BioGecko Vol 12 Issue 02 2023

ISSN NO: 2230-5807

A Machine Learning based Approach in Categorization of Traffic Intrusion in Vehicular Ad Hoc Network

Dr.P.Anand[1], Dr.L.Raja[2], Dr.A.Ganesan[3], Dr.A.Thankaraj[4], Dr.E.Mohan[5] ,Dr.E.Gajendran[6]
[1]Associate Professor, Department of EEE, DHAANISH AHMED College of Engineering, Chennai, India
[2] Professor, Department of ECE, RRASE College of Engineering, Chennai, India
[3,4]Professor, Department of EEE, RRASE College of Engineering, Chennai, India
[5]Professor, [6]Associate Professor, Department of Computer Science Engineering, Mohamed Sathak A.J College of engineering, Chennai, India
[1]virudaianand@gmail.com,[2]kaushikraja2000@gmail.com,
[3]dragmephd@gmail.com,[4]atraj77@gmail.com,[5]emohan1971@gmail.com, [6]gajendrane@gmail.com

Abstract: Modern vehicles go beyond being mere machinery. It uses a number of electronic control units connected by intra-vehicle networks to accomplish various features and perform operations. Intelligent Transportation Systems (ITS), a promising technology for VANET communication systems, has the potential to increase security gaps by linking all of the vehicle networks together. There are several obstacles to overcome in order to satisfy the varied security and privacy standards of VANET. In a real-time vehicle network, data is transmitted reliably and effectively using the Controller Area Network (CAN) standard. Because the CAN protocol broadcasts messages in broadcast mode, an attacker can simply bring about system failures by injecting any message into the CAN. The most crucial components of in-vehicle security are ECU and CAN bus security, thus an intrusion detection system (IDS) that use machine learning techniques to identify hostile cyber-attacks is created to ensure that this doesn't happen again. Our system uses the K-Nearest Neighbor, SVM, Nave Bayes, and Decision Tree supervised algorithms to identify Dos and Fuzzy assaults and categorize intrusion traffic. The four performance metrics employed are recall, accuracy, F-score, and precision. Based on the performance criteria mentioned above, our proposed system recommends the optimum algorithm for classifying legitimate and attacked traffic in a VANET context. In our studies, we use various ratios of training to test data to compare the performance of various machine learning algorithms. Our findings show that data separation is essential for assessing the effectiveness of machine learning algorithms.

Keywords: KNN, SVM, Recall, Accuracy, Precision

1 INTRODUCTION

Intelligent transportation systems (ITS) are crucial in today's digital age for making citizens' lives easier in every aspect. Vehicle Ad-hoc Networks (VANETs) are an ITS system that shows promise. The main objective of VANET is to make mobile users always connected when driving in order to increase security, comfort, driving directions, and traffic congestion. Some of the problematic aspects of VANET include self-organization, mobility, quick inter-contact intervals, network fragmentation, topology, changeable vehicle densities, and short-range communications. As VANETs' entire correspondence takes place in an open access environment, attacks that could compromise and corrupt the entire VANET system are more likely to occur. Also, in order to protect the safety of both users and the road, VANET offers reliable information to drivers. The vehicular system must face multiple security threats and attacks, which makes them the primary difficulties that must be prioritized in the development of solutions. These cars need to be reliable, safe, and predictable. The study of attack mitigation, such as that related to distributed denial of service assaults, is of interest to many scholars. The authors of [4-6] found that the network and communication are vulnerable to several attacks and threats. Data risks and attacks to the VANET system are the most dangerous. Denial of service is one of the malicious attacks on availability. Each layer of the network can be the target of a DOS attack. Denial-of-service attacks aim to overwhelm a host by flooding it with a large amount of data, effectively stopping it from accepting or processing data from authorised users. Basic, extended, and distributed denial of service (DDOS) attacks are the three categories into which DOS attacks fall [3].

As a result, an intrusion detection system (IDS) is a security tool that continuously scans host and network traffic for any unusual activity that defies the security policy and endangers its confidentiality, integrity, and availability. When malicious conduct is found, the IDS will notify the host or network administrators. The researchers looked into the application of machine learning (ML) and deep learning (DL) methods to satisfy the requirements of a successful IDS. A potential way for precise detection and prediction processes employed in many fields and domains, including VANETs, is provided by the machine learning approach.

