Review of Computer Engineering Research
2022 Vol. 9, No. 3, pp. 169-180.
ISSN(e): 2410-9142
ISSN(p): 2412-4281
DOI: 10.18488/76.v9i3.3145
© 2022 Conscientia Beam. All Rights Reserved.

DEEP LEARNING BASED INTRUSION PREVENTION SYSTEM IN VEHICULAR NETWORK

Badugu Samatha[1+]
Thalakola Syamsundararao[2]
Nagarjuna Karyemsetty[3]

[1]*Department of CSE, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur, India.*
Email: drbs_cse@vignan.ac.in
[2]*Department of CSE, Kallam Haranadhareddy Institute of Technology (A), Chowdavaram, Guntur, India.*
Email: Syamsundar@khitguntur.ac.in
[3]*Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India.*
Email: nagarjunak@kluniversity.in

(+ Corresponding author)

Article History
Received: 17 June 2022
Revised: 25 August 2022
Accepted: 7 September 2022
Published: 30 September 2022

Keywords
CICIDS 2018
Deep learning
Google Colab
Intrusion prevention system
KDD 99
Safety
Vehicular network.

ABSTRACT

In the internet of vehicles, safety-based communication is carried out for prevention, mitigation, and alleviation of accidents through cooperative messages, position sharing, and the exchange of speed data between the vehicle (nodes) and corresponding roadside units. However, such networks are susceptible to false alarms and mispositioning of vehicles. It is therefore imperative to authenticate and identify normal messages from aggressive and incorrect messages. In this context, this paper has emphasized on a deep learning technique utilizing binary classification for segregating normal and malicious packets. The procedure is initiated by the preparation of training datasets from KDD99 and CICIDS 2018-type of open-source datasets having 1,20,223 packets and 41 features. An autoencoder is used in the preprocessing stage for the elimination of undesirable data right from the beginning. The 23 salient features are filtered out of 41. For training of the models, a structural deep neural network is utilized along with a Softmax classifier and rectified linear unit (ReLU) activation functions. The complete intrusion prevention (IP) mechanism is further trained & tested with Google co-labs as the open platform cloud service with open-source tensor flow. Furthermore, simulation data set developed in a network-simulating procedure is used for validation of the model. The results of the experimentation have established an accuracy 99.57% greater than existing recurrent neural network and convolution Neural Network models. For work in future, various datasets can be used for training to improve the accuracy and efficacy.

Contribution/Originality: This study provides a model for preventing intrusive messages from the vehicular network and a proposed model validated with real-time vehicular network-simulation data in addition to training and testing.

1. INTRODUCTION

For communication of messages relating to safe driving in protecting vehicular systems and their respective drivers and passengers, IOV (Internet of Vehicles) carries a significant contribution. Besides gateways, farewells similar to wired networking structures, vehicle-related networking is susceptible to many types of attacking transgressions. It is known that vehicular ad hoc networks (VANETs) as part of the IOV works on an ad hoc basis and operates in unsafe areas of high sensitivity, covering the manipulation of communication devices, spams, sending away, and masquerades. Serenity execution [1] for IOV is rather challenging in nature. For effective execution, it has to conform to the stipulation of the protecting type with respect to attacks as well as fallacious vehicular nodal points. Detection of intrusion [2] is very important to ensure the secure operation of any vehicle-related networking. Many detection & prevention techniques are in practice for intruding messages: these involve covering statistics, clustering, ANNs, as well as deep learning. Owing to adoptive characteristics as well as self-related learning, the last method is preferred. Figure 1 shows the IDP VANET model consisting of three OBU (OBU1 to OBU3)-mounted DL-based IPS to detect and prevent intruder messages in the network and One OBU4 is an intruder vehicle node going to attack in the network. All three OBUs are mounted with a proposed DL-trained classifier and able to prevent false node from invading the network. Each OBU generates the safety message and communicates to other OBUs with a specific pattern. Each OBU shares its location and speed to nearby OBU2, OBU3, and OBU4 nodes. OBU2 and OBU3 are able to classify the intruder node or normal node. Each node uses a Global Positioning System (GPS) to receive the position data and speed data from satellite. The accuracy of the location varies by ± 1 meter and accuracy of speed varies by ±5 km/h. As this is sensitive data, to increase the accuracy, dual-band GPS or military GPS can used without compromising the price to mitigate the error. Figure 1 demonstrates four cars in which three are normal (OBUs) integrated with DL_IPS model and one is Intruder OBU4 (red color vehicle OBU in given picture), which can generate abnormal messages in the network. OBU1 - OBU3 are fabricated with a GPS receiver, speed monitor module, alert system, and a DL-based intelligent safety classifier. Open-source tools such as NS2 2.34 [3] were used to simulate the network ranging from low to high (OBU1 to OBU100, shown in image 2) with varying simulation durations, low to complex traffic congestions, and most importantly, varying speeds with 10 km/h to 300km/h. OBU1_to_OBU100 were capable of generating safety packets, normal packet, and route to destination OBU as per the predefined routing algorithm. OBU101 to OBU105 are defined as OBU_Intruders capable of generating abnormal messages and misguiding the normal OBUs. Figure 2 represents the sample output of simulation of a real-time vehicular network experiment by considering the Vignan's University campus scenario using an open-source real-time traffic simulator, SUMO 1.13, latest version.

