IAOADL-TCC technique has recognized 56 samples under positive class and 392 samples under negative class. Along with that, with 70% of TR data, the IAOADL-TCC approach has recognized 171 samples under positive class and 1355 samples under negative class.

Table 2 provides the overall results of the IAOADL-TCC model on 80% of TR data and 20% of TS data. Fig. 4 demonstrates the results offered by the IAOADL-TCC model on 80% of TR data. The figure implied that the IAOADL-TCC model has gained effectual outcomes under both classes. For example, the IAOADL-TCC method has identified positive samples with $accu_y$ of 96.38%, $prec_n$ of 91.67%, $reca_l$ of 84.62%, F_{score} of 88.00%, and error rate of 3.62%. Also, the IAOADL-TCC algorithm has identified negative samples with $accu_y$ of 96.38%, $prec_n$ of 97.18%, $reca_l$ of 98.57%, F_{score} of 97.87%, and error rate of 3.62%.

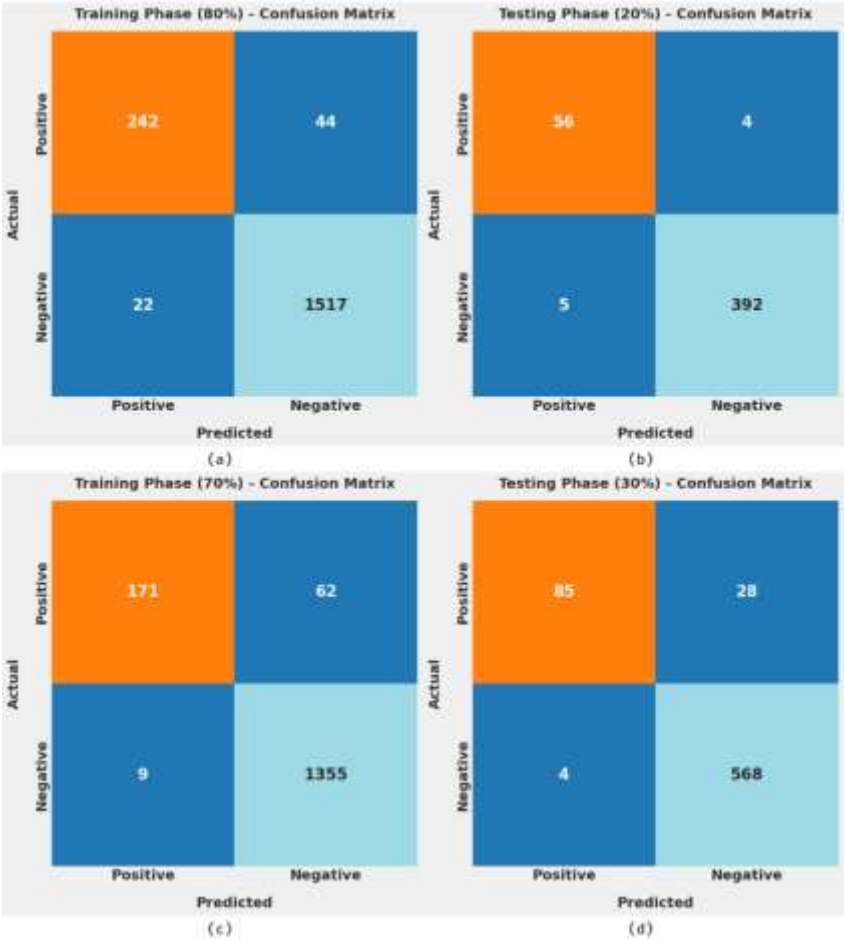


Figure 3: Confusion matrices of IAOADL-TCC approach (a) 80% of TR data, (b) 20% of TS data, (c) 70% of TR data, and (d) 30% of TS data

Table 2: Result analysis of IAOADL-TCC approach with various measures under 80:20 of TR/TS data

