Comparative Study of HDL algorithms for Intrusion Detection System in Internet of Vehicles

Manoj Srinivas Botla, Jai Bala Srujan Melam, Raja Stuthi Paul Pedapati, Srijanee Mookherji[0000−0001−8650−2463], Vanga Odelu[0000−0001−6903−0361], and Rajendra Prasath[0000−0002−0826−847X]

Computer Science and Engineering Group, Indian Institute of Information Technology Sri City, Chittoor, 630 Gnan Marg, Sri City - 517646, Andhra Pradesh, India
{manojsrinivas.b19, jaibalasrujan.m19, rajastuthipaul.p19, srijanee.mookherji, odelu.vanga, rajendra.prasath}@iiits.in

Abstract. Internet of vehicles (IoV) has brought technological revolution in the fields of intelligent transport system and smart cities. With the rise in self-driven cars and AI managed traffic system, threats to such systems have increased significantly. There is an immediate need to mitigate such attacks and ensure security, trust and privacy. Any malfunctioning or misbehaviour in an IoV based system can lead to fatal accidents. This is because IoV based systems are sensitive in nature involving human lives either on or off the roads. Any compromise to such systems can affect user safety and incur in service delays. For IoV users, the Intrusion Detection System (IDS) is crucial to protect them from different malware-based attacks and to ensure the security of users and infrastructures. Machine Learning approaches are used for extracting useful features from network traffic and also for predicting the patterns of anomalous activities. We use two datasets, namely *B*alanced DDoS dataset and *Car-Hacking Dataset* for comparative study of intrusion detection using various machine learning approaches. The comparative study shows the differences of various machine learning and deep learning approaches against two datasets.

Keywords: Intrusion Detection Systems, Internet of Vehicles, Machine Learning, Deep Learning, Hybrid Deep Learning

1 Introduction

An Intrusion Detection System (IDS) is a device or software programme that monitors all network traffic and notifies the user or administrator when unauthorised attempts at or accesses are detected. The Internet of Vehicles (IoV) is a network application that links smart vehicles to the internet and vehicles to each other via the Internet of Things (IoT).The IoV network is divided into two sub networks: *intra-vehicular* and *inter-vehicular*. Attackers have the ability to launch a Distributed Denial of Service (DDoS) attacks that disable the CAN bus and prevent IoV-based vehicles from accessing the brakes or any critical parts of the vehicle at critical moments. DDoS attacks on inter-vehicle networks keep the channels busy, thereby resulting in preventing signalling lights in hazardous areas from turning into red instead of keeping them green.

This could eventually cause accidents. DoS, fuzzy, and spoofing attacks are some other of the types of attacks in the IoV networks. These attacks could potentially cause priority vehicles to experience service delays and may possibly result in accidents.

The Distributed Denial of Service (DDoS) attack, also known as a brute-force attack, is a major threat to vehicles because it overwhelms their cache and computing power. Therefore, intrusion detection techniques have drawn a lot of attention in IoV in order to protect user's security and privacy. IDS is required in IoV to prevent false emergency reports and false congestion reports. Automated detection systems powered by Machine Learning (ML) techniques deliver an impressive performance. Moreover, ML techniques have good capabilities to detect unknown attacks. ML algorithms are simple to train and can handle non-linearity in the data. To identify malicious attacks in IoV, an efficient and faster algorithm is required. In recent days, Deep learning (DL) algorithms offer more effective performances than traditional ML algorithms.

In Internet of Vehicles (IoV), Some cars are linked for extended periods of time, making it difficult for traditional ML models to predict long-term outcomes. Every DL algorithm needs multiple layers of layers for the improved performance. By using an Hybrid Deep Learning, we can benefit from the advantages of every algorithm and perform more effectively.

This comparative study is organized as follows: Section 2 describes the review of literature on machine learning, deep learning and Hybrid DL algorithms for intrusion detection problem. In Section 3, we present various machine learning / deep learning classifiers; Section 5 describes the comparative analysis of ML / DL algorithms on various datasets. Finally Section 6 concludes the paper.