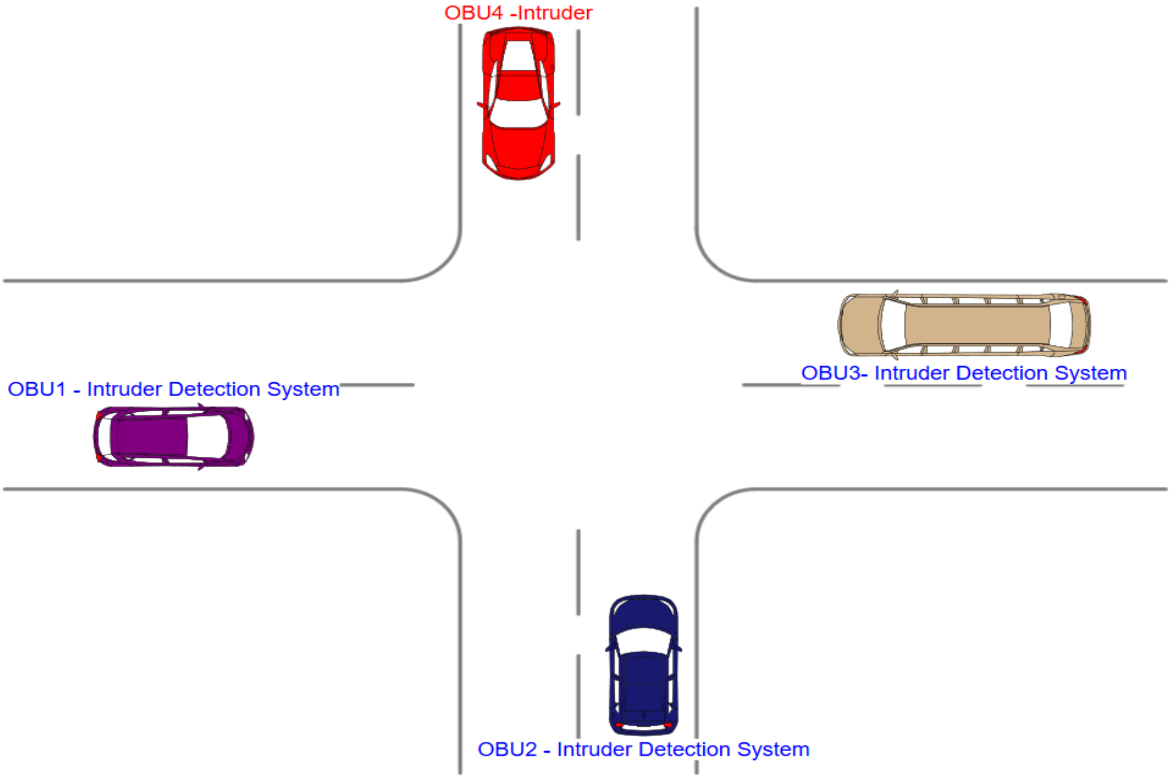
OBU4 -Intruder
OBU1 - Intruder Detection System
OBU3- Intruder Detection System
OBU2 - Intruder Detection System

Figure 1. Vehicular network model.

Correction & interaction with cameras (CAM), VANETs, supporting infrastructure on the RSUs and vehicle-based nodal parts are utilized for the detection of malicious characteristics. Intrusion Detecting System (IDS) can effectively help with the identification of intruding elements. This warrants the need of obtaining or transferring packets among the vehicular parts and examining the same. Utilization of information of IOV; helps in detection normally obtained and maliciously available characteristics. Figure 2 describe the experiment results of vehicular network consisting of 100 OBUs, out of 100 OBUs 5 OBUs are Intruder OBUs. So in the network all OBUs including normal OBUs and intruder OBUs will generate network traffic. All normal OBUs will able to classify the normal safety message generated from normal OBUS from Abnormal messages generated from intruder OBUs. Each normal OBU preprocess the generated data by selecting necessary feature and apply the feature selection to narrow down the traffic data.

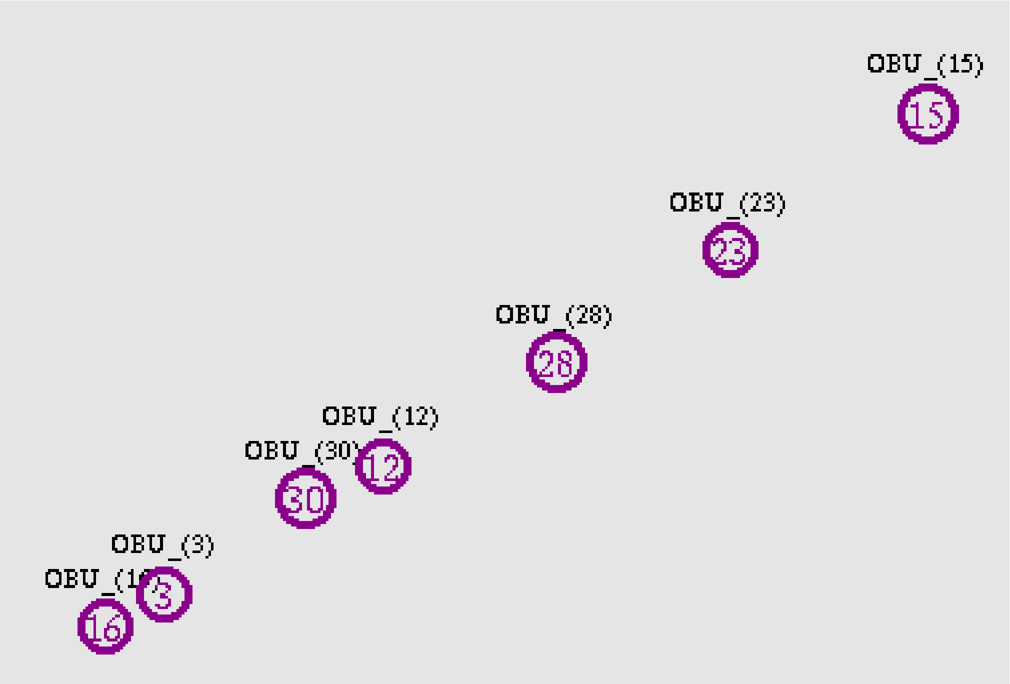

Figure 2. Experimental vehicular network.