Labels	Accuracy	Precision	Recall	F-Score	Error Rate
Training Phase (80%)					
Positive	96.38	91.67	84.62	88.00	03.62
Negative	96.38	97.18	98.57	97.87	03.62
Average	96.38	94.42	91.59	92.94	03.62
Testing Phase (20%)					
Positive	98.03	91.80	93.33	92.56	01.97
Negative	98.03	98.99	98.74	98.87	01.97

Average	98.03	95.40	96.04	95.71	01.97

Fig. 5 establishes the results provided by the IAOADL-TCC technique on 20% of TS data. The figure implied that the IAOADL-TCC algorithm has obtained effectual outcomes under both classes. For instance, the IAOADL-TCC approach has identified positive samples with $accu_y$ of 98.03%, $prec_n$ of 91.80%, $reca_l$ of 93.33%, F_{score} of 92.56%, and error rate of 1.97%. Also, the IAOADL-TCC methodology has identified negative samples with $accu_y$ of 98.03%, $prec_n$ of 98.99%, $reca_l$ of 98.74%, F_{score} of 98.87%, and error rate of 1.97%.



Figure 4: Result analysis of IAOADL-TCC approach under 80% of TR data

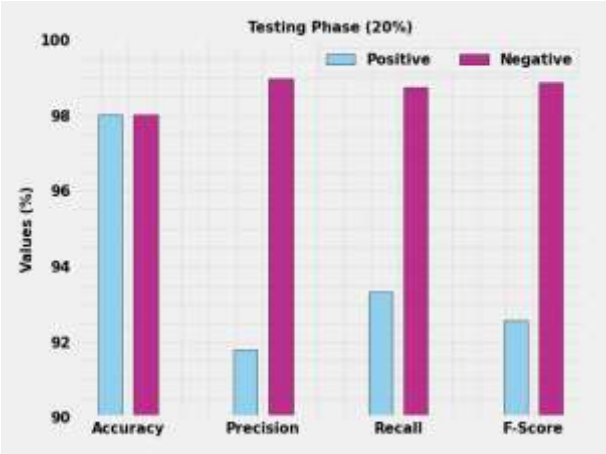


Figure 5: Result analysis of IAOADL-TCC approach under 20% of TS data

Tab. 3 offers the overall results of the IAOADL-TCC algorithm on 70% of TR data and 30% of TS data. Fig. 6 validates the results presented by the IAOADL-TCC methodology on 70% of TR data. The figure denoted the IAOADL-TCC technique has attained effectual outcome under both classes. For example, the IAOADL-TCC approach has identified positive samples with $accu_y$ of 95.55%, $prec_n$ of 95%, $reca_l$ of 73.39%, F_{score} of 82.81%, and error rate of 4.45%. In addition, the IAOADL-TCC approach has identified negative samples with $accu_y$ of 95.55%, $prec_n$ of 95.30%, $reca_l$ of 99.30%, F_{score} of 97.26%, and error rate of 4.67%.

Table 3: Result analysis of IAOADL-TCC approach with various measures under 70:30 of TR/TS data

Labels	Accurac	Precisio	Recal	F-	Error

	y	n	l	Score	Rate
Training Phase (70%)					
Positive	95.55	95.00	73.39	82.81	04.45
Negative	95.55	95.62	99.34	97.45	04.45
Average	95.55	95.31	86.37	90.13	04.45
Testing Phase (30%)					
Positive	95.33	95.51	75.22	84.16	04.67
Negative	95.33	95.30	99.30	97.26	04.67
Average	95.33	95.40	87.26	90.71	04.67



Figure 6: Result analysis of IAOADL-TCC approach under 70% of TR data

Fig. 7 establishes the results rendered by the IAOADL-TCC technique on 30% of TS data. The figure represented that the IAOADL-TCC methodology has reached effectual outcomes under both classes. For instance, the IAOADL-TCC approach has identified positive samples with $accu_y$ of 95.33%, $prec_n$ of 95.51%, $reca_l$ of 75.22%, F_{score} of 84.16%, and error rate of 4.67%. Moreover, the IAOADL-TCC algorithm has identified negative samples with $accu_y$ of 95.33%, $prec_n$ of 95.30%, $reca_l$ of 99.30%, F_{score} of 97.26%, and error rate of 4.67%.