2 Review of Literature

The most recent research on intrusion detection systems for Internet of Vehicles is presented in this section. Rohit and Amit [4] presented a machine learning based IDS using PART(Partial Decision Tree) to counter passive and active attacks in both Network-IDS and Host-IDS. The KDDcup99 dataset was used to evaluate this model.

Jing and Chen [6] proposed a Support Vector Machine based IDS with a new scaling method which detects intrusions that lacks in KDDcup99 dataset. The testing accuracy of this model was 6.17% better than Naive Bayes approach. The UNSW-NB15 dataset was used to evaluate this model. In another work, Ayesha and Mourad Elhadef [2] found that V2V communication comprises specific vulnerabilities which can be manipulated by attackers to compromise the whole network. They proposed a DL-based IDS using MLP(multilayer perceptron) neural network to detect intruder on VANET or an IoV network. The KDD Cup 1999 dataset was used to evaluate the model. Nie *et* al. [9] designed a data-driven IDS by analyzing the link load behavior of Road Side Unit (RSU) in IoV against various attacks using the Convolution Neural Network (CNN) to extract features and detect intrusions. The network traffic dataset was used to evaluate their proposed model.

Traditional IDS explode when dealt with extremely large amount of vehicular data. Tejasvi and Varun [1] proposed an Artificial Intelligence based IDS using CNN-LSTM which is a combination of CNN and Long Short Term Memory (LSTM) model. Two

CNN layers and one LSTM layer with the ReLU activation are present in the model. The VeReMi Extension termed dataset, which was created using the VEINS simulation tool, was used to evaluate this model.

Safi and Muazzam [12] proposed a Hybrid Deep Learning (HDL) model for cyber attack detection in IoV. The proposed model is built using LSTM and GRU(Gated Recurrent Unit). Two datasets—the combined DDoS dataset and the car-hacking dataset—are used to analyse the performance of the model. LiYang and Abdallah [13] proposed a multitiered hybrid IDS that incorporates an anamoly-based IDS based on CL-k-means with a signature-based IDS based on DT(Decision Tree), RF(Random Forest), ETs(extra trees), and XGBoost(Extreme Gradient Boosting). On the CAN-intrusion-dataset and CICIDS2017 dataset, the proposed model was evaluated.

A comparative study of the related works is presented in Table. 1

	Works	**Algorithm Used**	**Datasets Used**	**Type of Classification**	**Accuracy**
ML	R.K.S.Gautam and E.A.Doegar (2018) [4]	PART(Partial Decision Tree)	KDDcup99	multi class	99.95
	D.Jing and H.B.Chen (2019) [6]	SVM(Support Vector Machine)	UNSW-NB15	binary & multi class	binary - 85.99 & multi - 75.77
DL	A.Anzer and M.Elhadef (2018) [2]	MLP(Multi-layer Perceptron)	KDD Cup 1999	multi class	98.49
	L.Nie *et* al. (2020) [9]	CNN(Convolutional Neural Network)	Network Traffic dataset	binary class	97.60
HLD	T.Alladi *et* al. (2021) [1]	CNN-LSTM(Long Short Term Memory)	dataset termed VeReMi extension	multi class	99.42
	S.Ullah *et* al. (2022) [12]	LSTM-GRU(Gated Recurrent Unit)	combined DDoS dataset & car hacking dataset	binary & multi class	combined DDoS - 99.85 car hacking - 99.99
	L.Yang *et* al. (2021) [13]	XGBoost with CL-k-means	CAN-intrusion-dataset & CICIDS2017	multi class	CAN-intrusion - 99.99 CICIDS2017 - 99.88

Table 1. Comparative study of the existing IDS using ML and DL algorithms

3 ML / DL Classifiers

In this section, we describe various ML / DL / Hybrid DL algorithms, that were used in the comparative analysis [12,11], including $k-$Nearest Neighbour, Logistic Regression, Random Forest, Support Vector Machine, Long Short Term Memory, Gated Recurrent Unit, Multi-Layer Perceptron, Convolutional Neural Network, and LSTM-GRU Hybrid Model;

