Expert Systems With Applications 213 (2023) 119033

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Hybrid of deep recurrent network and long short term memory for rear-end collision detection in fog based internet of vehicles

Mubarak S. Almutairi [a], Khalid Almutairi [b], Haruna Chiroma [c,*]

[a] College of Computer Science and Engineering, University of Hafr Al Batin, Saudi Arabia
[b] Mechanical Engineering Technology, Applied College, University of Hafr Batin, Saudi Arabia
[c] Computer Science and Engineering Technology, Applied College, University of Hafr Al Batin, Saudi Arabia

ARTICLE INFO

Keywords:
Collision detection
Internet of Vehicles
Long short term memory
Deep Recurrent Neural Network
Deep learning algorithm

ABSTRACT

The development and use of intelligent transportation systems as an emerging trend in the application of computational intelligence within the concept of internet of vehicles (IoV) is attracting attention in the academia and industries. The computational intelligence algorithms such as the deep learning is playing a significant role in the IoV. The detection of rear end collision in the IoV is a critical aspect of the IoV because of safety. Previous research applied genetically optimized artificial neural network (ANN) (GA-ANN) for the detection of rear end collision in the IoV. However, GA-ANN has limitations such as stretching of the entire network leading to reduced flexibility, deteriorating performance as data increases to very large size, lack of generalization power resulting from over-fitting of the training data and calling for additional effort for regularization functions to monitor the model's complexity. In this research, as an alternative to GA-ANN for improving the accuracy of rear end collision detection in IoV, we propose a hybrid of Deep Recurrent Neural Network (DRNN) and Long Short Term Memory (LSTM) (DRNN-LSTM) for rear end collision detection in the IoV. The scenario for the IoV is created and simulated to generate the dataset. The propose DRNN-LSTM is applied to detect the rear end collision in IoV. The performance of the proposed DRNN-LSTM is compared with the constituent algorithms of the DRNN-LSTM, ANN and GA-ANN. The results show that the proposed DRNN-LSTM detects rear end collision in IoV better than the GA-ANN, LSTM, DRNN and ANN. Thus, our propose approach has potential to improve safety, and effectiveness of the overall vehicle mobility in the IoV environment. Keywords: Collision Detection; Internet of Vehicles; Long Short Term Memory; Deep Recurrent Neural Network; Deep Learning Algorithm.

1. Introduction

As the Internet of Things (IoT) (Ok et al., 2021) brings about intelligent health, home, energy and transportation systems, internet of vehicle (IoV) brings about intelligent vehicles (Feki et al., 2013). In the IoV environment (Labriji et al., 2021), vehicles are allowed to communicate with one another, other devices such as the mobile phones and so on, via wireless access technology with the help of road side units (RSUs) (Kaiwartya et al., 2016). Fig. 4 depicts a typical IoV environment showing the buildings, infrastructural facilities, RSUs, the vehicles and other components found in a typical IoV environment.

Due to varying speed, vehicular conditions, braking system state and poor environmental conditions, vehicles on the road may encounter challenges leading to accidents, which usually takes place on high ways during high speed vehicles mobility in the IoV environment (Wang et al., 2014). A typical scenario is the rear-end collision that amounts to 60 %-70 % of total accidents which in turn affects the overall safety of lives, and properties (Kim et al., 1999).When we talk about a rear-end collision, we simply consider it as a possibility of a moving vehicle from behind coming to hit the one before it.

Rear end collision has been proven to cause a huge traffic congestion and danger to the safety of the vehicles and the commuters, some of which includes damage of vehicles, loss of properties, injury, and possible death. This usually affects young people of age from 15 to 29 years (Nkenyereye et al., 2019). About 90 % of accidents have been found to be caused by absence of reliable traffic management system, poor road safety infrastructures (Nkenyereye et al., 2019) and over speeding (Zhang et al., 2014; WHO 2015), developing a rear end collision detection mechanism will go a long way in enhancing the overall safety of the vehicles, commuters, and thereby saving us time, energy

* Corresponding author.
E-mail address: chiromaharun@fcetgombe.edu.ng (H. Chiroma).

https://doi.org/10.1016/j.eswa.2022.119033
Received 24 September 2021; Received in revised form 5 October 2022; Accepted 11 October 2022
Available online 5 November 2022

and resources in view of the fact that in the IoV about 60 % to 70 % of total accidents are due to rear end collision (Kim et al., 1999).

Lack of sufficient time for the drivers to react to dangers is the main reason for rear end collision because of the inaccurate timing to apply the brake. It is found that 60 % of the collision can be avoided if the driver can have warning of 0.5 s ahead (Board, 2001). Similarly, 90 % of the collisions can be avoided if the driver can have 1.5 s warning ahead, as such, rear end collision detection gained tremendous attention from the research community in the area of IoV (Wang et al., 2020). As a result, researchers proposes different approaches for the detection of collision in the IoV.

Literature review indicated that the research community heavily relied on single algorithm for the detection of collision in the IoV. However, hybrid algorithm has been proven to perform better than the single intelligent algorithms. In real life situation, there is possibility of bad weather that can result to poor visualization but previous studies mainly ignore bad weather in developing collision detection warning mechanism for deployment in the IoV. Many of the previous studies overlook to evaluate the performance of propose intelligent algorithm by comparing it with other classical algorithms to show the effectiveness of the propose algorithm, therefore, it is hard to measure the effectiveness of the propose algorithm over the classical algorithms in the detection of collision in IoV.

To solve the limitations pointed out from the previous studies, we consider hybrid of deep recurrent neural network (RNN) and long short term memory (LSTM). The RNN is good in handling time series problems (Joshi, 2019; Rebala et al., 2019a,b). However, the RNN has the limitation of exploding gradient and vanishing gradient. As a result, the LSTM is meant to handle the limitations of the RNN such as failure to analyze series of data with different trends and seasonality as a result of exploding gradient, vanishing gradient as well as oscillating weights (Hochreiter and Schmidhuber, 1997). Learning rate of RNN tends to approach a certain limitation when an amount of data samples were utilized such that any update proceeding that point were very small and not actually improving the workability of the network. Hence, the weights of the network kept oscillating and fail to converge. The LSTM can learn new trends by preservation, prior knowledge of the states is not needed in making the model to learn from default values and no need for fine tuning hyper parameters for the model enhancement (Joshi, 2019).

To improve accuracy for the detection of rear end collision in IoV, robustness of the mechanism for the detection of collision in bad weather condition and evaluate the performance of the proposal by comparing the proposal with classical algorithms, this paper propose the hybrid of Deep RNN (DRNN) and LSTM (DRNN-LSTM) for the rear-end collision detection in IoV in fog weather condition.

The summary of the paper contributions are as follows:

- The study proposes hybridized DRNN-LSTM for the detection of rea end collision in IoV environment.
- The propose hybrid DRNN-LSTM was found to perform better than the constituent algorithms and the algorithms already discussed in the literature for the detection of rear end collision in IoV.
- The IoV architecture was designed based on fog to accommodate different weather condition that improve the rea end collision detection accuracy in fog condition.
- As the project is implemented on Google collab., it is found that browser has influence on the performance of the DRNN-LSTM in the detection of rea end collision in IoV environment.

The remaining sections of the paper are organized as follows: Section 2 reviewed related works. Section 3 presents the basic operations of DRNN and LSTM while IoV foundation, architecture and motivation is presented in Section 4. Section 5 presents the proposed DRNN and LSTM hybridization framework. Section 6 presents description of data collection. Section 7 present results and discussion before making remarks about conclusions in Section 8.

2. Review of related works in internet of vehicle collision detection

This section present the works related to the detection of collision in IoV to show state-of-the-art information regarding the subject. Many works exist on collision detection as published in the literature such as the collision warning (Baek et al., 2020), collusion detection in cooperative inland vessel (Hammedi et al., 2022), human –vehicle collision detection (Qu et al., 2020), etc. However, the focus of this research is on collision detection within IoV environment. Therefore, the related works presented in this section focuses on the collision detection in IoV. One of the key services in the IoV environment is the detection of collision among vehicles. To guarantee low-latency communication and real-time reactions for avoiding vehicles collision, the services is typically deployed at the edge of the multi-access edge computing (Brik & Ksentini, 2021). The next paragraphs present the studies that applied machine learning for detecting collision in IoV.

Chang et al. (2021) developed IoV based collision detection framework using YOLOv4 for the prediction of vehicle next position to avoid collision. The framework incorporated dynamics of vehicles and machine learning procedure. The YOLOv4 is used for the prediction of the next positions of the vehicle. It is found that the propose method predict future position of vehicles more accurate and stable. However, the study utilizes single algorithm without hybridization as studies shows that hybrid algorithms perform better than the constituent algorithms as evident in (Chiroma et al., 2020). In addition, the study only compared the results with the already published work but didn't run other algorithms on the same dataset to measure the performance over other classical algorithms. To predict vehicle density and computing resources needed to detect collision, deep learning framework is proposed. The application is deployed at the mobile edge computing to detect collision in the IoV. The proposal was evaluated with dataset from mobility in a big city. The result indicated that it is possible to predict the required resources and computing resources required for each of the collision detection application (Brik & Ksentini, 2021). However, in reality, harsh/bad weather plays a role in vehicles movement in the IoV, as such, the study is limited because bad weather such as fog that is known for posing difficulty in driving is not considered in the study. So deploying such services in the real world can face difficulties for vehicular movement in the event of bad weather like fog weather condition or any other bad weather condition.

Chang et al. (2019) proposes IoV based squeeze-and-excitation networks and densely connected convolutional networks system for the detection of vehicle collision. The deep learning model is trained for the detection of traffic collision. The model is deployed on the cloud server for training serving as the cloud based management platform. Similarly, Chen et al. (2018) proposes a deep learning framework based on IoV comprising of collision detection sensors, server on the cloud for training the deep learning and service platform on the web. The deep learning is applied to detect traffic accident collision. Experimental result indicated that the deep learning framework is able to detect the traffic accident collision with high accuracy. However, the propose frameworks in (Chang et al., 2019; Chen et al., 2018) were not evaluated by comparing the proposal with other competitive algorithms to show the effectiveness of the propose frameworks. Therefore, it is very difficult to measure the advantage of the propose framework over the classical algorithms.

Wang et al. (2020) proposes rear-end collision detection system based on deep learning model – convolutional neural network. The imbalance dataset issue in the study was alleviated by smoothen and the genetic theory expansion. The convolutional neural network was applied to detect rear-end collision in the IoV. It was found that the convolutional neural network performs better than the conventional algorithms - Honda, Berkeley and MCWA in the detection of rear-end collision. However, the study utilizes single algorithm without

hybridization as studies show that hybrid algorithms perform better than the constituent algorithms as evident in (Chiroma et al., 2020). In addition, the study compared the effectiveness of the convolutional neural network with only conventional algorithms: Honda, Berkeley and MCWA without comparing it with corresponding family of the deep learning intelligent algorithms. The work in (Chen, Xiang et al., 2018) applied genetic algorithm (GA) to train artificial neural network (ANN) (GA-ANN) for the detection of rear end collision in the IoV. However, the study is accompanied with number of challenges. The GA-ANN performance deteriorates as data increases to very large size. According to (Fong et al., 2018); some scholars are of the view that the use of meta heuristics is not necessarily required for optimization of an ANN to avoid local minima. This is because local optima come from the little and gradual changes resulting from the different interactions of nodes and weights at the hidden layer. Hence, getting reliable local optima through direct error minimization proves to be very good. Also, putting more efforts to look for a global optimum can further stretch the entire network thereby resulting into reduced flexibility, and likely results in over-fitting of the training data. An ANN (Liu, 2008) that is over-fitted may be accompanied by absence of generalization power as against an ANN trained through gradient descent with a good local minima to a certain extent (Fong et al., 2018).