The training accuracy (TA) and validation accuracy (VA) acquired by the IAOADL-TCC algorithm on test dataset is demonstrated in Fig. 8. The experimental outcome denoted the IAOADL-TCC method has attained maximal values of TA and VA. In Particular, the VA is greater than TA.

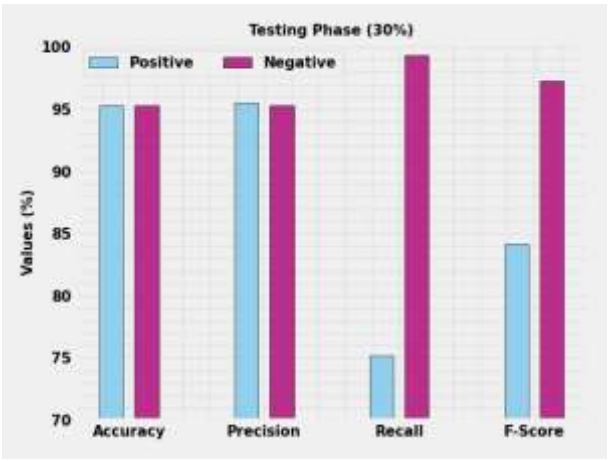


Figure 7: Result analysis of IAOADL-TCC approach under 30% of TS data

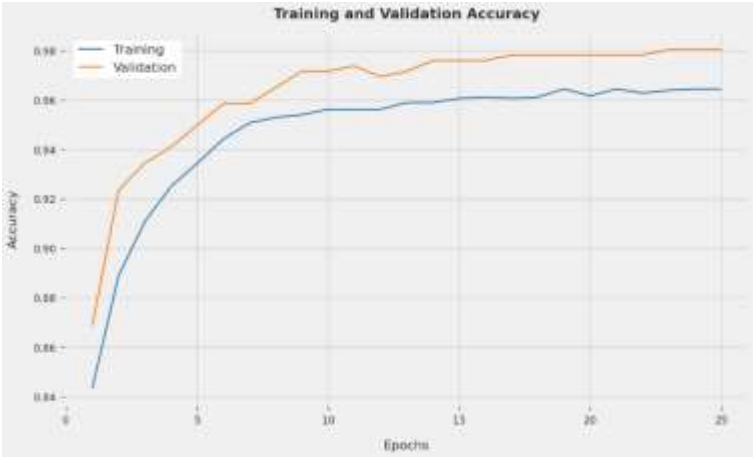


Figure 8: TA and VA analysis of IAOADL-TCC approach

The training loss (TL) and validation loss (VL) attained by the IAOADL-TCC method on test dataset are shown in Fig. 9. The experimental outcome indicated the IAOADL-TCC method has established least values of TL and VL. Particularly the VL is lesser than TL.



Figure 9: TL and VL analysis of IAOADL-TCC approach

A clear precision-recall analysis of the IAOADL-TCC technique on test dataset is portrayed in Fig. 10. The figure represented the IAOADL-TCC method has resulted to enhanced values of

precision-recall values under all classes.

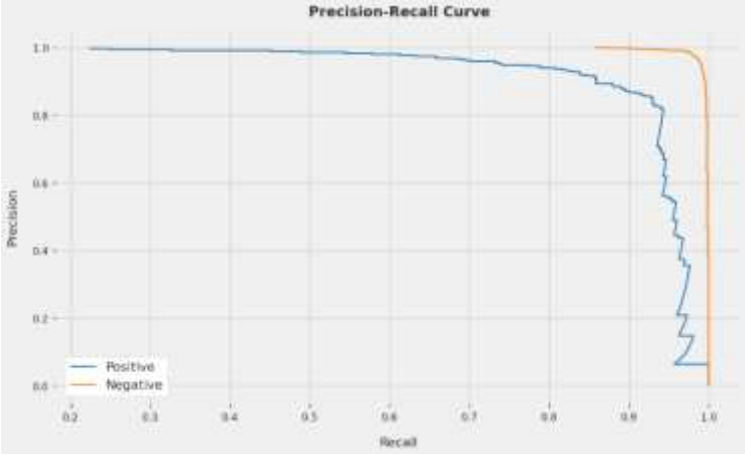


Figure 10: Precision-recall analysis of IAOADL-TCC approach

A brief ROC analysis of the IAOADL-TCC algorithm on test dataset is depicted in Fig. 11. The results signify the IAOADL-TCC approach has shown its ability in categorizing distinct classes on the test dataset.