3.1 $k-$Nearest Neighbour

The $k-$nearest neighbors algorithm, abbreviated as $k-$NN, is a supervised learning classifier that uses proximity to perform classification or predictions about grouping the given set of data points. This non-parametric classifier works on the assumption that similar points can be found in a close proximity. This memory-based learning algorithm heavily depends on memory to store all training data and as the dataset size grows, this algorithm becomes increasingly inefficient in terms of overall performance.

3.2 Logistic Regression

The logistic regression is a statistical model that estimates the probability of an event (such as does belong to the class or not) based on a set of independent variables. Linear regression is an approach for modelling the relationship between a scalar response (dependent variables) and one or more explanatory variables (independent variables). In linear regression, linear predictive functions are used to model the unknown parameters using the relationships estimated from the data. Conditional mean of the responses is assumed to be an affine function provided the values of the predictors are given. This type of regression approach mainly focuses on the conditional probability distribution of the responses given the values of the predictors. Depends on the number of explanatory variables, we classify the regression approach as either a simple linear regression or multiple linear regression.

3.3 Random Forest

A Random Forest is consisting of several tree-structured classifiers in which each tree votes for the most popular class at the given input. The training set is used to grow the tree such that for the k^{th} tree, a random feature vector is generated and a tree is grown using the training set and the random vector. After generating a large number of trees, the features are used in the voting process for binary prediction of the labels (it can either be an entity or not an entity). This outcome is then used as a feature in the next step in which the Conditional Random Field model performs the classification of the entities [3].

3.4 Support Vector Machines

Support Vector Machine(SVM) is a class of learning algorithm under supervised learning setup. The main purpose of this algorithm is to explore a hyperplane in an n dimensional space that distinctly classifies the labelled instances. The number of features determine the dimensions of the underlying hyperplane. The choice of the hyperplane is crucial to represent the maximum separation between two classes. Thus this approach is also called as a maximal margin classifier that transforms low dimensional input space into higher dimensional space so as to convert non-separable instances into linearly separable instances of the input space. For this purpose, different kernel functions may be used for decision functions and its variations to specify custom kernels.

3.5 Long Short Term Memory (LSTM)

Long short-term memory (LSTM) is a neural network that has feedback connections enabling a recurrent neural network architecture that processes not only single data points but also entire sequences of data. The common weights and biases change once per episode of training (once per time-step). The LSTM architecture mainly focuses on providing a short-term memory for recurrent neural networks that last for thousands of time-steps. This architecture consists of a cell, an input gate, an output gate, and a forget gate [14]. This cell remembers values over an arbitrary time intervals and regulates the sequential flow of information. LSTM networks may also suffer from exploding the gradient problem.

3.6 Gated Recurrent Unit (GRU)

In order to solve the vanishing gradients problem encountered during the operation of a recurrent neural network, the Long Short Term Memory Network is proposed. Another variations of the recurrent network is the Gated Recurrent Unit Network(GRU) [5]. GRU consists of three gates namely Update Gate, Reset Gate, and Current Memory Gate and does not maintain an internal cell state. The work flow of a GRU network is similar to the RNNs and the primary difference is the internal working within each recurrent unit as GRUs consist of gates that modulate the current input and the previous hidden state.

3.7 Multi-Layer Perceptron

Multi layer perceptron (MLP) is a feed forward neural network that consists of three types of layers - the input layer, output layers and hidden layer. MLPs are models that perform as universal approximators. A number of hidden layers is placed between the input and output layer and the data flows in the MLP in the forward direction from input to output layer. Back propagation learning algorithm is used by the neurons in the MLP that approximate any continuous function for solving problems that are not linearly separable.