3. The deep learning algorithm

The deep learning architectures adopted for hybridization in the study are briefly discussed in this section. The basic operation of the algorithms is explained to show how each of the algorithms operate to achieve its goal.

3.1. Deep recurrent neural network

The RNN is a variant of neural networks (Lin et al., 1998; Maeda and Wakamura, 2005) that is good in dealing with sequential data processing. Series or sequential data in this case, can be viewed as more than one problem dealing with sequences or series of data with a loop property (Rebala et al., 2019a,b). As it appears from the word recurrent, it means that the RNN network/structure is arranged or organized in an iterative manner. The operation of the RNN is described as follows:

A typical RNN has the following features as depicted on Fig. 1. U and W are weights, S is a state, V represents vector, X represents input vector while O serves as the output vector. Computing the RNN State (S) at time (t) is usually performed in the following ways:

$$S_t = f(W_{t-1}S_{t-1} + b_{t-2}) \quad (1)$$

where:

$W_{t-1}$is the weight vector between layers t-1 andt, s_{t-1} is the state of the layer t-1, tis the index of the hidden layer, fis the activation function, usually sigmoid, $b_{t-2}$is the bias term of the previous layer. Output of the network can be obtained by:

$$O_{t+1} = f(S_t W_t + b_{t-1}) \quad (2)$$

where Ot-1 is the output of the network, fis the activation function, usually sigmoid, stis the state of the last layer, wtis the weight vector from the previous layer and bt-1 is the bias term of the last layer. The DRNN (Ridge et al., 2020) goes beyond a single hidden layer the RNN. A typical DRNN with N hidden layers, where $N > 2$ is made up of hidden states where each is continuously passed to both the next time step of the current layer and the current time step of the subsequent layer.

3.2. Long short term memory

The LSTM (Shu et al., 2020) is meant to solve the challenges faced by the RNNs with the help of gates that manages the flow of sequences at the current state and output of the current sequence.

The structure of the LSTM (Schmidhuber et al., 2002) as shown in Fig. 2, seems complex with numerous operations in the form of gates. Let us consider how signals flows in the structure, at time t-1, x is the input to the memory cell, h is the output received from previous block, C is the memory received from previous block while at time t, h is the output of the current block, C is the memory from the current block. The sigma sign starting from the input X represents the three gates, namely, forget, input and output respectively. The forget gate, previous output is merged to form values between 0 and 1 which clearly indicates how much of the state will be neglected or accepted. The output and the previous state get multiplied. The input gate assist in determining the kind of new information that the LSTM state should have, input gate generates same output as a fraction between 0 and 1 which is then multiplied with the output of the tanh block that provides new result to be summed up with that of the previous state. The vector obtained adds up together with the previous state to come up with the current state. Output gate goes this way, the previous state and the input are gated together to obtain fractional values between 0 and 1, the result is then produced. The state and the output are resent into the LSTM cell/block (Rebala et al., 2019a,b).Therefore, LSTM has the following advantages

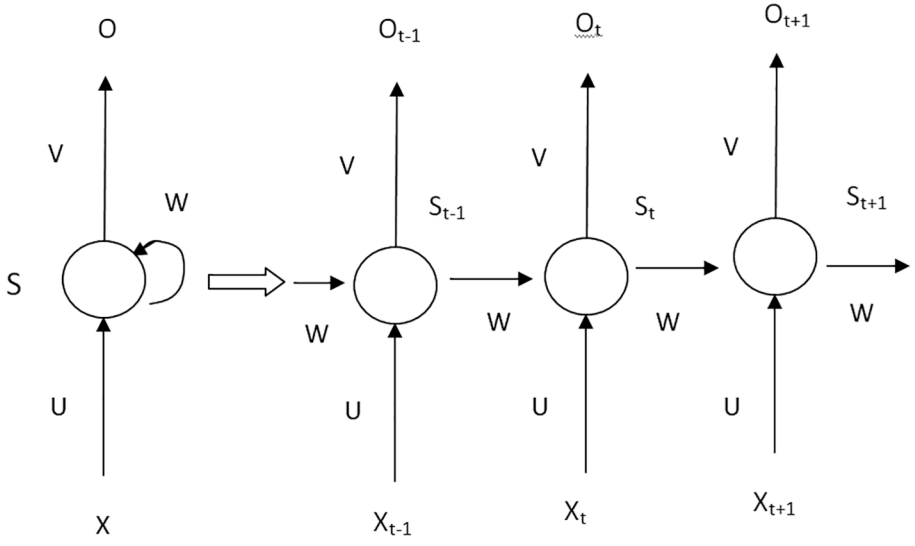

Fig. 1. Typical Recurrent Neural Network Structure.

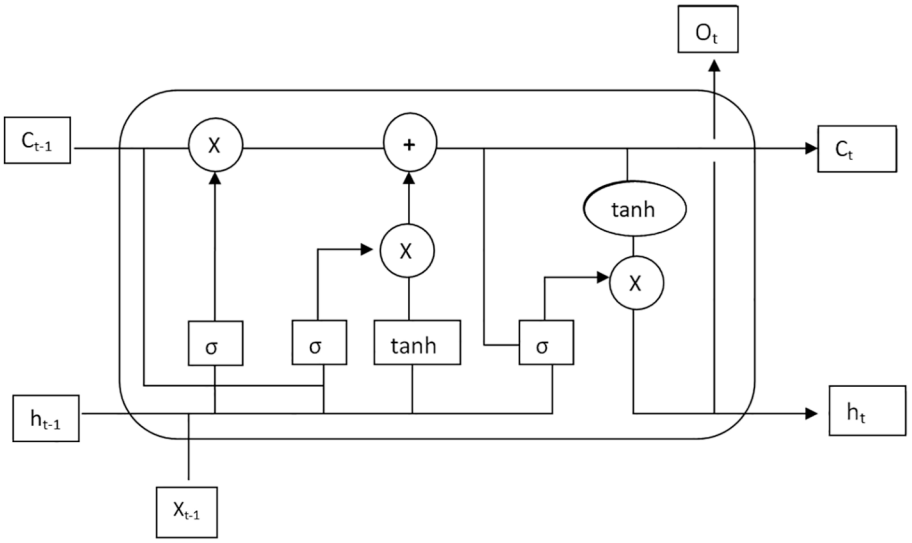

Fig. 2. Long Short Term Memory Structure.

over RNN:

i. The presence of the forget gate at the initial point of the LSTM cell assist in handling the vanishing gradient problem of the RNN.
ii. The LSTM could learn new changes/behavior of data.
iii. Early information or details about a state is not necessarily required which makes the LSTM to start learning from any default value.
iv. When compared to other deep learning algorithms, the LSTM is found to have minimum number of hyper-parameters needed for fine tuning (Rebala et al., 2019a,b).

We will use Fig. 3 to explain the Equations of the LSTM according to (Rebala et al., 2019a,b):

The sequence output (y't + 1) is denoted by:

$$Y'_{(t+1)} = f\left(W_y a^{t+1} + b_y\right) \tag{3}$$

Equation (3) implies the normal NN procedure where weights and preceding activation input defines the activation function:

$$a^{t+1} = OG\tanh\left(C^{t+1}\right) \tag{4}$$

$$C^{t+1} = FGC^{(t+1)} + IGC^{t} \tag{5}$$

$$C^{(T+1)} = \tan_h\left(W_c a^{t} + W_x X^{t-1} + b_c\right) \tag{6}$$

Eqs. (4), (5) and (6) show how changes in the activation functions take place depending on the gates.

$$FG^{t+1} = f\left(W_u a^{t} + W_x X^{t+1} + b_c\right) \tag{7}$$

$$IG^{t+1} = f\left(W_i a^{t} + W_x X^{t+1} + b_c\right) \tag{8}$$

$$OG^{t+1} = f\left(W_o a^{t} + W_x X^{t+1} + b_c\right) \tag{9}$$

where:

A_{t+1}gives the calculated activation input from sequence t + 1,x^{t-1} gives the input at sequence t + 1.

Y'^{t+1}gives the output at sequence t + 1, Ct + 1 gives the cell output at sequence t + 1, C't + 1 gives the most recent cell output at sequence t + 1, W_X gives the input vector's weight, $W_Y$gives the output vector's weight, W_C gives the activation input's weight, W_U gives weight for calculating the forget gate, W_I gives weight for calculating the input gate, $W_O$gives the weight for calculating the output gate, Fgives the activation functions, tanh for input while sigmoid for output.

4. Internet of vehicles: motivation, communications, layers, and architecture

As the main purpose of the IoV is to allow on-road vehicles to communicate for traffic enhancement, different forms of interaction in the IoV environment includes Vehicle-to-Sensor (V2S), communication from vehicle to another vehicle (V2V), interaction between vehicle and personal device (V2D), interaction between vehicle and pedestrian (V2P), communication between vehicle and RSUs (V2RSUs), and interaction between vehicle and road infrastructure (V2I) (Yang et al., 2017).

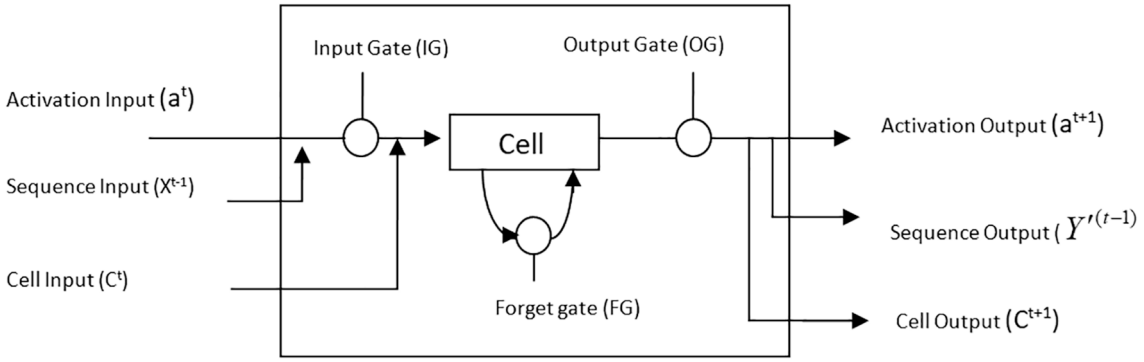

Fig. 3. Long Short Term Memory cell and Mode of Operation.

The V2S communication allows the use of sensors mounted on the vehicles to detect signals for appropriate action. TheV2V gives room for the on-road vehicles to interact and share information directly without the use of any third party. TheV2D allows for communication between vehicles and intelligent devices such as smart phones with drivers/passengers or a customized device that is mounted on the dashboard for vehicle to driver communication within the IoV environment. TheV2P is a form of communication where the moving vehicle can detect and send a warning message to the pedestrian moving along the road for the appropriate response or action. TheV2RSUs is a communication between vehicle and wireless access points situated along the roads side for information sharing and sometimes the regular moving buses can be used as RSUs. TheV2I communication provides a room for communication between vehicles and road infrastructure such as filling stations, packing space, stadium and so on. Fig. 4 shows a typical IoV architecture with its components. The entire IoV environment can be viewed as a five layered architecture: perception, coordination, AI, application and business layers (Kaiwartya et al., 2016).

The perception layer is the layer that is considered to be close to the vehicles, it is a point where the use of sensors, actuators, RSUs, intelligent mobile phones as well as other personal devices can be ensured. At the coordination layer, details of vehicle's particulars, velocity, location, and direction of movements, acceleration, engine status, and density level of road, status of weather, multimedia details and infotainments information regarding individuals are gathered. It then channels the intercepted information to the coordination layer. The basic challenge of this layer is the gathering and variation of the intercepted data in such a way that energy and cost can be saved (Kaiwartya et al., 2016).