Architectural decision[2] of normal IOV is shown in fig 1 that has on bond units software integration along with IDS OB01& OB03 as well as red-colored intruding OB04 are displayed NS2 2.34[3] has been used on an appearance basis for simulating beginning with low up to extrusive Figure 2 SOMO [Simulation of Urban Mobility] as a pen source simulating setup [4] is utilized for generating paths and networking rotes. Many simulating not mention like low medians 7 high types of dense networking for parametric evaluation of size, routes, range and size of packets.

Deep learning a kind of learning in which train the neuron for learn from the existing datasets until the satisfactory accuracy of model. Each neuron will become intelligent by learning from available data such as KDD99 and CICIDS 2018. In order to maximize the amount of information that can be gleaned from the datasets, this method expands the data gather. In addition, deep learning automatically examines all of the dataset's micro-characteristics and selects the most relevant and ongoing learning. Convolution, pooling, Softmax, and optimization layers are also used in this method. Every computer and communication device is unique in the way that it operates. It is critical that the relationships between the various levels be established in the correct order. Each iteration of the network's weights should be re-optimized to stabilize it. Analyzing the data yields the error differences, which are then used to adjust the weights in subsequent rounds. Because of its adaptive character, it aids in the improvement of networks' learning capabilities and the precision and accuracy of testing.

Table 1. Sample output of network simulation.

Event	Time	Node_id	X_Pos	Y_Pos	Pkt_Type	Protocol	Type_msg	Pkt_size
s	0.067	13	6704.93	6910.51	-99	AGT	DSRCApp	100
r	0.068	28	6704.93	6910.51	0	RTR	DSRCApp	100
s	0.025	14	6704.93	6910.51	-99	RTR	DSRCApp	120
s	0.057	16	6704.93	6910.51	-99	MAC	Message	148
r	0.079	21	6704.93	6910.51	-99	MAC	DSRCApp	120
r	0.082	12	6704.93	6910.51	0	RTR	DSRCApp	120
r	0.082	13	6704.93	6910.51	-99	AGT	DSRCApp	120
s	0.213	14	11478.7	1429.32	0	AGT	Message	100
r	0.021	12	11478.7	1429.32	0	RTR	Message	100
s	0.027	15	11615.1	3812.88	0	AGT	DSRCApp	100
r	0.028	18	11615.1	3812.88	-99	RTR	DSRCApp	100
s	0.034	12	15326.7	4352.5	-99	AGT	DSRCApp	100
r	0.035	22	15326.7	4352.5	-99	RTR	DSRCApp	100
s	0.037	24	11615.1	3812.88	0	RTR	Message	32
s	0.038	19	11615.1	3812.88	0	MAC	Message	60

The outputting part of the simulating system shown in Table 1 consists of types of packet time for transmitting, destination protocols, sources, destinations data & messaging, among other factors.

It is pertinent to note that DL [deep cleaning] [5, 6] is part of ML which integrated the features and characteristic in Artificial Intelligence (AI). It involves many intermediate neural learning as well as the I/O layers. The data collection is improved so that machine learning facilitates knowledge through data collection. All minute specifications of data collection are intermittently taken up by DL in related learning, operating continuously. Various CNNs (Conditional Neural Networks) are also in the framework of DL. It is essential to optimize the weighting pattern allocation following each interacting operation for stabilizing the structure. In the analyzing stage, the gaps between related weighting factors are noted and modified. This adaptive system precision improves the accuracy of the operations.

As the requirement of higher computational power exists, traditional CPO is not sufficient. Hence, the GPU (Graphical Processing Unit) of Google is needed for the creation of a proper framework for deep cleaning & machine learning. Various Python models are used to manage huge amounts of data at higher speeds from the cloud stage. Suitable testing & storing backup is also provided.

It is known reinforcement that makes DNN suitable for KDD-CUP99 with DR 99% and FAR 0.08%. Dong, et al. [7]; Tuohy, et al. [8] have used a fast DNN with activation and software layer for supervised learning. Wang, et al. [9]; Lopez, et al. [4] have used NSL- KOD data for fast-track random forests (RF) and support vector models (SVM) with DR97.5% and FAR 3.5%. Rawat and Wang [6] have worked on Recursive Neural Networks (RNN) having 73.67% DR and 2.97% FAR for data analysis of traffic flow. Further, 3 classification methods have been tried on IDS; Bat has been used for the correlation-based on CICIDS 2018 database. Dong, et al. [7] have worked on the detection method of RNN and obtained DR72.95% and FAR 3.44%. Wang, et al. [9] have used L5TM – RNN and ADFA dataset for achieving DR of 90% and FAR of 16%. It is worthwhile to note that a deep belief network consists of layers to layers RBM (Restricted Boltzman machine). In this endeavor, DNNs have been used with 0.1% rate learning and variation of epochs for KDDCup -99 data. A novel machine learning method has been used for prediction. Random forest suggested by Elmsory et al. was used for feature selection. SVM with 12 selected features was used on KDD- 99 data for forty-one factors of classification.