3.8 Convolutional Neural Network

Convolutional Neural Networks have superior performance over the traditional neural networks and consists of three main layers namely, Convolutional layer, Pooling layer, Fully-connected (FC) layer. Majority of the computations take places in the convolutional layer and the convolution involves a kernel moving across the receptive fields of the input data. Over multiple iterations, a sequence of learning generates a feature map allowing the CNN to interpret the relevant portion of the data efficiently. The pooling layer focuses on reducing the number of parameters in the input thereby reducing the computational complexity. In the fully-connected layer, each node in the output layer is connected to a node in the previous layer. This layer performs the classification based on the feature map that was generated in the previous layers. Different activation functions can be used in the convolution and pooling layer and a softmax activation function is used in the fully connected layer to classify the inputs appropriately

3.9 LSTM-GRU Hybrid Model

The LSTM-CGU model is an hybrid model that exploits not only the characteristics and learning capabilities, but also the strength of both GRU and LSTM models [15], so as to produce a more accurate and reliable predictions of the data set. In this integrated model, the output of the LSTM model is fed into the GRU model in order to produce a single, final output as it is being concatenated and formed a fully-connected layer.

4 Implementation challenges

Implementation of LSTM: The designed architecture is made up of 100 LSTM layers and one output Dense layer. The output is produced using batch size 32 and six epochs. We obtained 0.999 accuracy for the car hacking dataset and 0.492 accuracy for the distributed DDoS dataset under these conditions.

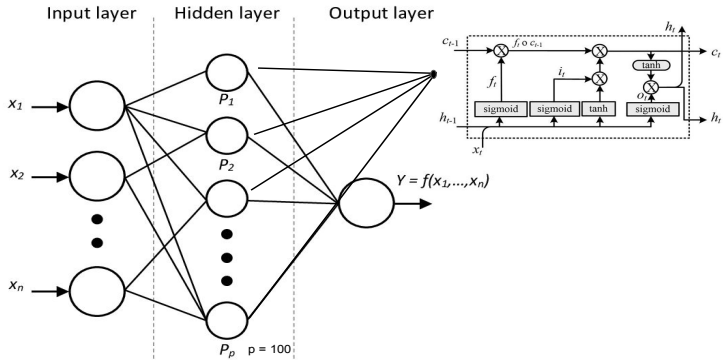

Fig. 1. LSTM Architecture

Implementation of GRU: The designed architecture is made up of 100 GRU layers and one output Dense layer. The output is produced using batch size 32 and six epochs. In these conditions, we obtained 0.999 accuracy for the distributed DDoS dataset and 0.992 accuracy for the car hacking dataset.

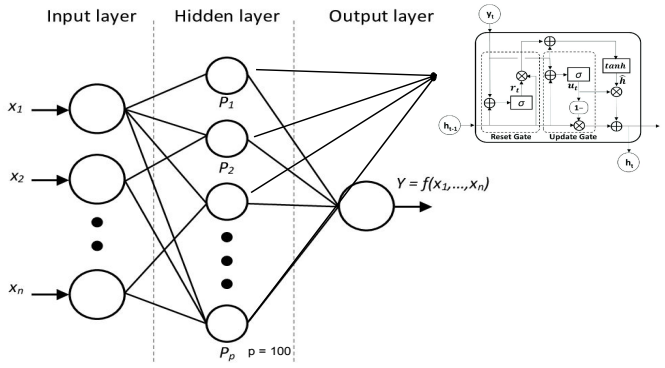

Fig. 2. GRU Architecture

Implementation of CNN: The designed architecture is made up of 2 hidden layers (1 convolutional layer and 1 pooling layer). A sigmoid activation function and 100 filters with a kernel size of 2 are present in the 1D convolutional layer. The pool size for

the 1D average pooling layer is 2. Accuracy metrics and the Adam optimizer are used to build the sequential model. The output is produced using batch size 32 and epochs 6. The input of shape (4,1) is provided to the model for the car hacking dataset. the output shape becomes (3,1) after convolution, then after average pooling the output shape becomes (2,1). The model receives the input of shape (70,1) for the distributed DDoS dataset. After convolution, the output shape is (69,1), and following average pooling, it is (68,1). We obtained 0.79 accuracy for the car hacking dataset and 0.50 accuracy for the distributed DDoS dataset under these conditions.