The next layer is the coordination layer which combines different (heterogeneous) networks that involves Wireless Access for Vehicular Environment, Wireless Fidelity, 4G/long term evolution as well as satellite networks so that the data that has been captured from the perception layer is then passed to the AI layer. The coordination layer is responsible for making transfer of data/information among the various heterogeneous networks easily and accessible (Kaiwartya et al., 2016).

The AI layer serves as the brain of the IoV, it is the layer where all decision making processes are done which includes storage, analysis as well as processing of information obtained from the preceding layer, AI layer is made up of lower sub-layer consisting of service protocols, Vehicular cloud computing and big data analytics are available at the upper sub-layer consisting of cloud service storage as a service, infrastructure as a service, network as a service, cooperation as a service, entertainment as a service, gateway as a service, picture as a service, and computing as a service. The application layer is made up of two different applications namely, smart safety and efficiency, and smart business oriented. Application layer is the next after the AI layer; this layer includes the basic application services that have to do with smart applications, programs that supports web-based services, infotainment services and so on. Subsequent layer is the business layer; this layer involves a range of activities that can define the future of the IoV through the use of statistical tools upon generated data in the form of what is called business models. This is usually done with the help of tools like flowcharts, graphs, comparison table, use case diagram, etc. There is what is called a protocol stack besides the five layers. The role of the PS is to assist in completing the responsibility of each of these layers, and it is made up of security, operation and management planes. The security plane consists of IEEE 1609.2, security information connector, security management information base and hardware security module. The operation plane ensures the availability of transmission services for data/information sharing. The management plane serves as the centre of control. The entire IoV environment can be viewed as a network model with three basic entities: cloud, connection and clients (Kaiwartya et al., 2016).

5. Dataset collection and internet of vehicle architectural environment

In this section, we discussed the procedure the architecture of the IoV was designed and the process for datasets generation and collection.

Fig. 4. Internet of vehicle environment.

5.1. Dataset collection

In this section, we have provided the IoV architectural environment, method of data collection, creation of the IoV scenario and generating of the dataset. The entire IoV environment for the vehicle mobility model is a simulation of terminus round-about as the case study with little modifications to meet the simulation requirements. The IoV environment covers dimension of 1,500 m by 1,500 m and is made up of single lane road for bends, multiple (triple) one way lane road, required speed distribution, reduced speed area, signal heads for signal control, a 2D and 3D view of the IoV environment with minimum speed of 50 km/h, average speed of 95 km/h, maximum speed of 140 km/h and an approximate IoV nodes (vehicles) of 30 to 180. Table 1 presents the possible collision influencing features.

Based on the correlation of the features determined in (Chen et al., 2018) using data processing software (Tang and Zhang, 2013), the following features have been found to be correlated:

i. Number of Lanes
ii. Status of the driver
iii. Nature of the environment
iv. Velocity of Vehicle
v. Distance between vehicles
vi. Braking capability

5.2. Scenario for generating dataset

Traffic control mechanism is put in place for the traffic management system to enable collision free movement and mimic the real life situation. Fig. 5 depicts the traffic control program for the traffic management whereas Fig. 6 shows the actual IoV architectural environment of the simulation with some vehicles in mobility and the traffic signal heads for the signal control displaying the red, green, and amber colours.

The collision influencing features as described in Table 2 with respect to status, contain variable/index values that will be returned and served as input to the DRNN-LSTM model. Table 3 provides the express road parameter setting for the simulation of the IoV environment. The datasets were collected in notepad with an extension (.FZP) and pre-processed using Microsoft Excel.

5.3. Description of dataset attributes

The list of datasets attributes have been provided, the gross/net following distances, interaction state, delay time, drive state, lane and speed (PTV-Group 2019), these attributes are specifically the attributes required for our work.

i. FOLLOWDISTGR: With a short form of FollowDistGr, following distance (gross) is the distance to the front of the interaction vehicle before the time step, meter (m) is the unit of measurement.

Table 1
Collision influencing features.

Feature	Classes
Human feature	Status of the driver
	Gender
	Age
Nature of the Environment	Weather condition
	Level of traffic
	Light situation
	Nature of the road surface
Rest of the on-road vehicles	Velocity
	Distance between vehicles
The Vehicle itself	Braking capability
	Sliding property
Road	Number of lanes

ii. FOLLOWDISTNET: It has a short form of FollowDistNet, Following distance (net) is the distance to the back of the interaction vehicle before the time step and it has meter (m) as a unit of measurement.
iii. INTERACTSTATE: With a short form of InteractState, Interaction state is the state at which the vehicle is at normal or abnormal state.
iv. DELAYTM: It is written as DelayTm in its' short form, Delay time is the difference between optimal (ideal, theoretical) driving time, it is measured in (s).
v. DRIVSTATE: It is written as DrivState in a short form, it is therefore a state at which a break is successfully applied or not.
vi. LANE: It refers to the number of lanes in which the vehicle interacts; the lane can be either single or multiple.
vii. SPEED: Is the rate at which the vehicle moved at the end of the time step. It is measured in kilometer per hour (km/h).

Each of the dataset attribute values collected is as shown in Table 4.

5.4. Pre-processing of the dataset for the DRNN-LSTM model's input

The Microsoft Excel was used to pre-process the dataset for the following reasons:

i. It is user friendly.
ii. It supports the use of formulae (calculation) and dragging mechanism.
iii. It is easier to monitor the behavior of the regenerated data output.
iv. It saves us from further coding (additional source code) in the development of our models.
v. It reduces running time for the models.

Table 4 was used to translate the attributes of datasets into the corresponding index values for the six features for collision detection and prevention.

1. Number of lanes: Considering the lane on which the vehicle interacted, each, and every link (road) on the simulation has a number. A multiple lane road has a numbering order of 1 to 12 since there were 12 multiple lane roads on our simulation, while a single lane road has a numbering of 10,001 and above depending on the number of single lane road used. For instance, on the Lane column, any movements of vehicle on a lane of the form 6–1, 1–3, 11–3 and so on, is considered a multiple lane. Hence, 2 is returned as an index value, whereas 10007–1, 10005–3, 10004–2, 10006–5 and so on, considered a single lane, as such, 1 is returned as an index value for the model's use.

The following expression prints the required computation:

$$f(x) = \begin{cases} 1, & if\ x = \ single\ lane \\ 2, & otherwise \end{cases} \tag{10}$$

Velocity of vehicle: The velocity of vehicle at 50 km/h, 90 km/h and 140 km/h are considered as minimum (low), normal and maximum (over speeding). As low speed returns 0.35 as index value for the model's to use, normal and over speeding returns 0.65 and 1, respectively.

$$f(x) = \begin{cases} 0.35\ if\ x \leqslant 50 \\ 0.65\ if\ x \leqslant 90 \\ 1\ otherwise \end{cases} \tag{11}$$

2. Breaking capability: The braking system of the vehicle is considered as either available or not available, index value of 1 is returned for available while not available returns 0 as index value. Based on the attributes of data collected from the VISSIM, breaking can be described in the following instances:

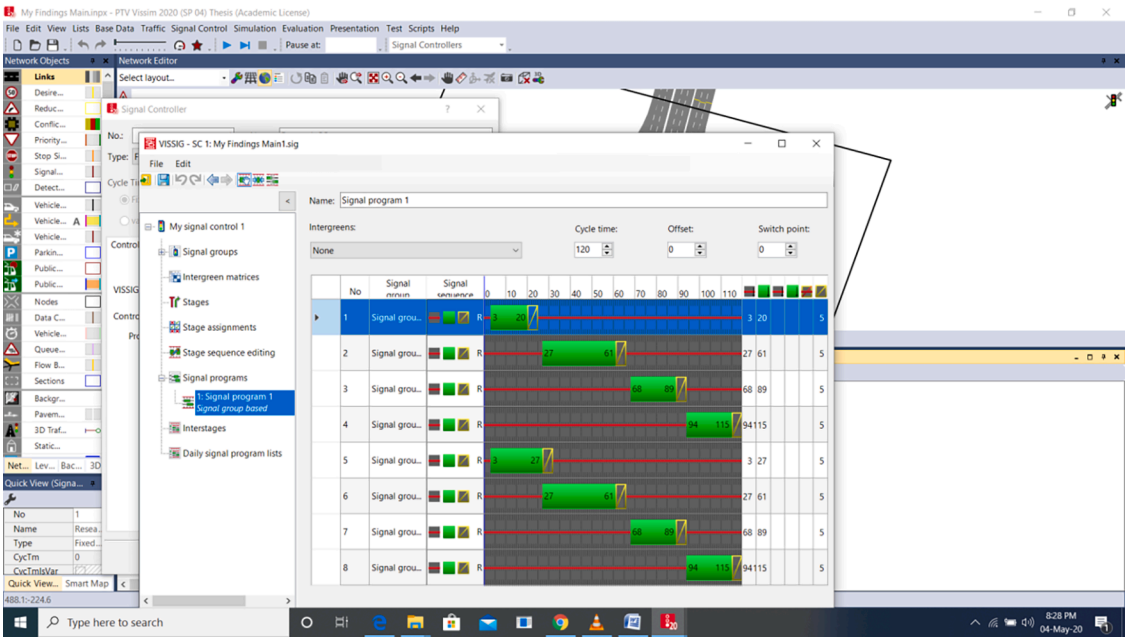

Fig 5. Traffic management control system for vehicles mobility.

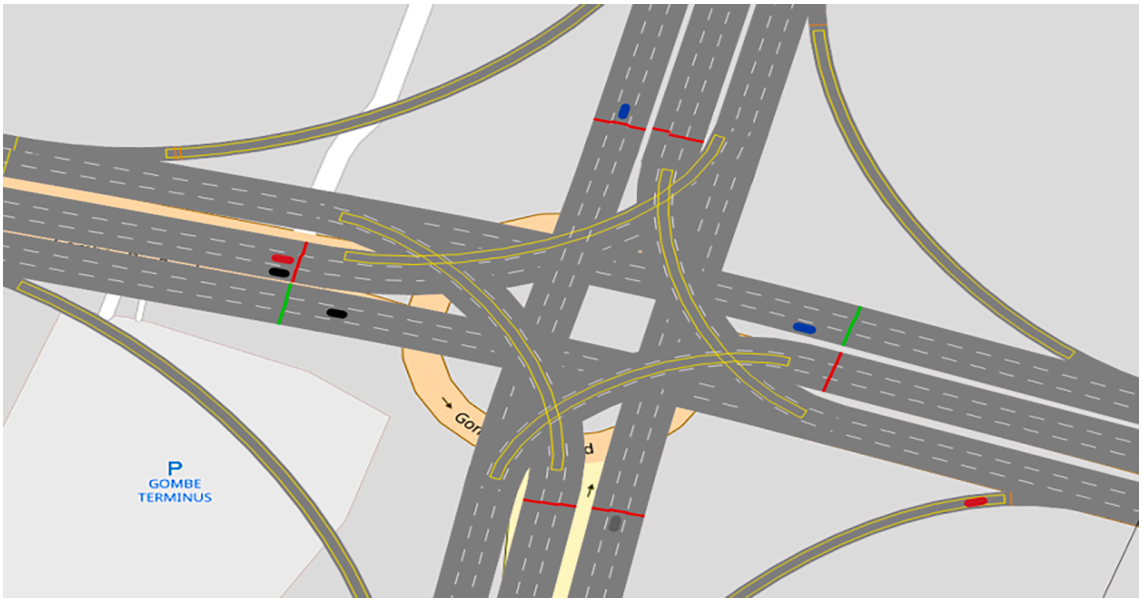

Fig. 6. Architectural designed scenario of the IoV environment.