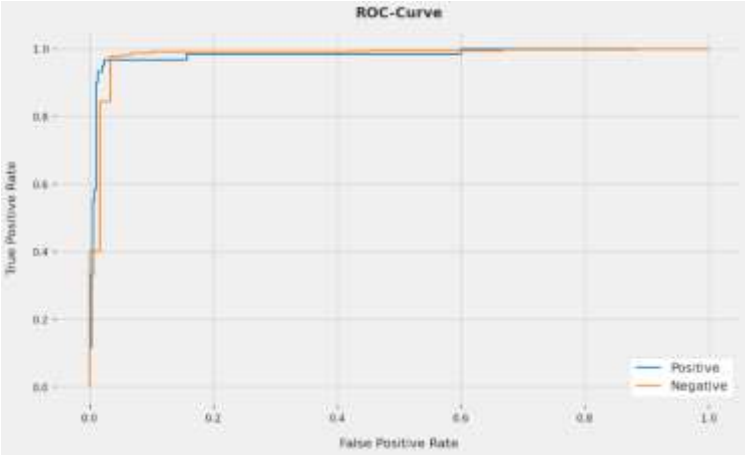


Figure 11: ROC analysis of IAOADL-TCC approach

To illustrate the better performance of the IAOADL-TCC method, a comparison study is made with recent models [8] in Tab. 4. Fig. 12 reports a comparative $accu_y$ inspection of the IAOADL-TCC model with existing models. The results pointed out that the UAV-CNN model has shown poor performance with $accu_y$ of 93.91%. Followed by, SVM, ANN, and FITCCS-VN models have reported moderately improved $accu_y$ values of 94.96%, 94.62%, and 94.90%. At the same time, the J48 and AAN-fuzzy logic models have attained reasonable performance with $accu_y$ of 95.19% and 95.08% respectively. However, the IAOADL-TCC algorithm has shown superior results with $accu_y$ of 98.03%.

Table 4: Comparative analysis of IAOADL-TCC approach with existing methodologies

Methods	Accuracy	Miss Rate	Computational Time (min)
IAOADL-TCC	98.03	1.97	1.18

J48 Algorithm	95.19	4.81	3.29
UAV-CNN Model	93.91	6.09	4.53
AAN-Fuzzy Logic	95.08	4.92	1.94
SVM Model	94.96	5.04	1.62
ANN Model	94.62	5.38	3.84
FITCCS-VN	94.90	5.10	3.32

Fig. 13 reports a comparative miss rate inspection of IAOADL-TCC technique with existing methods. The results pointed the UAV-CNN approach has displayed poor performance with miss rate of 6.09%. Then, SVM, ANN, and FITCCS-VN models have reported moderately improved miss rate values of 5.04%, 5.38%, and 5.10%. Meanwhile, the J48 and AAN-fuzzy logic approaches have gained reasonable performance with miss rates of 4.81% and 4.92% respectively. However, the IAOADL-TCC algorithm has shown superior results with miss rate of 5.10%.