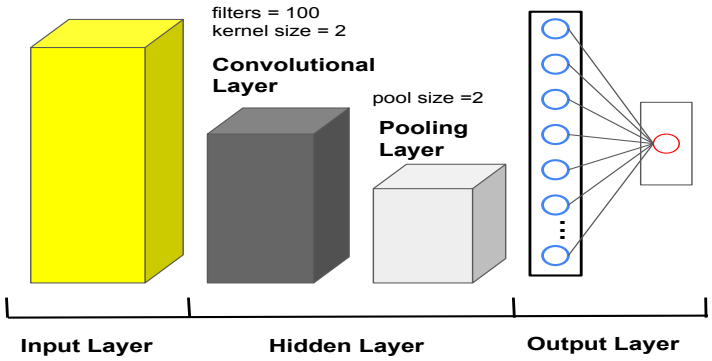

Fig. 3. CNN Architecture

Implementation of LSTM-GRU Hybrid Model: This hybrid model is made up of three layers: LSTM, DENSE, and GRU. The initial hidden layer is LSTM. DENSE is the second layer, connecting LSTM and GRU. GRU is the third layer, which accepts values from the prior DENSE layer and provides the final output.

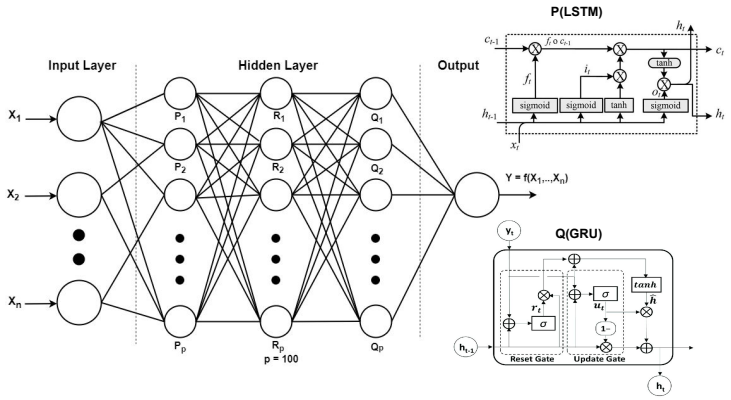

Fig. 4. LSTM-GRU Hybrid Model Architecture

5 Experimental Results

The experimental environment is a PC with a Windows 11 64-bit operating system, an i5-9300H processor clocked at 2.40 GHz, 8 GB of RAM, and Python 3 (version 3.10.4) installed. The free edition of Google Colab can also run code. There are 84 features and 12794627 total data-points (DDoS + Normal) in the balanced DDoS dataset (6.79GB in size). Each data point represents the actual flow (either forward or reverse). The Car-Hacking dataset has 12 features and is 818 MB in size. It includes DoS attacks, fuzzy attacks, drive gear spoofing, and RPM gauge spoofing attacks. scikit-learn[1], keras[2] and tensorflow[3] libraries are used for ML, DL and Hybrid DL. dask API[4] library is used for parallel processing. pandas[5] library used to read csv(dataset) files.

5.1 Datasets

In this section, we describe different datasets used for Intrusion Detection problems.

KDDcup99 Dataset KDDcup99[6] was generated based on intrusion detection analysis software by the Defense developed research undertaking corporation DARPA. They created a computer network simulation that was used to represent a normal environment that has been compromised by several different types of attacks. There are 24 attack patterns in all, and they are divided into 4 classes. There are Denial of Service(DoS), User to Root Attack(U2R), Remote to Local Attack(R2L) and Probing Attack(PROBE).