Table 2
Quantization of collision influencing factors.

Feature	Feature status	Variable/Index value
Number of Lanes	Single, Multiple Lanes	1 is single, 2 is multiple
Status of driver	Normal, Abnormal	1 is normal, 2 is abnormal
Nature of the environment	Normal, Abnormal	1 is normal, 2 is abnormal
Velocity of Vehicle	Low, normal, over speeding	1 is speeding, 0.65 is normal, 0.35 is low
Distance between vehicles	Less than 5 m, less than 10 m, less than 20 *m*	1 for less than 5 m, 0.5 for less than 10 m, 0 for less than 20 *m*
Braking capability	Available or Not available	1 for available, 0 for not available

Table 3
Express road parameter setting.

Parameters	Values
Size of simulation area	1500 m X 1500 *m*
Type of Lane	One-way three-lanes
Minimum speed (m/s)	50 km/h
Normal speed (m/s)	90 km/h
Maximum speed (m/s)	140 km/h
Vehicle mobility model	Freeway mobility
Number of nodes (vehicles)	30 ~ 180

Table 4
Samples of collected data attribute values.

S/N	Driving state	Following distance (gross)	Following distance (net)	Interaction state	Delay time	Lane	Speed
1	Default	31.480	31.480	Close up	0.35	10007–1	22.92
2	Default	47.030	42.820	Close up	1.77	10007–1	21.72
3	Default	191.79	187.14	Close up	0.00	6–1	54.55
4	Default	218.55	218.05	Close up	0.00	1–3	4.160
5	Brakes hard	32.440	31.940	Close up	2.40	11–3	37.13
6	Default	25.110	25.110	Close up	0.35	10007–1	22.92
7	Default	41.000	36.780	Close up	1.77	10007–1	21.72
8	Default	177.52	172.88	Close up	0.00	6–1	60.48
9	Default	217.39	216.89	Close up	0.02	1–3	14.48
10	Brakes hard	20.460	19.960	Close up	3.05	11–3	25.73
11	Default	18.740	18.740	Close up	0.35	10007–1	22.92
12	Default	34.960	30.750	Close up	1.77	10007–1	21.72
13	Brakes moderately	61.030	61.030	Brake ZX	0.10	6–1	53.37
14	Default	214.80	214.30	Close up	0.05	1–3	23.51
15	Brakes moderately	11.730	11.230	Close up	3.83	11–3	15.35
16	Default	12.380	12.380	Close up	0.35	10007–1	22.92
17	Default	28.930	24.720	Close up	1.77	10007–1	21.72
18	Brakes moderately	45.220	45.220	Brake ZX	0.36	6–1	46.25
19	Default	209.53	209.03	Close up	0.10	1–3	31.31
20	Brakes moderately	6.0300	5.5300	Close up	4.70	11–3	6.900
21	Brakes hard	6.0100	6.0100	Close up	0.35	10007–1	4.400
22	Default	22.900	18.690	Close up	1.77	10007–1	21.72
23	Brakes moderately	31.380	31.380	Brake ZX	0.76	10007–1	39.14
24	Default	201.91	201.41	Close up	0.15	1–3	38.25
25	Brakes moderately	2.9300	2.4300	Close up	5.65	11–3	2.070

i. Default: in this situation, the vehicle is considered, to be moving with zero break application, hence, index value of 0 is returned.
ii. Break hard: There is a total application of break in this situation, hence, index value of 1 is returned.
iii. Want to change lane: This time, the vehicle is only intending to change lane, therefore, the break needs not to be applied, and index value is 0.
iv. Break moderately: There is breaking capability, index value is 1.
v. Has changed lane: At this moment, index value is 0.
vi. Changing lane: Here, index value is computed as a probability distribution, this is because when changing lane, break may be applied or not, depending on the current situation. There are two possible events of either applying a break or not, probability of either applying a break or not will be$1/2 + 1/2 = (1 + 1)/2 = 2/2 = 1$. Therefore, there is tendency that break is applied when changing lane, hence, index value is 1

The following expression prints the required computation:

$$f(x) = \begin{cases} 0, & if\ x = \ default|want\ to\ change\ lane|has\ changed\ lane \\ 1, & otherwise \end{cases} \tag{12}$$

3. Status of driver: The vehicle's driver can have either normal status, with index value of 1, or abnormal status, with index value of 2. Based on the attribute collected from VISSIM, the status is considered as the interaction state of vehicle which is expressed as a function of five possible outcomes:
   i. Close up: the vehicle is stopping indicating abnormality; hence, index value returned will be 2.
   ii. Break: this is considered abnormal because the vehicle is being put to a halt; therefore, index value is 2.
   iii. Break ZX: it is considered abnormal because the vehicle experience break application, index value is 2.
   iv. Dwell: at this time, the vehicle experience delay that results into an index value of 2.
   v. Free: the vehicle is moving without any issue, this situation is considered normal, and the index number is 1.
   vi. Pass: the vehicle passes without any hitches; index number returned will be 1 because it is considered as a normal condition.

The following function computes the index value:

$$f(x) = \begin{cases} 1, & if\ x = \ pass|free \\ 2, & otherwise \end{cases} \tag{13}$$

4. Nature of the environment: If the environment condition is normal, the index value will be 1, otherwise it will be 2 for the abnormal condition.

The following function evaluates the environmental condition

$$f(x) = \begin{cases} 1, & if\ x = pass|free \\ 2, & otherwise \end{cases} \tag{14}$$

5. Distance between vehicles: this is known as inter-vehicles distance, distance of less than 5 m, 10 m and 20 m returns index numbers of 1, 0.5 and 0 respectively. The index values are obtained from the average of the net and gross following distances.

The following expression gives the corresponding index values:

$$f(x) = \begin{cases} 1\ if\ \left(\dfrac{x1 + x2}{2}\right) \leqslant 5 \\ 0.5\ if\ \left(\dfrac{x1 + x2}{2}\right) \leqslant 10 \\ 0\ otherwise \end{cases} \tag{15}$$

5.5. Reconfiguration of simulation setting

The simulation settings were reconfigured under different instances to capture different scenario in the IoV, this includes reduced speed, required speed distribution, number of vehicles travelling from a particular zone to another and etc. The reduced speed and required speed distribution were set as 5 km/h, 10 km/h, 15 km/h, 20 km/h and 50 km/h to capture different scenarios in the IoV environment. The

number of vehicles moving from one zone to another was set to 15, 14, 12, 20, 19, 2 and etc. to capture different scenario of the IoV. Various fog formations, linear and exponential were added to the simulation, these made the visibility to be very poor to capture additional IoV scenario, these instances have been made available on Figs. 7 and 8.

The numbers of vehicles traveling from a zone to another were given variable number, as the variable number grows, that will have effect on the traffic as well. Both the fog formation and the number of vehicles changing zone will have effect on the Delay Time of the moving vehicles.

Having established the six collision influencing features as the DRNN-LSTM models' inputs that will be passed into the DRNN-LSTM model as x-values, the resultant output for the detection of the rear end collision is obtained as the y-values that can either be a zero (0) or a one (1) corresponding to collision-free or collision.

5.6. Dataset size and partition

There is a total of two hundred and six thousand, seven hundred and seventy-six (206,776) datasets collected from the IoV environment through the simulator. The entire datasets were partition into two: training and testing. The training datasets constituted 80 % of the entire data equivalent to one hundred and sixty-four thousand, nine hundred and sixteen (164, 916). For performance consistency the dataset was partition into several ratios: 90:10, 80:20, 70:30, 60:40 and 50:50 to run the DRNN-LSTM. The training data is used for training the DRNN-LSTM model and the compared algorithms. The training datasets were further grouped into two: the x-train datasets (see Table 6 for sample) and y-train datasets. The x-train datasets is the DRNN-LSTM models' input. It is basically made up of the input features that cause collision to occur or not, data were passed to the DRNN-LSTM model for supervised learning. The x-train data were associated with the y-train datasets to train DRNN-LSTM model. The y-train datasets also called the models' output. The y-train datasets were obtained by mathematical comparison and relations in the following way:

Based on the values that were retuned, as shown in Table 5, when we sum up the entire inputs, raw wise, range of values VI where, i = 1,2 14were obtained as results, depending on what set of input numbers were passed or represented, and these range of values are were found to be: 3.35,3.65,4,4.35,4.65,4.85,5,5.35,5.65,6,6.35,6.65,6.85,7.35With each corresponding to Vi.where, 1 = 1,2…14. We defined a parameter p that was used to determine the y-train values into either a collision or collision-free scenarios by taking the average of these sets of numbers

$$Average(Av) = \sum_{i=1}^{14} V = \frac{3.35 + 3.65 + 4 + 4.35 + 4.65 + 4.85 + 5 + 5.35 + 5.65 + 6 + 6.35 + 6.65 + 6.85 + 7.35}{14} = \frac{74.05}{14} = 5.29 \tag{16}$$

Therefore, the value 5.29 is our p, and this implies that a function f (v)

$$f(V) = \begin{cases} 0, & if \ = \ 0 \\ 1, & otherwise \end{cases} \tag{17}$$

The function in Equation (23) evaluates the outputs, called the y-train values for the supervised learning (training) and detection of the rear end collision in IoV. An output of 0 represents collision-free scenario whereas an output of 1 represents collision scenario (see sample in Table 7). Where the V_{is} values are large, it signifies collision occurrence, whereas small values indicate collision-free scenario.

6. The proposed hybridization of deep recurrent neural network and long short term memory for collision detection

As represented by Fig. 9 (Wang et al., 2018), the LSTM has a memory cell that stores previous data. The input, forget and output gates are basic components of the LSTM that help update and utilize the stored information (Wang et al., 2018).Based on the discussion in (Wang et al., 2018), assuming that the variable letter h, cand × represents the LSTM cell output, parameter of the LSTM memory and the input data respectively. The LSTM unit update can be achieved as shown by steps:

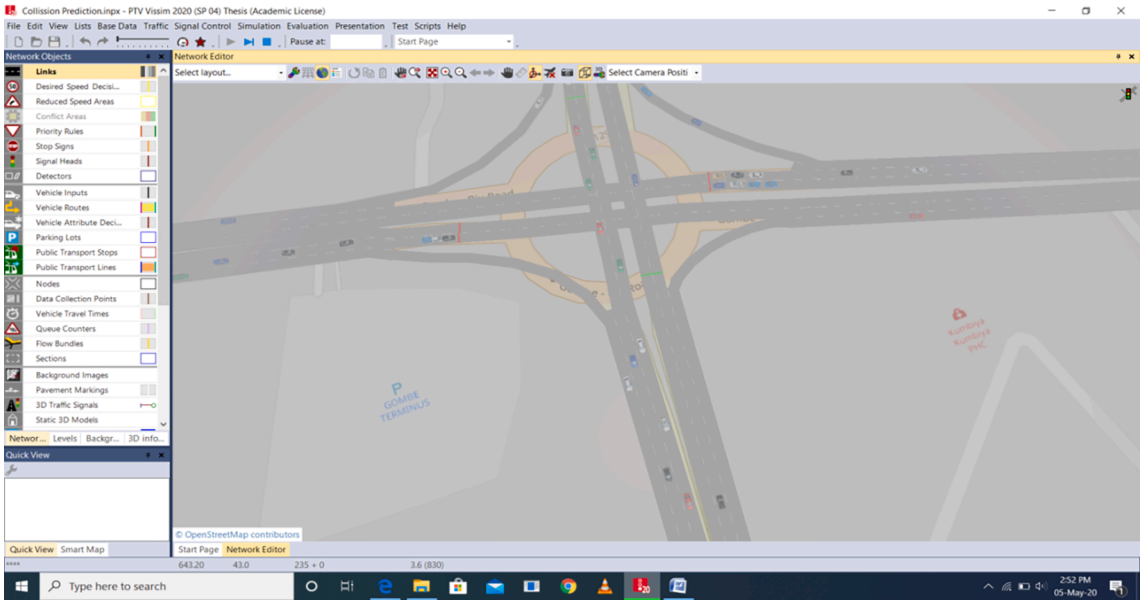

Fig. 7. Linear fog formation.