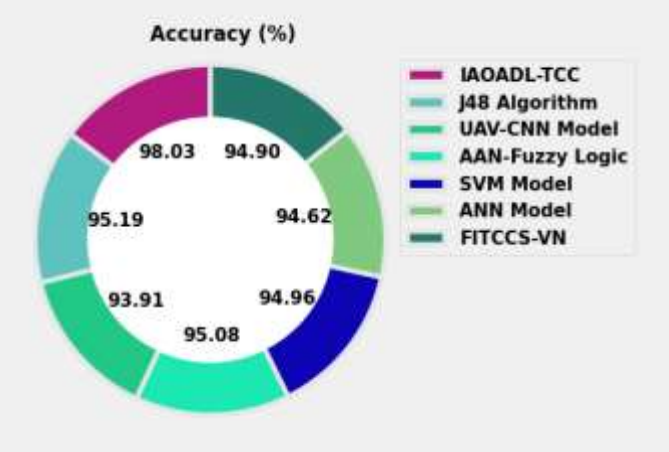


Figure 12: $Accu_y$ analysis of IAOADL-TCC approach with existing methodologies

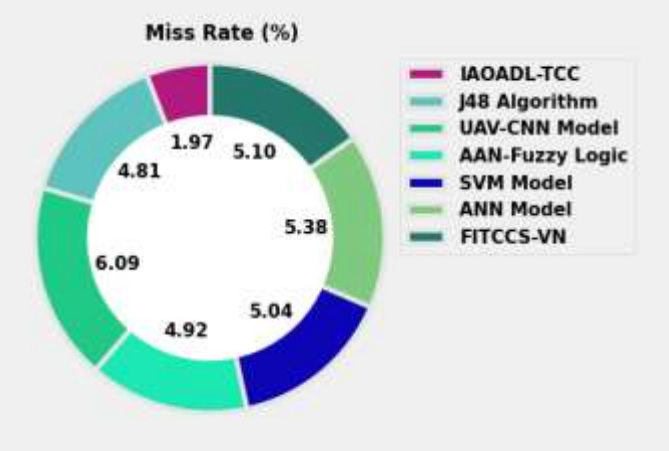


Figure 13: Miss rate analysis of IAOADL-TCC approach with existing methodologies

Fig. 14 reports a comparative CT analysis of the IAOADL-TCC method with existing models. The results signify the UAV-CNN algorithm has portrayed poor performance with CT of 4.53min. Followed by, SVM, ANN, and FITCCS-VN models have reported moderately improved CT values

of 1.62min, 3.84min, and 3.32min. Meanwhile, the J48 and AAN-fuzzy logic models have gained reasonable performance with CT of 3.29min and 1.94min correspondingly. But, the IAOADL-TCC technique has shown superior results with CT of 1.18min.

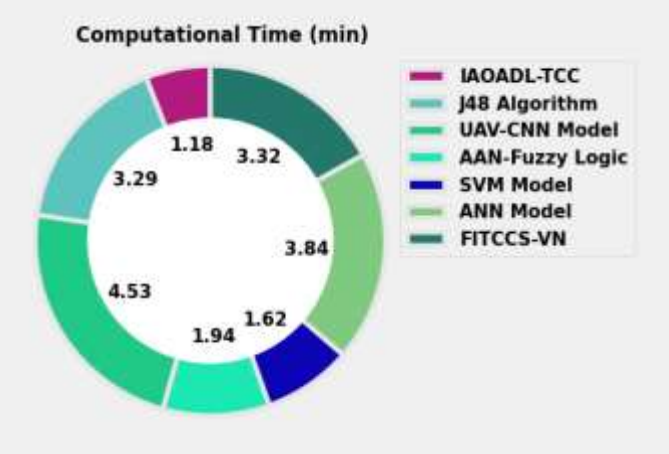


Figure 14: CT analysis of IAOADL-TCC approach with existing methodologies

From the results and discussion, it is assured that IAOADL-TCC model has accomplished improved results over other models.

5 Conclusion

In this study, a new IAOADL-TCC method was devised for effective control of traffic congestion in the ITS environment. The presented IAOADL-TCC model initially preprocessed the traffic data. In addition, the HCNN-ALSTM model is utilized to control traffic congestion. Finally, the IAOA based hyperparameter optimizer is derived to optimally tune the hyperparameters related to the HCNN-ALSTM model. The presented IAOADL-TCC model effectively enhances the flow of traffic and reduces congestion. The experimental result analysis of the IAOADL-TCC algorithm can be tested by making use of road traffic dataset from Kaggle repository. The experimental outcome stated the better performance of the IAOADL-TCC model over other DL techniques. In upcoming years, the performance of the IAOADL-TCC model was extended to the design of fusion-based DL models.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Shahid, M. Shah, A. Khan, C. Maple and G. Jeon, "Towards greener smart cities and road traffic forecasting using air pollution data," *Sustainable Cities and Society*, vol. 72, pp. 103062, 2021.