UNSW-NB15 Dataset There are nine attack classes namely, Normal Class and Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms are included in the UNSW-NB15 dataset. This dataset contains a large number of new attacks on existing networks. The UNSW-NB15 dataset[7] has 257,673 data instances, comprising of 82,332 testing data instances and 175,341 training data instances. To ensure the reliability of the Network Intrusion Detection System evaluations, there is no redundancy in the data in the training and testing dataset. Each data has 44 features. The matched features are classified into 6 groups including flow, basic, content, time, additional generated and labelled.

Network Traffic Dataset A testbed is built, using one road side Unit, thirty on-board Units, and four attackers, to mimic the Internet of Vehicles environment (IoV). The testbed also has a Low Orbit Ion Cannon (LOIC) mounted for the DDoS attack. The LOIC carries out the DDoS attack by making a number of malicious requests with

[1] scikit-learn-https://scikit-learn.org/stable/
[2] Keras-https://keras.io/api/
[3] Tensorflow-https://www.tensorflow.org/
[4] daskAPI-https://www.dask.org/
[5] Pandas-https://pandas.pydata.org/docs/
[6] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[7] https://research.unsw.edu.au/projects/unsw-nb15-dataset

regard to the particular protocols TCP, UDP, and HTTP. As in a traditional DDoS attack, an IoV's cost-effective network intrusion targets the OBU rather than the server since it is far simpler to drain OBU resources than it is to do so for the server. Wireshark was used to sample the packets sent across the RSU link.

VeReMi Extension Dataset The VeReMi extension dataset[8] was generated with the open-source simulation tool VEINS. This dataset consists of log files with the traces gathered from each car that travelled over the network during the course of 24-hour period. Each data point, representing a message sent by a moving vehicle in the network, has a variety of fields, including a timestamp, a vehicle's pseudo-identity, coordinates for its position, velocity, acceleration, and direction. It has 10 classes, including 1 normal class, and 9 attack classes such as disruptive, data replay, and DoS attacks, and the rest of the classes are different possible combinations of the four primary attack types.

CombinedDDoS Dataset The CombinedDDoS dataset[9] was created by combining the real-time network DDoS datasets from CIC DoS 2016, CICIDS 2017, and CSE-CIC-IDS 2018. DDoS and Benign data types are used to produce this dataset. This collection consists of 6,472,647 entries related to DDoS attacks and 6,321,980 records related to benign activity. Inter-vehicle network traffic statistics according to DDoS assaults were included in the CIC DoS 2016, CICIDS 2017, and CSE-CIC-IDS 2018 datasets. The slowbody2, ddosim, goldeneye, hulk, slowloris, rudy, and slowread assaults were included in the CIC DoS 2016 dataset. DDoS-LOIC and port scan attacks were included in the CICIDS 2017 dataset. DDoS assaults of the SlowHTTPTest, Hulk, Slowloris, and LOIC kinds were included in the CSE-CIC-IDS 2018 dataset. The various DDoS attack types were included in a collection of these datasets that had identical features.

CAN-intrusion Dataset This CAN-intrusion dataset is also called as Car-Hacking Dataset[10]. The car-hacking dataset was created to identify cyberattacks on the car's internal network. This dataset mostly focuses on the CAN bus that can be used to identify an attacker. DDoS, Fuzzy, gear, and RPM are four different files and gear and RPM are spoof attack files. The classes are represented by benign (R) and malicious (T) values in each file of the dataset. Timestamp, CAN ID, DLC, DATA[0], DATA[1], DATA[2], DATA[3], DATA[4], DATA[5], DATA[6], DATA[7], Flag are the attributes of the dataset. This dataset was constructed by logging CAN traffic via the OBD-II port from a real vehicle while message injection attacks were performed.