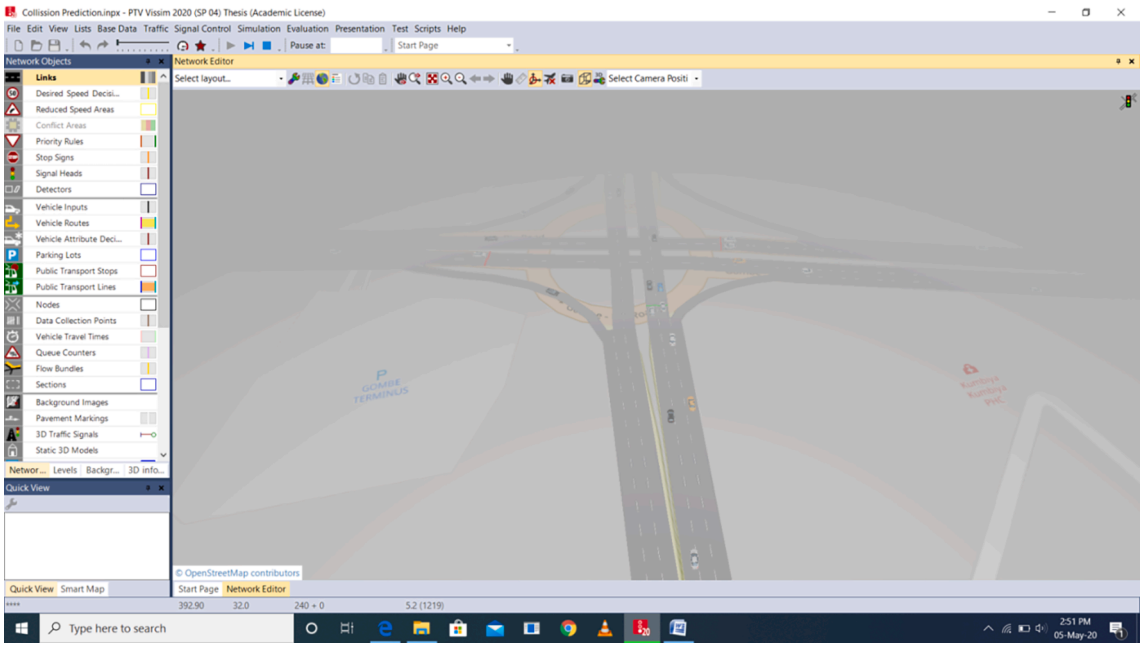

Fig. 8. Exponential fog formation.

Table 5
Describes the index values of collision influencing factors.

S/N	Driving state	Following distance (gross)	Following distance (net)	Interaction state	Delay time	Lane	Speed	No. of Lane	Status of driver	Nature of environment	Velocity of vehicle	Distance between vehicle	Breaking capability
1	Default	31.48	31.48	Close up	99.7	Singles	22.92	1	2	2	0.35	0	0
2	Default	47.03	42.82	Close up	1.77	Singles	21.72	1	2	1	0.35	0	0
3	Default	191.79	187.14	Pass	36.5	Double	54.55	2	1	1	0.65	0	0
4	Default	218.55	218.05	Close up	80	Double	112.3	2	2	2	1	0	0
5	Brakes hard	32.44	31.94	Close up	2.4	Double	37.13	2	2	1	0.35	0	1
6	Default	25.11	25.11	Close up	50.1	Singles	22.92	1	2	2	0.35	0	0
7	Default	41	36.78	Free	1.77	Singles	21.72	1	1	1	0.35	0	0
8	Default	177.52	172.88	Close up	0	Double	60.48	2	2	1	0.65	0	0
9	Default	217.39	216.89	Close up	65.8	Double	14.48	2	2	2	0.35	0	0
10	Brakes hard	20.46	19.96	Close up	3.05	Double	108.1	2	2	1	1	0	1
11	Default	18.74	18.74	Free	0.35	Singles	22.92	1	1	1	0.35	0	0
12	Default	34.96	30.75	Close up	1.77	Singles	21.72	1	2	1	0.35	0	0
13	Brakes moderately	61.03	61.03	Brake ZX	88.9	Double	53.37	2	2	2	0.65	0	1
14	Default	214.8	214.3	Pass	0.05	Singles	23.51	1	1	1	0.35	0	0
15	Brakes moderately	11.73	11.23	Close up	3.83	Double	15.35	2	2	1	0.35	0	1
16	Default	12.38	12.38	Close up	0.35	Singles	22.92	1	2	1	0.35	0	0
17	Default	28.93	24.72	Free	74.7	Singles	21.72	1	1	2	0.35	0	0
18	Brakes moderately	45.22	45.22	Brake ZX	0.36	Double	46.25	2	2	1	0.35	0	1
19	Default	209.53	209.03	Close up	0.1	Double	132.1	2	2	1	1	0	0
20	Brakes moderately	6.03	5.53	Close up	4.7	Double	6.9	2	2	1	0.35	0.5	1
21	Brakes hard	6.01	6.01	Close up	0.35	Singles	4.4	1	2	1	0.35	0.5	1
22	Default	22.9	18.69	Free	51.8	Singles	21.72	1	1	2	0.35	0	0
23	Brakes moderately	31.38	31.38	Brake ZX	0.76	Singles	39.14	1	2	1	0.35	0	1
24	Default	201.91	201.41	Close up	0.15	Double	38.25	2	2	1	0.35	0	0
25	Brakes moderately	2.93	2.43	Pass	67.7	Double	2.07	2	1	2	0.35	1	1

The individual memory cell content ct, w_{xc} and w_{ht} at the current instance are calculated based on the formula of the ordinary RNN representing the inputs and weight of the LSTM cell output at the previous time

$$\overline{c_t} = \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \quad (18)$$

The input gate value i_i is then evaluated which is used to manage the behavior of the current input data in respect to the state value of the memory cell. Evaluation of all gates is usually influenced by the current input data xt, output of the LSTM cell ht-1 and result of the memory cell $C_{c\text{-}1}$ based on the previous state.

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i) \quad (19)$$

The forget gate f_t which is used to manage the effect of the previous

Table 6
Sample of x-train values.

Number of Lanes	Status of driver	Nature of environment	Velocity of vehicle	Distance between vehicles	Breaking capability
2	1	1	0.35	0	0
2	1	1	0.35	0	1
2	1	1	0.35	0	1
2	1	1	0.35	0	1
2	1	1	0.35	0	1
2	1	1	0.35	0	0
2	2	1	0.35	0	0
2	1	1	0.35	0	0
2	2	1	0.35	0	0

Table 7
Coding sample of Collision or Collision-free for DRNN-LSTM to detect.

Collision free scenario or Collision scenario
0
1
1
1
1
0
1
0
1
0
1

information on the current memory cell is denoted by:

$$f_t = \sigma\left(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f\right) \tag{20}$$

The current memory cell status parameter c_t is expressed as:

$$c_t = f_t \odot c_{t-1} + i_t \odot \overline{c_t} \tag{21}$$

The symbol ʘ indicates the point by point product.

The memory cell gets updated based on its own state and the immediate candidate memory cell parameters: ct-1and ct which are realigned by the input and oblivion gates.

The output gate ot is evaluated by the expression:

$$o_t = \sigma(w_{x0}x_t + w_{ho}h_{t-1} + w_{co}c_{t-1} + b_o) \tag{22}$$

Last LSTM unit output is given by:

$$h_t = o_t \odot \tanh(c_t) \tag{23}$$

The sigmoid function in equation (15) ranges from 0 to 1. The three doors and the isolated memory cell enable the LSTM unit to store, extract, reset and update prolonged historical information.

Fig. 10 shows the architecture of the propose DRNN-LSTM. Based on our proposal, the DRNN and LSTM were hybridized for the rear-end collision detection in IoV environment with six input nodes corresponding to the features that cause the rear end collision. The two output nodes corresponding to collision-scenario and collision-free-scenario as the output.

The six inputs (x) from 1 to 6 present the inputs of the 6 collision influencing feature. The wt1 and wt2 are the corresponding outputs of the hybrid DRNN-LSTM: collision-free-scenario and collision-scenario, respectively. The model is accompanied by state as a function of time (t), i.e. the previous and the current state. L1, L2 and L3 are the forget

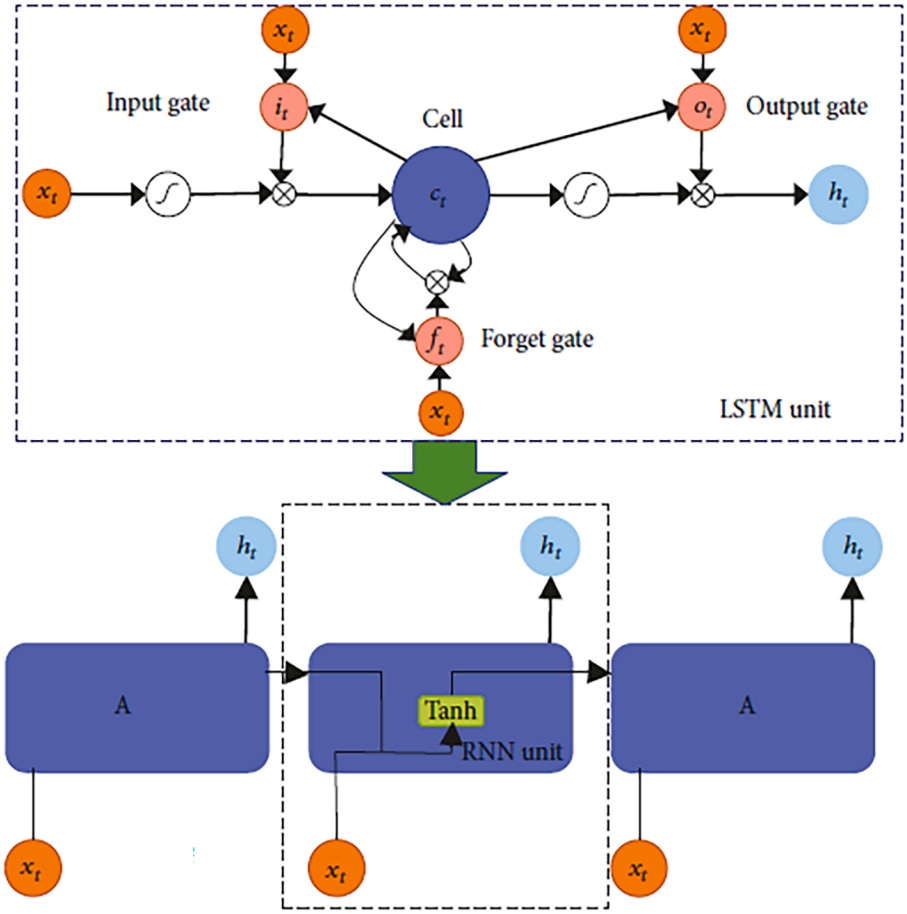

Fig. 9. The parameter of RNN-LSTM (Wang et al., 2018).