[2] B. Vijayalakshmi, K. Ramar, N. Z. Jhanjhi, S. Verma, M. Kaliappan *et al.* "An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city," *International Journal of Communication Systems*, vol. 34, no. 3, pp. 1-14, 2020.

[3] S. Khan, S. Nazir, I. G. Magariño and A. Hussain, "Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion," *Computers & Electrical Engineering*, vol. 89, pp. 106906, 2021.

[4] F. Wang, J. Xu, C. Liu, R. Zhou and P. Zhao, "On prediction of traffic flows in smart cities: a multitask deep learning based approach," *World Wide Web*, vol. 24, no. 3, pp. 805-823, 2021.

[5] L. R. Frank, Y. M. Ferreira, E. P. Julio, F. H. C. Ferreira, B. J. Dembogurski*et al.* "Multilayer perceptron and particle swarm optimization applied to traffic flow prediction on smart cities," in *International Conference on Computational Science and Its Applications*, Lecture Notes in Computer Science book series, Springer, Cham, vol. 11622, pp. 35–47, 2019.

[6] B. Alsolami, R. Mehmood and A. Albeshri, "Hybrid statistical and machine learning methods for road traffic prediction: A review and tutorial," *Smart Infrastructure and Applications*, EAI/Springer Innovations in Communication and Computing book series, Springer, Cham, pp. 115–133, 20119.

[7] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Computing and Applications*, vol. 33, no. 2, pp. 613-624, 2020.

[8] M. Saleem, S. Abbas, T. Ghazal, M. A. Khan, N. Sahawneh *et al.* "Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques," *Egyptian Informatics Journal*, 2022, doi: 10.1016/j.eij.2022.03.003.

[9] Z. Ullah, F. Al-Turjman, L. Mostarda and R. Gagliardi, "Applications of Artificial Intelligence and Machine learning in smart cities," *Computer Communications*, vol. 154, pp. 313-323, 2020.

[10] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Vehicular Communications*, vol. 20, pp. 100184, 2019.

[11] Y. Qi, M. Hossain, J. Nie and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328-337, 2021.

[12] F. Kong, J. Li, B. Jiang and H. Song, "Short-term traffic flow prediction in smart multimedia system for Internet of Vehicles based on deep belief network," *Future Generation Computer Systems*, vol. 93, pp. 460-472, 2019.

[13] C. Chen, Z. Liu, S. Wan, J. Luan and Q. Pei, "Traffic flow prediction based on deep learning in internet of vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3776-3789, 2021.

[14] W. Wei, H. Wu and H. Ma, "An autoencoder and lstm-based traffic flow prediction method," *Sensors*, vol. 19, no. 13, pp. 2946, 2019.

[15] I. Alam, D. M. Farid and R. J. Rossetti, "The prediction of traffic flow with regression analysis," in *Emerging Technologies in Data Mining and Information Security*, Advances in Intelligent Systems and Computing book series, Springer, Singapore, vol. 813, pp. 661-671, 2019.

[16] F. Zhang, T. Y. Wu, Y. Wang, R. Xiong, G. Ding*et al.* "Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[17] A. Boukerche and J. Wang, "A performance modeling and analysis of a novel vehicular traffic flow prediction system using a hybrid machine learning-based model," *Ad Hoc Networks*, vol. 106, pp. 102224, 2020.

[18] H. E. Cherng and C. H. Chang, "Short text conversation based on deep neural network and analysis on evaluation measures,"arXiv preprint arXiv:1907.03070, 2019.

[19] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, pp. 113609, 2021.

[20] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, Vienna, Austria, vol. 1, pp. 695-701, 2005.

[21] https://www.kaggle.com/arashnic/road-trafic-dataset?select=region_traffic.csv