2. MATERIALS AND TOOLS

Google colabs' GPU and Tensor Computing Unit (TPU) with 16GB of RAM and 547 dB was used. It was ensured that the GPU software is not utilized for illegalities like cryptocurrency. Open-source platforms are more encouraged for use in the development of research projects. The required computation power and resources utilized from google cloud is a free open-source environment.

The salient features of configuration are:

a) None indicating usage of CPU of PC & no external support.

b) GPU for Graphical processing.

c) TPU for tensor processing.

It is to be noted that a runtime form is used for processing after starting a google co- lab file.

Table 2. Description of sample data.

prtocl_type	ser_type	src_bytes	dst_bytes	num_root	num_shell	count	diff_rate	outcome
tcp	http	215	45076	1	1	0	0	normal.
tcp	http	162	4528	2	2	1	1	normal.
udp	ftp	236	1228	1	1	2	0.5	abnormal.
tcp	finger	233	2032	2	2	3	0.33	normal.
tcp	http	239	486	3	3	4	0.25	normal.
icmp	http	238	1282	4	4	5	0.2	normal.
tcp	ftp	235	1337	5	5	6	0.17	abnormal.
udp	http	234	1364	6	6	7	0.14	abnormal.
tcp	finger	239	1295	7	7	8	0.12	abnormal.
tcp	http	181	5450	8	8	9	0.11	normal.
udp	http	184	124	1	1	10	0.1	normal.
tcp	http	185	9020	2	2	11	0.09	normal.
tcp	http	239	1295	1	1	12	0.08	normal.
udp	login	181	5450	2	2	13	0.08	normal.

Table 2 shows that single data CICIDS 2018 has no malicious, up to date information simulating real-world infarction and covering the results of traffic analysis with time stamps, protocols, and files of attack. There are eight thousand normal and similar attacks of brute force, DOS, port scan, & ping scan. In total, 120,223 data attributes are used in experimentation. The outcome of the data is learned from most other relevant attributes, including destination Bytes (dstByte), source bytes (srcBytes), super user attempts count (suAttmt), diffRates, no. of root users attempts, no. of shell user attpemts, etc. Protocol type contains tcp, upd, icmp, etc. Values, which are filtered using one hot encoding technique by converting into numerical data for computer processing. KDD'99 [10] DARPA developed a dataset utilizing recorded network traffic in 1999. It is preprocessed into 40 characteristics without a final outcome per network connection. Basic traffic features (No. 1 to No.9), core features (No.10 to No.22), time-related traffic specifications (No.23 to No.31), and host-related traffic specification (No.32 to No.40) make up the four categories of the KDD-99 Dataset. Our project employs the KDD-99 open-source dataset to assess the accuracy of the proposed classifier in intrusion detection for classifying normal or assault packets. The dataset contains 4,94,021 network packets and 40 characteristics without considering the outcome of the dataset. There are 21 distinct output or outcome categories with a normal OBUs in transport network, while the remaining 20 cat reflect various problematic connections. 96.89 (19.79 percent) of the total tuples records were normal, 394.458 (78.24 %) Daniel Service, 5.107 (0.79 %) R2L and 51 (0.012 %) U2R attribute, 1.136 (0.22%) R2L filed and 51 (0.011%) U2R attribute (0.01%) Table 1 displays the training name (class representations) and its labels, as well as the number of training samples. Table 2 contains field description of sample data consumed in experiments.

The Canadian Institute for Cyber security's (CIC) Intrusion Detection System (IDS) 2018 [11] is a modern anomaly-based NIDS dataset that was suggested in 2018 and is openly accessible via the Internet upon request from its owners.

The CICIDS2018 dataset consists of benign and up-to-date common assaults, resembling actual real-world data (PCAPs). It also contains the findings of a CICFlowMeter network traffic analysis with labelled flows based on the time stamp, source and destination IP addresses, source and destination ports, protocols, and attack type (CSV files). It contains 16,000 samples distributed evenly as follows: Normal (8 000), Attacks (8,000, including DoS, brute force, port scan, and ping scan). The model is trained and evaluated using KDDCUP 99 and CICIDS2018 datasets. Due to the huge number of datasets, 1,20,223 data were selected for experimentation, and Table 3 provides a summary of the total number of training, testing, and validation datasets.

3. TRAFFIC AND NETWORK SIMULATION

In NS2 2.34, low-density and high-density having 20-300 nodal parts and 10 to 30 intruder points have been shown in Figure 2. A log file is used for recording different attacks. Figure 2 and Table 1 give examples of VANET simulation. Few features are kept in the packets from 2 other sourcing. Table 1 gives the instance of network packets. By converting log files, standard data is used for normalization. It may be noted that simulation distractions as used for normalization. It may be noted that simulation distractions are used for the validation of deep learning models for accurate results. In Ali Alheeti and McDonald-Maier [12], intrusion detection for self-driving vehicles is done by simulation with the latest tools.

4. PROPOSED METHODS

The world of transportation is everything but straightforward. Both spatial and temporal qualities are evident in a wide range of contexts and at varied scales. It might be difficult to describe the interplay of variables, create generalized representations, and then apply such models to a specific issue domain. Modern intelligent transportation systems confront many more challenges than those described above (ITS). A survey of the role modelling techniques used in deep learning is presented in this work, which focuses on the function they've had in ITS as well as on the issue of formulations and architectural and problem-specific considerations utilized by practitioners to build solutions to these challenges. We anticipate that this poll will act as a link between the machine learning and transportation communities, shedding light on potential future topics and concerns.