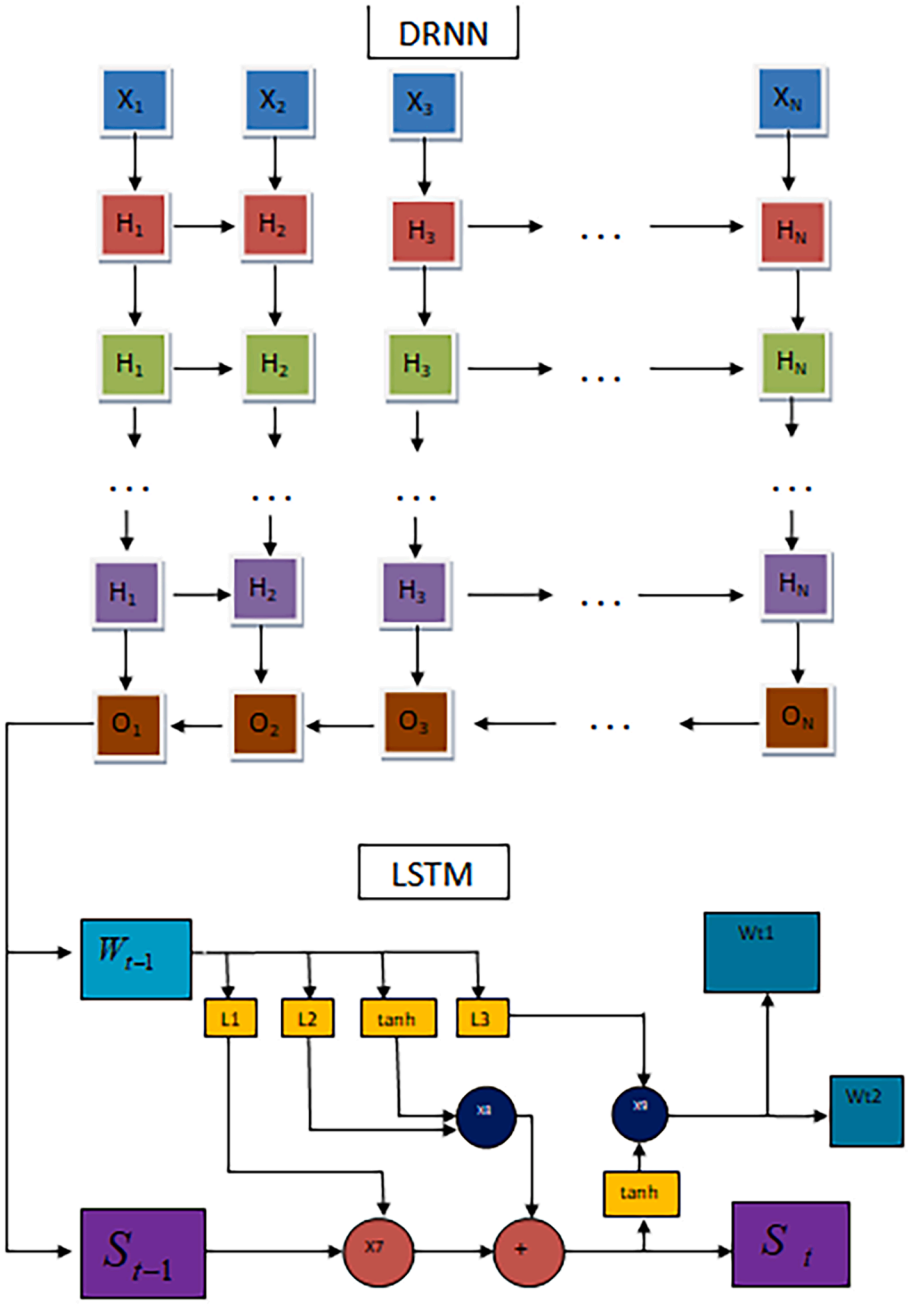

Fig. 10. Hybrid deep recurrent neural network – Long short term memory for the collision detection.

gate, input gate and the output gate. The keep gate helps in storing useful/relevant data, read gate helps in recovery of data whereas write gate helps in deleting irrelevant data. S_{t-1} serves as the output from the previous block, S_t is the output of the current block, W_{t-1} serves as the memory from the previous block while W_{t1} and W_{t2} are the outputs of the current block.

The pictorial representation of the proposed conceptual framework for the rear-end collision detection in IoV environment is presented in Fig. 11.

7. Results and discussion

This section presents discussion of results including the experiment setup, the propose DRNN-LSTM hyper-parameter settings, performance of the propose DRNN-LSTM in relation to browser, performance on collision detection in IoV and comparative study.

7.1. Experimental setup

The deep learning algorithm architectures were implemented on google colab (Colaboratory, 2018) using Python libraries, particularly Keras and Tensor-flow. The algorithms were run several times. The results of the propose DRNN-LSTM were compared with the constituent algorithms, hybrid, and shallow algorithm: ANN, BPNN-GA, LSTM and DRNN. The performance of the algorithms was measured based on the following performance metrics: AES, MSE and RMSE. The software platform used for the library is Keras and TensorFlow because both TensorFlow and Keras are opened source, as tensor is written both in C++ and python. The Keras is written in python which make the duo to look common; both are also popular and are used for academic purposes. Keras is used to easily define deep learning models on the back-ends of TensorFlow (Nguyen et al., 2019). For the Hardware, processor type: GPU. System specification: Windows 10 (64 bits) operating system and

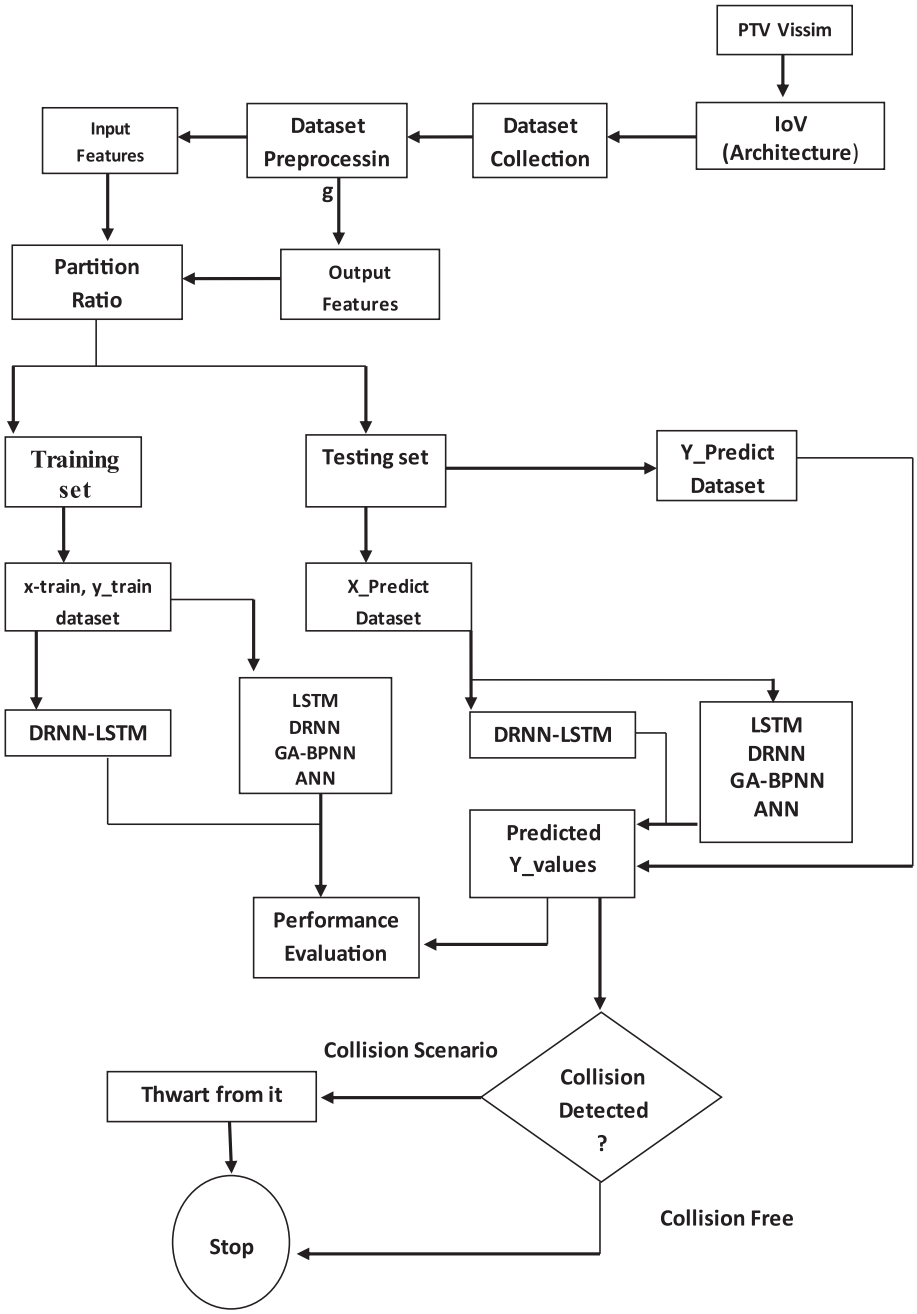

Fig. 11. Proposed conceptual frameworkfor the hybrid deep recurrent neural network – Long short term memory for rear end collision detection in IoV environment.

Table 8
Showing the performance for each deep learning model in collision detection in IoV.

Model	AES	MSE	RMSE
ANN	5 s, 29us/sample	0.000082	0.0090
BPNN-GA	39 s, 198us/S	0.000051	0.0071
LSTM	18 s, 1092us/sample	0.000038	0.0062
DRNN	24 s, 145us/sample	0.000078	0.0088
DRNN-LSTM	41 s, 246us/sample	0.000037	0.0060

processor of 2.90 GHz.

7.2. Preliminary experiment trials

A preliminary experiment was conducted to ascertain the performance of the algorithms on small sample size of the whole datasets. The propose hybrid DRNN-LSTM and the compared algorithms were applied for collision detection in IoV environment. Table 8 shows the performance of the models at the initial run.

At the initial run, the shallow and deep networks algorithms have the following setup:50 epochs, 1step for a set of data per epoch, 1 hidden layer for ANN and BPNN-GA, 3 hidden layers for LSTM, DRNN and DRNN-LSTM models. Data partition ratio 80:20 for training and testing, 0 drop out regularization and default batch size. We run an initial experiment with all the models and the result is presented in Table 8. The preliminary results as shown in Table 8, demonstrates effectiveness of proposed DRNN-LSTM in collision detection in IoV environment. Our proposed DRNN-LSTM achieved the least MSE and RMSE compared with those of the other four models. This means that the DRNN-LSTM has the best performance in terms of early detection and prevention of collision in IoV environment, followed by LSTM, BPNN-GA, DRNN and ANN respectively. Though, the DRNN-LSTM exhibits certain delay (AES).

Looking at the values in Table 8, DRNN-LSTM has the MSE: 0.000037 and RMSE: 0.0060; LSTM with MSE: 0.000038 and RMSE: 0.0062; BPNN-GA with MSE: 0.000051 and RMSE: 0.0071; DRNN with MSE: 0.000078 and RMSE: 0.0088; and then ANN with MSE: 0.000082 and RMSE: 0.0090. Based on the results in Table 8, ANN executes at 5 s (s), LSTM at 18 s, DRNN at 24 s, and BPNN-GA at 39 s and DRNN-LSTM at 41 s.

7.3. Hyperparameter tuning for the propose DRNN-LSTM

The setting of parameters has influence on the performance of the deep learning algorithms. As such, hyper-parameter of the propose DRNN-LSTM were fine-tuned to ascertain the optimal hyper-parameter settings of the algorithm that can yield optimal level of performance in detecting collision in IoV. Five different models were run with varying parameters while keeping the following parameters fixed: the number of hidden layers, epochs, batch size, training, and testing dataset ratio.