Attack Type	No. of messages	No. of normal messages	No. of injected messages
DoS Attack	3,665,771	3,078,250	587,521
Fuzzy Attack	3,838,860	3,347,014	491,847
Spoofing the drive gear	4,443,142	3,845,890	597,252
Spoofing the RPM gauze	4,621,702	3,966,805	654,897

Table 2. Different Types of attacks and statistics of normal and injected messages

[8] https://github.com/josephkamel/VeReMi-Dataset
[9] https://www.kaggle.com/datasets/devendra416/ddos-datasets
[10] https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset

CICIDS2017 Dataset The Canadian Institute for Cybersecurity has created the dataset - CICIDS2017 Dataset[11]. By defining two different types of profiles they offered a unique systematic method for producing a reliable dataset. The dataset includes several recent multi-stage attacks, such as Heartbleed and various DoS and DDoS attack variants. A number of modern protocols are also included. It can be easily imported into machine learning tools because it is in CSV format and includes 80 features for each Netflow record.

5.2 Preprocessing

Preprocessing in the DDoS dataset [12] involves removing columns with only one value, categorical columns with a predominance of one category, columns with more than 50% of the values missing, rows with no more than 5% of the values missing in a column, replacing NAN values with infinity, and converting the IP address to an integer. In preprocessing Car-Hacking Dataset [11] string columns are converted to integer columns and data value byte columns are converted to one integer value.

5.3 Evaluation Metrics

The effectiveness of the classification algorithms in terms of their accuracy in predicting the instances correctly is measured by the following metrics [8,10,7]: *Precision*, *Recall* and *Accuracy*.

The evaluation of a classifier measures its ability to predict the right classification decisions based-on a 2 x 2 contingency matrix given in Table. 3. Here, TP_i is true

		Classifier Judgments	
		Yes	No
Expert Judgments	Yes	TP_i	FN_i
	No	FP_i	TN_i

Table 3. The Contingency Table for a category c_i

positives with respect to c_i; FP_i is false positives wrt c_i; FN_i is false negatives; and TN_i is true negatives.

Precision (P_i): Precision (P_i) is defined as follows:

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

Recall (R_i): Recall (R_i) is defined as follows:

$$R_i = \frac{TP_i}{TP_i + FN_i}$$

[11] https://www.unb.ca/cic/datasets/ids-2017.html

Accuracy (Acc_i): Accuracy (Acc_i) of various ML, DL and HDL classification algorithms is measured as follows:

$$Acc_i = \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i}$$

Error (Err_i): This is used to estimate the classification error of various ML, DL and HDL classification algorithms and this can also be estimated as (1 - $accuracy_i$):

$$Err_i = \frac{FP_i + FN_i}{TP_i + FP_i + FN_i + TN_i}$$

5.4 Results and Discussions

Table 4 shows the performance (Accuracy) comparison of various machine learning, deep learning and HDL algorithms implemented on the CombinedDDoS Dataset.

	Classification Algorithms	Accuracy
ML	K Nearest Neighbor	99.71
	Logistic Regression	89.25
	Random Forest	99.07
	Support Vector Machine	94.93
DL	Long Short Term Memory	99.36
	Gated Recurrent Unit	99.992
	Multi-Layer Perceptron	93.743
	Convolutional Neural Network [9]	50.741
HDL	LSTM-GRU Hybrid Model [12]	50.157
	CNN-LSTM Hybrid Model	99.912
	CNN-GRU Hybrid Model	98.278

Table 4. Comparative Study of ML, DL and HDL algorithms implemented on the Combined-DDoS Dataset

The comparative analysis shows that the hybrid deep learning algorithms hardly perform better when compared with the classical machine learning classification algorithms. More specifically, both CNN and LSTM-GRU hybrid models are performing very poor on the combined DDoS dataset due to the fact that the network traffic statistics according to DDoS assaults / attacks have identical features. This is being investigated further to explore the false positive rates / false negative rates.

Table 5 shows the performance (Accuracy) comparison of various machine learning, deep learning and HDL algorithms implemented on the Car-hacking Dataset.