Suggested methods of DNN for intrusion prevention are shown in Figure 4 in which there are 3 phases: preprocessing, feature extracts, and classification. Totally 41 factors have been considered for the KDD dataset having high potentiality for engaging investigation in the intrusion. The numerical strength of the factors has an important influence in the total framework; decrease helps inaccurate work and reduction of the processing period.

Table 3. Data set for experiment.

Message type	KDD-99		CIC-IDS-2018		VANET-Sim
	Training	Testing	Training	Testing	Validation
Normal	20277	4057	22278	4057	7000
Attack	20459	4091	23913	4091	7000
Total	40736	8148	46191	8148	14000

ID requires three distinct phases: (i) the preprocessing (eliminating unnecessary data) phase, (ii) the feature (more related data) extraction phase, and (iii) the classification (model building) phase.

In this research, the KDD-99 dataset contained 40 features (outcome not considered) for each network safety packet. Those with the most potential to investigate intrusions must be selected from these characteristics. The detection approach and the lowered number of predicted features from the datasets are crucial contributors to this research. Typical traits are also a significant factor. The amount of utilized features plays an important part. Accuracy and processing time are the primary reasons for the decline in the amount of network features.

5. PREPROCESSING

Preprocessing, data filtering, and nomination are taken into consideration. Each packet number is set in the range of 0 to 1 by the following equation for non-direction. By training using proper data, ANN can give the best prediction.

$$Z - \text{score} = \frac{\text{Acutal value (x)} - \text{mean } (\mu)}{\text{Standard deviation (sd)}} \quad (1)$$

In the Equation 1, Z score is normalized in 0-1 range ; x is the dataset with mean μ and SD. These valuations are suitable for higher & lower limits of the activation function of signal variety.

5.1. Feature Extraction

The primary features are selected to increase or enhance accuracy, the precision of the proposed DL-based IPS model, and the number of false (intruder messages) alarms. In the feature selection (eliminating unnecessary) phase [13], a mathematical intelligence is applied to choose key related quality features with a high weight and more influence based on the Position selection approach. In contrast, the removal or eliminating of a few unneeded features improves IDS's detection rate, calculation time, and memory, hence increasing its overall effectiveness. With the addition of 12 features, the necessary time lowers by 11.24% and the required memory decreases by 28.7%.

The salient points of evaluation are the precision of classification and the number of false alarms. Critical analysis helps in greater weight for POS methodology in selecting features [13]. By selecting extra features, the importance of efficiency is possible. For

Additional 13 characteristics, time is reduced by 11.4% and measuring by 27.7%

Model = Seq () (Create sequential hidden layers).

Model = add (Dense (10, input_dim = x.shape [1], activation = ‘ relu’)).

6. BUILD THE MODEL

The dataset was contributed by the MIT Lincoln Labs. As revealed by Lincoln Labs, twenty percent of the collection, comprising 494,021 links, was utilized in this investigation for our training technique. Test set consists of the series of tagged connections, which totals around 4.8 millions. Therefore, it is possible to test a programme for unanticipated connections using the complete dataset. A six-state rule set is constructed to appropriately classify seven distinct attack attribute up to the present development or implementation step. From the two attack groups (the 11% training data set), the top three label distributions for attacks are as follows: DoS and Search. Smurf, Neptune, and land are DOS attack techniques; Satan, ipsweep, and portsweep are examples of attacks.

$$N_{ij} = \sum_{i=1}^{i=2} (X_i * W_{1i}) + b_1 \quad (2)$$

$$N_{11} = \frac{1}{1 + e^{-H_{ij}}} \quad (3)$$

Where N_{ij} is one hidden layer in deep network and N_{11} is Softmax logic X_i is filtered feature input from network traffic and $W_{1i}$is a weighted factor and b_1 is first bias.

The functional nodal logic is shown in Equations 2 and 3.

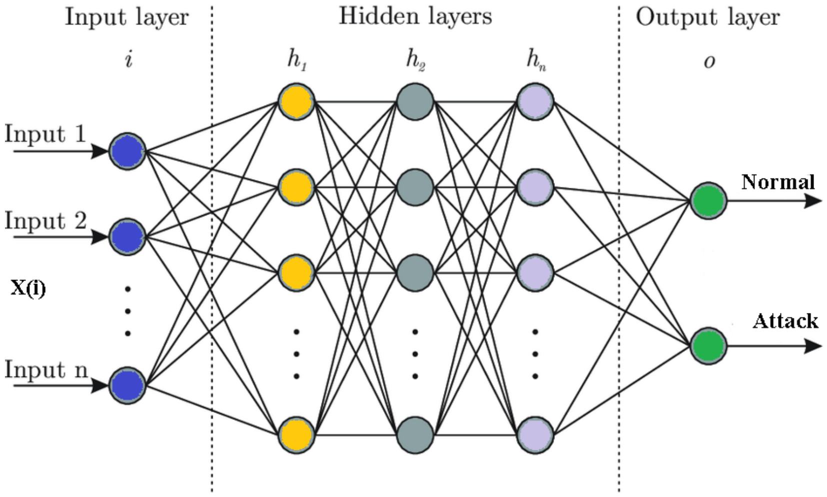

Figure 3. DL model.