Having established the fact that our proposed model outperformed the other four models as shown in Table 8.The hyper parameters of the propose DRNN-LSTM were tuned to find out the optimal hyper parameters behavior of the model in terms of early collision detection and prevention in IoV environment. Based on the second column of Table 9 (Parameter tuning), we adopted the following parameters for hyper parameter tuning: number of step(s) which represents the number of rows of the training datasets that were passed at a time for a particular epoch, batch size indicates how the entire sets of the training data are taken by the model in a single epoch, the number of epochs shows the number of iteration perform by the model for error minimization, number of neurons were set to 50 for each of the 3 hidden layer of the deep learning algorithms while 50 neurons for the only 1 hidden layer of the shallow algorithms, drop out regularization was aimed at deactivating some of the neurons for better result and faster convergence.

Table 9
Hyper-parameter tuning.

S/N	Parameter tuning	AES	MSE	RMSE
1	3 steps, D_batch, 0 drop, 50 epochs	48 s, 291us/ Sample	0.0000120	0.00340
2	1 steps, D_batch, 0 drop, 50 epochs	22 s, 130us/S	0.0000720	0.00850
3	4 steps, D_batch, 0 drop, 50 epochs	56 s, 326us/S	0.0000560	0.00740
4	5 steps, D_batch, 0 drop, 50 epochs	66 s, 396us/S	0.0000610	0.00780
5	6 steps, D_batch, 0 drop, 20 epochs	80 s, 485us/S	0.0000240	0.01560
6	6 steps, D_batch, 0 drop, 50 epochs	82 s, 493us/S	0.0000122	0.00349
7	3 steps, 32_batch, 0 drop, 20 epochs	46 s, 279us/S	0.0000120	0.00340
8	3 steps, 100_batch, 0 drop, 20 epochs	22 s, 132us/S	0.0000128	0.00347
9	3 steps, 32_batch, 0 drop, 50 epochs	92 s, 556us/S	0.0000120	0.00340
10	3 steps, 100_batch, 0 drop, 50 epochs	41 s, 250us/S	0.0000820	0.00900
11	6 steps, 100_batch, 0 drop, 20 epochs	22 s, 132us/S	0.0000120	0.00347
12	3 steps, D_batch, 0.1 drop, 50 epochs	52 s, 312us/S	0.0000120	0.00347
13	3 steps, D_batch, 0.1 drop, 20 epochs	50 s, 304us/S	0.0000490	0.00700
14	6 steps, D_batch, 0.1 drop, 50 epochs	80 s, 485us/S	0.0000120	0.00340
15	6 steps, D_batch, 0.1 drop, 20 epochs	81 s, 491us/S	0.0000870	0.00930
16	3 steps, 32_batch, 0.1 drop, 20 epochs	52 s, 313us/S	0.0001200	0.01000
17	3 steps, 32_batch, 0.1 drop, 50 epochs	53 s, 322us/S	0.0001100	0.01000
18	3 steps, 100_batch, 0.1 drop, 20 epochs	24 s, 148us/S	0.0001100	0.010000
19	3 steps, 100_batch, 0.1 drop, 50 epochs	26 s, 154us/S	0.0000121	0.003480
20	6 steps, 100_batch, 0.1 drop, 50 epochs	41 s, 248us/S	0.0000120	0.003477
21	3 steps, D_batch, 0.2 drop, 50 epochs	50 s, 305us/S	0.0000121	0.003479
22	6 steps, D_batch, 0.2 drop, 50 epochs	79 s, 474us/S	0.0000120	0.003477
23	3 steps, 100_batch, 0.2 drop, 50 epochs	24 s, 146us/S	0.0000121	0.003480
24	4 steps, 64_batch, 0.2 drop, 20 epochs	35 s, 214us/S	0.00001501	0.012200
25	5 steps, 100_batch, 0.2 drop, 20 epochs	38 s, 230us/S	0.00001826	0.004288

Observing results in Table 9, it shows that the proposed DRNN-LSTM model performs the best at 3 steps on comparing row one (R1), R2, *R*3, R4 and R6.This means that the collision detection by the DRNN-LSTM can be achieved at the right time for possible avoidance of the collision in IoV. With 1 step, the proposed DRNN-LSTM realized an MSE: 0.000072 and RMSE: 0.0085 that makes the DRNN-LSTM to be very poor in detecting and prevent collision in IoV, whereas at 3 steps, the MSE was 0.000012 and RMSE was 0.0034. Hence, the more the number of steps, the more effective is the DRNN-LSTM as observed. But increasing the number of steps from 3 to 4 and 5 creates some randomness on the MSE and RMSE values as shown on *R*3 and R4. By implication, it means that the randomness will render the model to have uncertain behavior that will put the vehicle in the IoV at higher risk of collision. Proceeding to 6 steps or beyond, the result remains the same as shown in comparing R1 and R6. Therefore, setting the DRNN-LSTM model to 3 or 6 steps achieve the same result, meaning that the result cannot improve beyond that point. Therefore, it does not make any difference in its' ability to detect and prevent collision in IoV.

Increasing the number of steps makes the DRNN-LSTM model to run a little bit longer in accomplishing a collision detection in IoV, this can be seen on column AES, rowsR1, R2, *R*3, R4 and R6.Increasing the number of epochs seemed to enhance the effectiveness of the model as in R5 (20 epochs) and R6 (50 epochs) as observed, the MSE: 0.000024, RMSE: 0.0156 and MSE: 0.0000122, RMSE: 0.00349 respectively, therefore, large number of epochs value for the model increases its ability to detect and prevent collision in IoV. However, the AES seemed to have a slight change, as it was 80 s at R5 while 82 s at R6. A small value of batch size seems to enhance the learning rate of the model with slow execution whereas large value of batch size yield too much error with a faster program execution, R9 and R10 of Table 9 gives more insight. It means that a small batch size enhances the model performance towards more reliable/early collision detection in IoV. Drop out regularization seem to have little impact on AES as the MSE and RMSE remain the same. Considering the MSE and RMSE of R1, R12 and R21, all the values remained the same whereas 0, 0.1 and 0.2 drop outs achieved 48 s, 52 s and 50 s respectively. Therefore, dropout regularization does not seem to enhance the model performance in collision detection in IoV.

7.4. Impact of browser on the DRNN-LSTM model performance

The study investigated the impact of different browsers on the performance of the propose DRNN-LSTM in detecting and avoiding collision in IoV environment. The DRNN-LSTM is run on different browsers. Table 10 present the result of running the proposed DRNN-LSTM model

Table 10
The effects of browser on DRNN-LSTM model performance.

S/N	Browser	AES	MSE	RMSE
1	Firefox	68 s, 408us/S	0.000012091	0.0034772
2	Microsoft edge	51 s, 306us/S	0.0000079831	0.0028254
3	Google Chrome	50 s, 302us/S	0.000012091	0.0034772

with different browsers to find out the impact of the browsers on the DRNN-LSTM model performance. The types of the browser are shown in Table 10 with the corresponding result of detecting rear-end collision in IoV environment using DRNN-LSTM. The result suggested that the Mozilla Firefox and Google Chrome achieved the same MSE and RMSE of 0.000012091 and 0.0034772 respectively, whereas Microsoft Edge achieved the best performance in terms of MSE: 0.0000079831 and RMSE: 0.0028254.

Google chrome and Microsoft Edge convergence speeds are very close AES (50 s and 51 s respectively) while Firefox has the worst convergence speed. In terms of accuracy in the detection of collision in IoV environment, Microsoft Edge performs better than the compared browsers. The deduction here, is that, Microsoft Edge contains better resources that can enhanced DRNN-LSTM model's ability to detect and prevent rea end collision in IoV more than the Firefox and Google Chrome.

7.5. Number of hidden layer neurons to DRNN-LSTM model performance

From Table 11, column shows the number of neurons on each of the 3 hidden layers of the DRNN-LSTM architecture. Considering the values in Table 11, the DRNN-LSTM model comprised of 3 hidden layers. For instance, the R1 of Table 11 signifies that each of the 3 hidden layers of the model has 20 neurons. We made the number of the neurons in each hidden layer to be equal at the initial trials before running the experiment with different number of neurons on the hidden layers as shown in the last row to ascertain its effect on the model performance. Our observation of the results indicated that the more the number of neurons the better the performance/effectiveness of the DRNN-LSTM model before it stops improving. With 20 neurons, the MSE and RMSE were found to be 0.000012172 and 0.0024883 respectively, whereas 100 neurons give MSE: 0.000012056, RMSE: 0.00347764, hence, as the number of the neurons increases, the error rate decreases. Alternatively, by making the first, second and third hidden layers of the model to be 100, 90 and 70 respectively, the error rate tends to increase thereby making the DRNN-LSTM model less effective in detecting collision in IoV. Hence, there is need to keep growing the number of neurons in a uniform pattern to achieve the best outcome as evident in our experiment (see Table 11). This means that the DRNN-LSTM model will achieve an enhanced collision detection rate with more uniform number of neurons. The AES of the model keeps growing as the number of neurons grow, this suggest that when there are many neurons in the network, the model becomes slow in accomplishing its' task or execution. For instance, 20, 50, 70 and 100 neurons converges at 27 s, 55 s, 57 s, and 94 s respectively for the detection of the collision in IoV. However, the priority is accurate detect of collision with minimum error.

7.6. Impact of number of hidden layers on DRNN-LSTM model performance

Varying number of layers from 2 to 5 as shown in Table 12 were configured with a uniform number of neurons to find out its effects on the performance of DRNN-LSTM model. The experiment starts with 2 hidden layers, the model attained an MSE: 0.000012386 and RMSE: 0.0035193749, 3 hidden layers, the MSE: 0.000012056 and RMSE: 0.003472175, 4 hidden layers yields an MSE: 0.000012101 and RMSE: 0.003478649, 5 hidden layer gives MSE: 0.000012092 and RMSE: 0.0034773553. The effectiveness of the DRNN-LSTM improves as the number of hidden layers increases and it starts declining as the hidden layers increases to 4 and 5. As such, the experiment stopped at 5 hidden layers. Therefore, the DRNN-LSTM model is most effective with 3 hidden layers because it has the minimal error rate. This means that, the proposed DRNN-LSTM model can detect collision in IoV better with DRNN-LSTM that has 3 hidden layers than any other architecture with 4 hidden layers and above.

Table 11
How varying number of hidden layer neurons affects DRNN-LSTM model performance in detecting rear end collision in IoV.

Number of Neurons	AES	MSE	RMSE
20, 20, 20	27 s, 164us/S	0.000012172	0.0034883
40, 40, 40	43 s, 259us/S	0.000012099	0.0034780
50, 50, 50	55 s, 333us/S	0.000012098	0.00347821
60, 60, 60	56 s, 338us/S	0.000012092	0.00347720
70, 70, 70	57 s, 346us/S	0.00002090	0.00347706
100, 100, 100	94 s, 566us/S	0.000012056	0.00347218
100, 90, 70	56 s, 341us/S	0.000012094	0.00347764

Table 12
Number of layers in the hidden part of DRNN-LSTM model vs performance.