6 Conclusion and Future Work

This paper presents a comparative analysis of various classification algorithms used for intrusion detection system in Internet of Vehicles scenarios. We have considered Machine Learning approaches, deep learning approaches and Hybrid deep learning approaches for this comparative analysis on different datasets. We have extracted useful

	Classification Algorithms	Accuracy
ML	K Nearest Neighbor	97.48
	Logistic Regression	88.63
	Random Forest	98.50
	Support Vector Machine	93.79
DL	Long Short Term Memory	99.97
	Gated Recurrent Unit	99.205
	Multi-Layer Perceptron	88.959
	Convolutional Neural Network [9]	81.891
HDL	LSTM-GRU Hybrid Model [12]	99.219
	CNN-LSTM Hybrid Model	95.000
	CNN-GRU Hybrid Model	96.080

Table 5. Comparative Study of ML, DL and HDL algorithms implemented on the Car-Hacking Dataset

features from network traffic dataset and also for predicting the patterns of anomalous activities. More specifically, we have used two datasets, namely *B*alanced DDoS dataset and *Car-Hacking Dataset* for comparative study of intrusion detection using various machine learning approaches. The comparative study shows the differences of various machine learning and deep learning approaches against two datasets. Subsequently, we plan to apply CNN-LSTM Hydrid Deep Learning [13,1] and the following combinations: CNN-GRU Hybrid Deep Learning. MLP-CNN, MLP-LSTM, and MLP-GRU approaches.

References

1. Alladi, T., Kohli, V., Chamola, V., Yu, F.R., Guizani, M.: Artificial intelligence (ai)-empowered intrusion detection architecture for the internet of vehicles. IEEE Wireless Communications **28**(3), 144–149 (2021)
2. Anzer, A., Elhadef, M.: A multilayer perceptron-based distributed intrusion detection system for internet of vehicles. In: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC). pp. 438–445. IEEE (2018)
3. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
4. Gautam, R.K.S., Doegar, E.A.: An ensemble approach for intrusion detection system using machine learning algorithms. In: 2018 8th International conference on cloud computing, data science & engineering (confluence). pp. 14–15. IEEE (2018)
5. Gulli, A., Pal, S.: Deep Learning with Keras. Packt Publishing (2017)
6. Jing, D., Chen, H.B.: Svm based network intrusion detection for the unsw-nb15 dataset. In: 2019 IEEE 13th international conference on ASIC (ASICON). pp. 1–4. IEEE (2019)
7. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, USA (2008)
8. Mitchell, T.M.: Machine Learning. McGraw-Hill, Inc., USA, 1 edn. (1997)
9. Nie, L., Ning, Z., Wang, X., Hu, X., Cheng, J., Li, Y.: Data-driven intrusion detection for intelligent internet of vehicles: A deep convolutional neural network-based method. IEEE Transactions on Network Science and Engineering **7**(4), 2219–2230 (2020)
10. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**(1), 1–47 (mar 2002). https://doi.org/10.1145/505282.505283, https://doi.org/10.1145/505282.505283

11. Seo, E., Song, H.M., Kim, H.K.: Gids: Gan based intrusion detection system for in-vehicle network. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST). pp. 1–6 (Aug 2018). https://doi.org/10.1109/PST.2018.8514157
12. Ullah, S., Khan, M.A., Ahmad, J., Jamal, S.S., e Huma, Z., Hassan, M.T., Pitropakis, N., Buchanan, W.J.: Hdl-ids: a hybrid deep learning architecture for intrusion detection in the internet of vehicles. Sensors **22**(4), 1340 (2022)
13. Yang, L., Moubayed, A., Shami, A.: Mth-ids: a multitiered hybrid intrusion detection system for internet of vehicles. IEEE Internet of Things Journal **9**(1), 616–632 (2021)
14. Zaccone, G., Karim, M.R.: Deep Learning with TensorFlow - Second Edition: Explore Neural Networks and Build Intelligent Systems with Python. Packt Publishing, 2nd edn. (2018)
15. Zafar, N., Haq, I.U., Chughtai, J.u.R., Shafiq, O.: Applying hybrid lstm-gru model based on heterogeneous data sources for traffic speed prediction in urban areas. Sensors **22**(9) (2022). https://doi.org/10.3390/s22093348, https://www.mdpi.com/1424-8220/22/9/3348