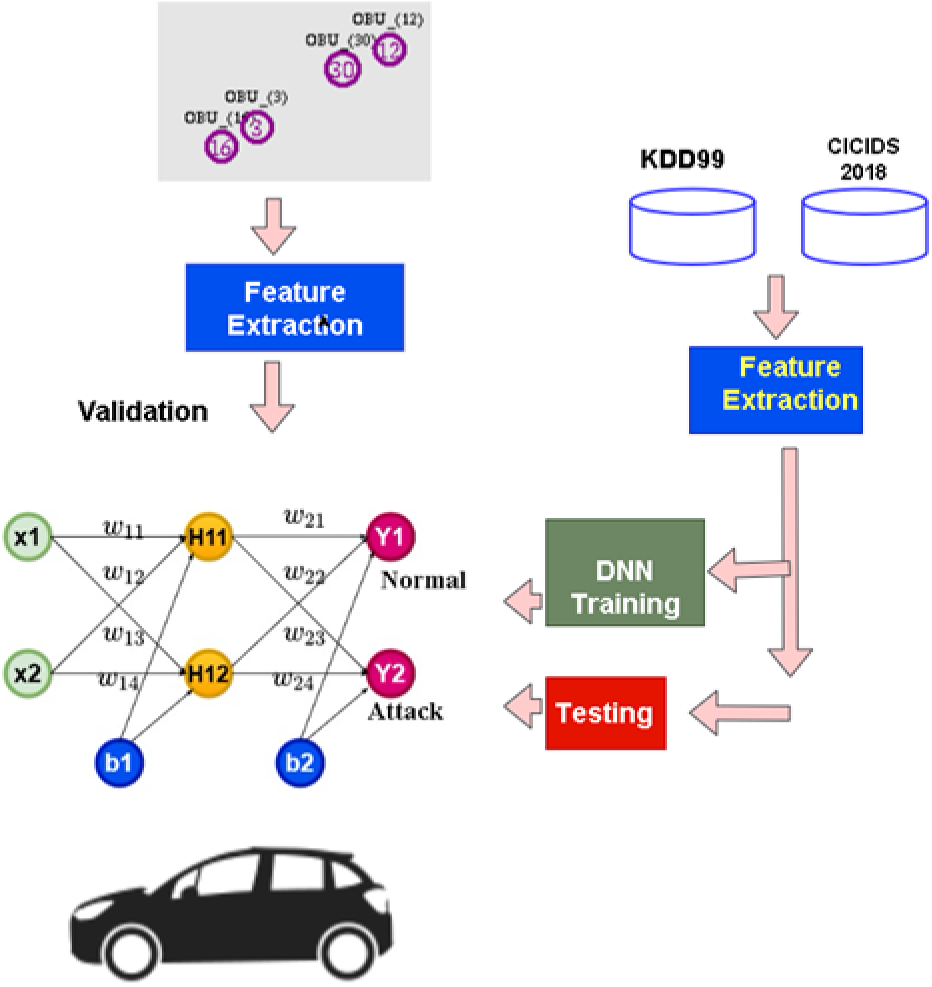

Figure 4. Proposed DL classifier.

6.1. Learning Mechanism

From the 41 features of KDD, the computing features with high potential are selected to engage in the intrusion by protocol. For transforming such features, a unique number is arranged. There are 494,021 limits in the dataset of MIT Lincoln labs. The proposed method has all the labeling of sequential events having 4.9 million connecting for checking rules are used on 6 states for classifying 6 different labels of attacks.

Back propagation used weights based on error.

$$Error_{tot} = \sum_{k=1}^{k=2} \frac{1}{2}(Target - Output) \qquad (4)$$

$$W_{new} = W_{old} - (\alpha * Error_{tot}) \qquad (5)$$

Where α is rate of learning

By computation of errors, weights are updated. It may be noted that the learning rate influences the speed of diagnostics and stable behavior.

For training purposes, eighty percent of the dataset (KDD99& CICIDS-2018) are used while twenty percent are training and rest for testing. Therefore, outcomes are True Normal, True Attack and False Normal & False Attack

For accurate results, the correct state of normal or Attack classification over a total number of packets is necessary.

7. PERFORMANCE EVALUATION

The proposed model was trained and evaluated using the KDD99 dataset generated from a real-time military network traffic environment and the CICIDS 2018 dataset made freely available by the Canadian Institute for Cyber security Research. 80% of both datasets were utilized for training, whereas 20% of the dataset was used to test the model. A network simulator produced a set of data used to validate the suggested model. Four outcomes depict the performance of the suggested binary classifier [14]. The following are four outcomes of model

T_N_(TrueNormal) msg: The capacity to classify normal message as normal.

T_A_(TrueAttachk) msg: Ability to predict incoming attack message as an attack.

F_N_msg: The capacity to incorrectly forecast attack message as normal.

F_A_msg: The capacity to incorrectly identify normal message as attack packets.

Table 4. Confusion table.

Model Prediction			
		1	0
Ground Truth	Attack	True Attack	False Normal
	Normal	False Attack	True Normal

The correct states of classifying mechanism for the packets with respect to the total number of packets give the accuracy. The exact relation regarding accuracy is given by Equation6. Table 4 shows how the confusion matrix records the disadvantages of accuracy tests.