Number of layers	Number of Neurons	AES	MSE	RMSE
2	100, 100	40 s, 242us/S	0.000012386	0.0035193749
3	100, 100, 100	94 s, 566us/S	0.000012056	0.003472175
4	100, 100, 100, 100	46 s, 288us/S	0.000012101	0.003478649
5	100, 100, 100, 100, 100	56 s, 340us/S	0.000012092	0.0034773553

7.7. Propose DRNN-LSTM rea-end collision detection in internet of vehicles

Results (Tables 8 to 12) of the preliminary experiments conducted shows that the optimal architecture of the DRNN-LSTM is as follows: 3 hidden layers, 100 neurons, 0 drop out regularization, 3 steps for a set of data per epoch and multiple dataset partition ratio. The preliminary experiment shows that our proposed DRNN-LSTM model demonstrated better ability for collision detection in the IoV environment compared to the comparison algorithms. The DRNN-LSTM is applied to detect collision in IoV environment, performance shows that the DRNN-LSTM is a better algorithm for detecting collision in IoV compared to the algorithms already discussed by previous researchers. To ensure consistency of the performance of DRNN-LSTM in detecting collision in IoV, different ratio for the training and testing datasets were configured to ascertain the influence of the data partition ratio with respect to the model performance. Varying data partition ratios were considered and run due to the non-deterministic nature of the *meta*-heuristic algorithms (Subotic et al., 2012). The summary of the hybrid DRNN – LSTM is presented in Table 13.

Varying training and testing dataset partitions were run. For instance, as shown in Table 14, each time the DRNN-LSTM and the compared algorithms were run to detect collision in IoV environment, our proposed model achieved the best performance. It is found that the more the training data, the more the effectiveness of the DRNN-LSTM. Therefore, with the ratio of 90:10, our proposed model achieved the best result with the least error because of the large number of training with less data for testing. However, the compared algorithms exhibit some randomness in their outcome with respect to the training and testing data partition ratio unlike the DRNN-LSTM model that exhibits

Table 13
The configuration of the hybrid DRNN – LSTM.

Parameter	Configurations
Number of hidden layers	3
Number of neurons in first hidden layer	100
Number of neurons in second hidden layer	100
Number of neurons in third hidden layer	100
Steps	3
Drop	0

Table 14
DRNN-LSTM detecting rea end collision in IoV with varying data partition ratio.

Algorithm	% Training	% Testing	AES	MSE	RMSE
DRNN-LSTM	50	50	45 s, 433us/S	0.000010050	0.0031701700
DRNN			14 s, 136us/S	0.000010096	0.0031774200
LSTM			17 s, 163us/S	0.000112810	0.0106212050
ANN			1 s, 11us/S	0.000012745	0.0035700140
BPNN-GA			21 s, 148us/S	0.000012001	0.0034642450
DRNN-LSTM	60	40	47 s, 381us/S	0.000016088	0.0040109849
DRNN			13 s, 121us/S	0.000016206	0.0040256670
LSTM			19 s, 149us/S	0.000052484	0.0072445800
ANN			1 s, 11us/S	0.000017189	0.0041459610
BPNN-GA			17 s, 119us/S	0.000017023	0.0041258930
DRNN-LSTM	70	30	82 s, 566us/S	0.000013751	0.0037266607
DRNN			19 s, 129us/S	0.000014787	0.0038453868
LSTM			24 s, 166us/S	0.000073175	0.0085542350
ANN			2 s, 11us/S	0.000014151	0.0037617814
BPNN-GA			25 s, 153us/S	0.000014121	0.0037577919
DRNN-LSTM	80	20	84 s, 566us/S	0.000012092	0.0034782179
DRNN			20 s, 124us/S	0.000013362	0.00365540695
LSTM			30 s, 181us/S	0.000017313	0.00416088930
ANN			2 s, 11us/S	0.000016462	0.00405733900
BPNN-GA			18 s, 163us/S	0.000015421	0.00392695810
DRNN-LSTM	90	10	87 s, 418us/S	0.000010347	0.00321667530
DRNN			33 s, 150us/S	0.000010783	0.00328374780
LSTM			28 s, 150us/S	0.000012229	0.00349699870
ANN			2 s, 11us/S	0.000010802	0.00328663960
BPNN-GA			36 s, 155us/S	0.000010713	0.00413884040

consistent performance with steady improvement of the MSE and RMSE.

Therefore, 90 % of data for training and 10 % of data for testing enhances the model's ability to detect collision in IoV much faster than using any other data partitioning ratio. Another observation is that, as the training data increases, the convergence time increases. In most cases, large values of the AES suggest possible delay as a result the model training experience with large data for training. It is found that with our proposed DRNN-LSTM in place, in the event that a particular vehicle from behind tends to hit another vehicle within the IoV environment, the proposed DRNN-LSTM can easily detect such a scenario immediately, thereby, alerting the immediate vehicle in advance in the IoV, possibly to apply break for speed reduction or total halt or maneuvering for safety of lives and properties. Early detection of collision in IoV can reduce casualties/damages resulting from collision occurrence.

8. Conclusions

In this paper, a rear-end collision detection algorithm is proposed. A hybrid LSTM-DRNN model is proposed rather than using the constituent algorithms to detect rear-end collision in the IoV environment. Shallow, deep learning and shallow hybrid algorithms such as ANN, DRNN, LSTM and BPNN-GA were used for comparison to ascertain the performance of our proposed scheme. It was found that the hybrid DRNN-LSTM model proposed in this research achieved better performance in terms of least MSE and RMSE compared to the comparison algorithms. This paper contributes to the IoV by providing an alternative means of rear-end collision detection with better reliability and performance compared to the algorithms already discussed by researchers. Thereby improve the guaranteeing of safety and efficiency of the vehicles and individual's mobility in IoV. It will be interesting to find out the performance of the propose hybrid DRNN-LSTM algorithm on multiple datasets generated from different IoV architecture with different features simulating different bad weather conditions such as heavy rainfall, high wine speed, sandstorm, etc.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the support of the Deanship of Scientific Research of University of Hafr Al Batin (UHB), Hafr Al Batin, Saudi Arabia through Research Grant no.: IFP-A-2022-2-1-11.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2022.119033.

References

Baek, M., Jeong, D., Choi, D., & Lee, S. (2020). Vehicle trajectory prediction and collision warning via fusion of multisensors and wireless vehicular communications. *Sensors, 20*(1), 288.

Brik, B., & Ksentini, A. (2021). Toward optimal MEC resource dimensioning for a vehicle collision avoidance system: A deep learning approach. *IEEE Network, 35*(3), 74–80.

Chang, W. J., Chen, L. B., & Su, K. Y. (2019). DeepCrash: A deep learning-based Internet of vehicles system for head-on and single-vehicle accident detection with emergency notification. *IEEE Access, 7*, 148163–148175.

Chang, C. C., Ooi, Y. M., & Sieh, B. H. (2021). IoV-based collision avoidance architecture using machine learning prediction. *IEEE Access, 9*, 115497–115505.

Chen, L. B., Su, K. Y., Mo, Y. C., Chang, W. J., Hu, W. W., Tang, J. J., & Yu, C. T. (2018, September). An implementation of deep learning based IoV system for traffic accident collisions detection with an emergency alert mechanism. In *2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)* (pp. 1-2). IEEE.

Chen, C., et al. (2018). A rear-end collision prediction scheme based on deep learning in the Internet of Vehicles. *Journal of Parallel and Distributed Computing, 117*, 192–204.

Chiroma, H., Ezugwu, A. E., Jauro, F., Al-Garadi, M. A., Abdullahi, I. N., & Shuib, L. (2020). Early survey with bibliometric analysis on machine learning approaches in controlling COVID-19 outbreaks. *PeerJ Computer Science, 6*, e313.

Colaboratory, G. (2018). "Wecome To Colaboratory." 2021, from https://colab.research.google.com.

Feki, M. A., et al. (2013). "The internet of things: the next technological revolution." Computer(2): 24-25.

Fong, S., et al. (2018). How meta-heuristic algorithms contribute to deep learning in the hype of big data analytics. Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, Springer: 3-25.

Hammedi, W., Brik, B., & Senouci, S. M. (2022). Toward optimal MEC-based collision avoidance system for cooperative inland vessels: A federated deep learning approach. *IEEE Transactions on Intelligent Transportation Systems.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735–1780.

Joshi, A. V. (2019). *Machine Learning and Artificial Intelligence*. Springer.

Kaiwartya, O., et al. (2016). Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. *IEEE Access, 4*, 5356–5373.

Kim, S. H., et al. (1999). *An experimental investigation of a CW/CA system for automobiles.* SAE Technical Paper.

Labriji, I., et al. (2021). Mobility aware and dynamic migration of MEC services for the Internet of Vehicles. *IEEE Transactions on Network and Service Management, 18*(1), 570–584.

Lin, T., et al. (1998). How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies. *Neural Networks, 11*(5), 861–868.

Liu, J. (2008). "Wavelet Basis Function Neural Networks for Sequential Learning." IEEE Access.

M. Subotic, M., Tuba, N. Bacanin, D. Simian (2012). Parallelized Cuckoo Search Algorithm for Unstrained Optimization, in. Proceedings of the 5th WSEAS Congress on Applied Computing Conference, and Proceedings of the 1st International Conference on Biologically Inspired Computation, World Scientific and Engineering Academy and Society (WSEAS), 2012, pp. 151-156, World Scientific and Engineering Academy and Society (WSEAS).

Maeda, Y., & Wakamura, M. (2005). Simultaneous perturbation learning rule for recurrent neural networks and its FPGA implementation. *IEEE Transactions on Neural Networks, 16*(6), 1664–1672.

Nguyen, G., et al. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review, 52*(1), 77–124.

Nkenyereye, L., et al. (2019). Towards secure and privacy preserving collision avoidance system in 5G fog based Internet of Vehicles. *Future Generation Computer Systems, 95*, 488–499.

N. T. S. Board. (2001). Special investigation report-highway vehicle and infrastructure-based technology for the prevention of rear-end collisions. NTSB Number SIR-01/01.

Ok, J.-S., et al. (2021). "A Survey of Industrial Internet of Things Platforms for Establishing Centralized Data-Acquisition Middleware: Categorization, Experiment, and Challenges." Scientific Programming 2021.

PTV-Group (2019). "PTV Vissim." from http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/.

Qu, H., Li, W., & Zhao, W. (2020). Human-vehicle collision detection algorithm based on image processing. *International Journal of Pattern Recognition and Artificial Intelligence, 34*(08), 2055015.

Rebala, G., et al. (2019a). *Deep Learning* (pp. 127–140). An Introduction to Machine Learning: Springer.

Rebala, G., et al. (2019b). *An Introduction to Machine Learning*. Springer.

Ridge, B., et al. (2020). Training of deep neural networks for the generation of dynamic movement primitives. *Neural Networks, 127*, 121–131.

Schmidhuber, J., et al. (2002). Learning nonregular languages: A comparison of simple recurrent networks and LSTM. *Neural computation, 14*(9), 2039–2041.

Shu, X., et al. (2020). Stage of charge estimation of lithium-ion battery packs based on improved cubature kalman filter with long short-term memory model. *IEEE Transactions on Transportation*. Electrification.

Tang, Q. Y., & Zhang, C. X. (2013). Data Processing System (DPS) software with experimental design, statistical analysis and data mining developed for use in entomological research. *Insect Science, 20*(2), 254–260.

Wang, D., et al. (2018). "Research on optimization of big data construction engineering quality management based on RNN-LSTM." Complexity 2018.

Wang, X.-S., et al. (2014). Analysis of risk factors for suburban highways using hierarchical negative binomial model. *Zhongguo Gonglu Xuebao, 27*(1), 100–106.

WHO (2015). "Global status report on road safety 2015." from https://www.who.int/violence_injury_prevention/road_safety_status/2015/en/.

Wang, X., Liu, J., Qiu, T., Mu, C., Chen, C., & Zhou, P. (2020). A real-time collision prediction mechanism with deep learning for intelligent transportation system. *IEEE Transactions on Vehicular Technology, 69*(9), 9497–9508.

Yang, et al. (2017). *Roadside Infrastructure Planning Scheme for the Urban Vehicular Networks*. IEEE Access.

Zhang, G., et al. (2014). Traffic violations in Guangdong Province of China: Speeding and drunk driving. *Accident Analysis & Prevention, 64*, 30–40.