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{FP}+\text{TN}+\text{FN})} \qquad (6)$$

It may be noted that the predicted values can be classified as results: true attack, true normal, false attack & false normal. The tests of accuracy can distinguish between normal and attack packets by Equation 6. Moreover, from the total attacks, the number of attacked packets is given by Equation 7.

$$\text{Recall} = \frac{(\text{TP})}{(\text{TP}+\text{FN})} \qquad (7)$$

The recall gives a level of integrity such that the labeled attack rate for the samples is known. Furthermore, sensitivity gives the number of predicted attacks from the total by the Equation 8:

$$\text{Precision} = \frac{(\text{TP})}{(\text{TP}+\text{FP})} \qquad (8)$$

Equation 9 gives the F1 score which is the harmonic mean of the indicators of precision and recall. It decides the best model for true & false cases.

$$F_1score = \frac{2\ X\ Precision\ x\ Recall}{Precision+Recall} \qquad (9)$$

It is worthwhile to note that the performance matrices can evaluate the efficacy of the proposed model the for classification of packets in terms of normal & attack.

7.1. Novelty of Work

- Vehicular Network with real-time traffic simulated using SUMO and NS2 open source tools and output of simulation are fed to the proposed model for validation of results and stability of model.
- Build intelligent model which trains and tests using standard data set such as KDD99 and CICIDS 2018 which will increase accuracy of proposed model.
- Proposed model validated with VANET simulated dataset which increases the reliability of model performance.
- Figure 4 shows the proposed deep learning model which trains, tests and validates using three different data sets include KDD99, CICIDS 2018 and experimental data and compared with standard and advanced approach

8. RESULTS AND DISCUSSION

One of the most often used models today is convolutional neural networks (CNN). This neural network computational model has one or more convolutional layers that can either be completely connected or pooled. It is based on a variant of multi-layer perceptions. When an image is divided into rectangles and sent for nonlinear processing, the feature maps produced by these convolutional layers serve as a record of the picture's region. CNN does not encode the position and orientation of images objects and its inability to spatially differentiate the input datasets. It requires lots of training datasets (around 20 lakshs of accidents data) to increase the accuracy as well complexity and computation time. In real time, it is highly difficult to get more data. That's why its performance low of around 91.44% accuracy and remaining performance metrics along with comparison are shown in the diagram.

Complexity increases with recurrent neural networks (RNN). They record the processing node output and incorporate the outcome back into the model. The model is said to learn to predict a layer's outcome in this way. In the RNN model, each node functions as a memory cell, continuing calculations and operation implementations. When a network makes an inaccurate prediction, the system self-learns and keeps trying to make the right forecast through back propagation. RNN model takes lot of time, similar to CNN, to learn the model. It is limited to relu and tanh activation functions or perform better with above activations. It performs better for short range sequences and consumes lot of time. LSTM RNN somewhat better performance when compared with GRU RNN. Accuracy increases by 0.21% due to the large training data.

The Dataset contains 1,20,223 data points and 41 attributes. The data comprises 23 distinct output classes with the normal class representing a suitable communication link [15] and the remaining 21 classes representing various forms of poor connections. Approximately 60.33 percent of the data points belong to the 'natural' group (excellent relationships). Class "Neptune." (35.594%) and "back" are the categories with the highest percentage of poor connections (0.665%). The rootkit classes 'load module.' 'FTP write,' and'multihop' have fewer than ten data points per class, with “spy” having the fewest. This dataset is somewhat unequal. Consequently, it is necessary to develop a model that appropriately categorizes the points that belong to these various groups. Figure 6 depicts the evaluation comparison [16] between the new model and the existing model in terms of accuracy, precision, recall, and F1-score. On the basis of KDD-CUP 99 statistics, the precision rate in this article is 99.57 percent higher than that of 98.62 percent of CNN, and 78.24 percent higher than that of literature.

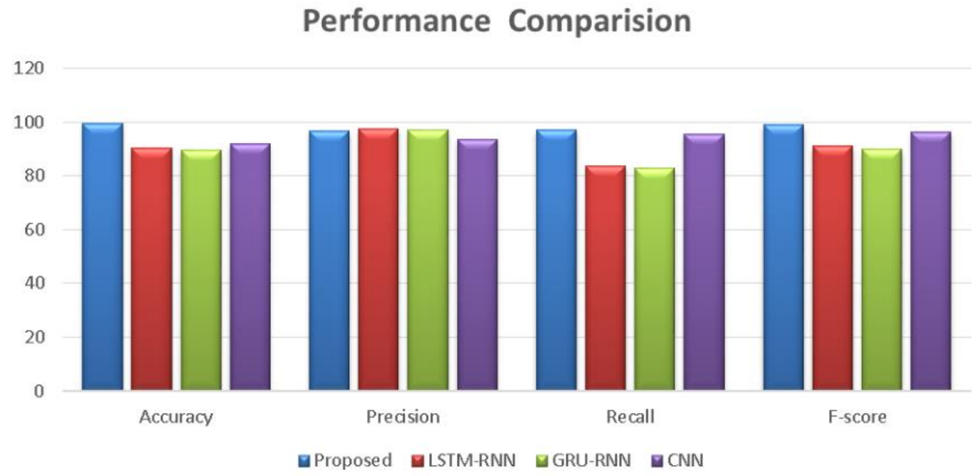

Figure 5. Performance comparison of proposed DL based IPS classifier.

Figure 5 gives the outcomes of various methods used in comparison for verifying the performance of the IDS model in case of intrusion data. Reasonable determination of indicators gives the efficiency. For higher values of indicators, the FAR is lower, giving higher efficiency. For precision and alert in the case of ideal classification, a hit is 1, and FAR is 0. Figure 6 shows the measure of accuracy of DL-based IPS model compared with CNN and RNN. The proposed DL improves its accuracy as it increases the number of training layers. After a depth of 12 layers, both proposed DLIPS and CNN performance 100 % accuracy whereas RNN degrades in accuracy due to reaching saturation of training.

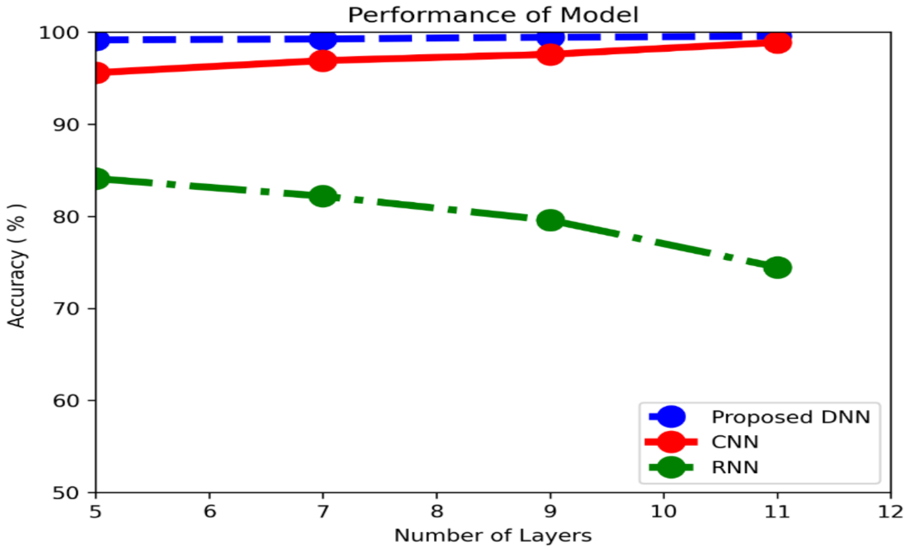

Figure 6. Measure the accuracy of IPS classifier.

9. CONCLUSION

The proposed model gives safe transmission of packets in the Internet of Vehicles based on KDD 99 & CICIDS 2018, deep neural networks-based classifier. Utilizing a supervised pre-training technique of deep belief networking with probability-related vectors and conventional stochastically operating gradient descent helped in the extraction of packets. Further, training was done by the above-mentioned data sets. Validation was done by the NS2 network simulator. Every class for the probability of normal / attack packets is distinguished by DNN for recognition of malicious types. The evaluation matrices were compared with CNN and RNN models. For future work, more data may be used for training the models for increasing the accuracy in real-time. Further, additional features may be given customization or filtering to reduce the time of training.

Funding: This study received no specific financial support.
Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study.

REFERENCES

[1] V. Praneeth, K. Kumar, and N. Karyemsetty, "Security: Intrusion prevention system using deep learning on the internet of vehicles," *International Journal of Safety and Security Engineering*, vol. 11, pp. 231-237, 2021.Available at: https://doi.org/10.18280/ijsse.110303.

[2] M. El Boujnouni and M. Jedra, "New intrusion detection system based on support vector domain description with information gain metric," *International Journal of Network Security*, vol. 20, pp. 25-34, 2018.

[3] Network Simulator, "Network simulator. Retrieved from: https://www.isi.edu/nsnam/ns/. [Accessed 2022]," 2021.

[4] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575-2582.

[5] Deep Learning basics, "Deep learning basics. Retrieved from: https://www.v7labs.com/blog/deep-learning-guide. [Accessed 2022]," 2021.

[6] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, pp. 2352-2449, 2017.Available at: https://doi.org/10.1162/neco_a_00990.

[7] H. S. Dong, K. K. An, and S. C. Choi, "Malicious traffic detection using Kmeans," *Journal of Korean Institute of Communications and Information Science*, vol. 41, pp. 277-284, 2018.Available at: https://doi.org/10.7840/KICS.2016.41.2.277.

[8] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 534-545, 2015.Available at: https://doi.org/10.1109/TITS.2014.2320605.

[9] Y. Wang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing*, 2017, pp. 635-638.

[10] S. Hettich and S. D. Bay, "The UCI KDD Irvine, CA: The university of California, department of information and computer science. Retrieved from: https://kdd.ics.uci.edu/," 1999.

[11] Canadian Institute for Cybersecurity IDS, "Canadian institute for cybersecurity IDS dataset. Retrieved from: https://www.unb.ca/cic/datasets/ids-2018.html," 2021.

[12] K. M. Ali Alheeti and K. McDonald-Maier, "Hybrid intrusion detection in connected self-driving vehicles," in *22nd International Conference on Automation and Computing*, 2016, pp. 456-461.

[13] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210-42219, 2019.Available at: https://doi.org/10.1109/access.2019.2904620.

[14] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017.Available at: https://doi.org/10.1109/ACCESS.2017.2762418.

[15] K. Nagarjuna and K. R. Kumar, "Road safety: An accident prevention using intelligent vehicular network," *International Journal of Safety and Security Engineering*, vol. 10, pp. 631-638, 2020.Available at: https://doi.org/10.18280/ijsse.100507.

[16] N. Karyemsetty, B. Samatha, and K. H. Rao, "Design and deployment of vehicle tracking system in VANETs using Xbee Pro: Prototype model," in *2015 International Conference on Communication Networks*, 2015, pp. 97-100.

Views and opinions expressed in this article are the views and opinions of the author(s), Review of Computer Engineering Research shall not be responsible or answerable for any loss, damage or liability etc. caused in relation to/arising out of the use of the content.