This study proposes an architecture for the analysis and classification of VANET traffic datasets using an AI algorithm. This method's pre-processing of the data and classification are based on K-Nearest Neighbor, SVM, Nave Bayes, and Decision Tree, four machine learning techniques. By changing the ratios of training and testing data, experimentation is conducted, and the outcomes are analyzed. For our analysis, we took into account the performance criteria Accuracy, Precision, Recall, and F-Score. The relevant research that has been done in this area is compiled in Section 2. An overview of machine learning algorithms is given in Section 3. The proposed architecture for the analysis and classification of VANET traffic datasets is the main topic of Section 4. The research results are summarized in Section 5 by contrasting the effectiveness of the chosen machine learning algorithms. The study work's conclusions and its future scope are presented in Section 6.

2 LITERATURE REVIEW

Vehicular ad hoc networks (VANETs) are a significant research topic in vehicular and distant innovations because of people's ongoing adaptability, the increase in automobiles on the road, and the need for framework-less correspondence innovation for intelligent transportation systems (ITS).

An anomaly detection method was suggested [10-12]. The proposed approach showed that there is no false-positive mistake, but the attacker's algorithm won't be able to identify the attack at all if they inject messages and are unable to result, break, and affect the CAN. The decision tree method was employed by Swamy and Lakshmi (2012) to create an intrusion detection system. The KDD'99 data set was examined using C4.5, an improved decision tree and a descendant of ID3. The models' true negative and false positive rates were contrasted. The results showed that the upgraded Decision tree categorised attacks more accurately than the standard one. A machine learning-based misbehavior detection system was proposed by Fuad et al. [13-14]. Data collection, data sharing, analysis, and decision-making are the four processes that make up this process. Feed forward and reverse propagation algorithms employ artificial neural network (ANN) techniques. It functions by using methods for classification and training based on past data from both legitimate and harmful data. Because they used the real-world traffic dataset known as, this model is more successful (NGSIM).A Bayesian classification model for Denial-of-Service assaults was put forth by Hema and Shyni (2015) in an effort to increase detection rates and lower false positive alarms in the system. In order to prevent DoS attacks, the study sought to identify secure routes between a source and a destination.

Long Short-Term Memory (LSTM)-based neural network-based IDS was proposed by the authors [3,9], who also looked at several CAN bus attacks such DoS, fuzzy attacks, and spoofing attacks. The CAN Vehicle Spy 3 programme was used to collect data on both regular and attack communications. They executed DoS attacks by flooding the network while capturing normal messages every 120 seconds. The dataset in this instance was created using a real car, and DoS, fuzzy attacks, and spoofing attacks were also conducted on it.For the CAN Bus Message Attack Detection Framework, Tariq et al. proposed combining rule-based and Recurrent Neural Network (RNN) anomaly detection models (CAN-ADF). In order to gather 7,875,792 CAN messages for the dataset, two cars—a Kia Soul and a Hyundai Sonata—were set up to run nonstop for 24 hours.

Al-Saud et al. [8,10] suggested a data-driven method for anomaly identification based on the support vector machine (SVM) model. Data from an electric vehicle is gathered, including regular data and data related to DoS attacks. Their detection method exhibited a greater intrusion detection accuracy when compared to other machine learning algorithms like k-Nearest Neighbor (k-NN), the Decision Tree, RBF NN, and a traditional SVM. [6,11] CAN timing-based proposed specification-based anomaly detection.

3 MACHINE LEARNING ALGORITHMS – AN OVERVIEW

The various issues that arise in data networks can be addressed by a variety of machine learning algorithms, including Random Forest, Gradient Boosting Machine (GBM), Support Vector Machine (SVM), Logistic Regression, Multinomial Logistic Regression, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Principal Component Analysis, K-Means, Nave Bayes, Decision Trees, and others.

4 THE PROPOSED INTRUSION DETECTION SYSTEM FOR VANET TRAFFIC

As a result of the shortcomings in in-vehicle security, attackers will be keen to take advantage of CAN Bus security flaws. We examine two categories of in-vehicle assaults in our proposed VANET traffic dataset analysis and classification model: DoS and Fuzzy attacks. In order to disrupt the system, the message injection attack injects fake messages into the original ECUs.

Effectively identifying questionable network connections requires an intrusion detection system (IDS). In our suggested system implementation, we incorporated two automotive hacking datasets provided by the renowned Hacking and Countermeasure Research Lab (HCRL).

The dataset was created by recording CAN traffic over an OBD-II connection while message injection attacks were being conducted on actual vehicles. There are 300 message injection invasions in each dataset. Every dataset had 30 to 40 minutes of CAN traffic, and each intrusion lasted 3 to 5 seconds [1]. A DoS dataset includes 3,665,771 messages, of which 3,078,250 are regular messages and 587,521, are used by the proposed IDS system. A total of 3,838,860 message rows—2,759,492 normal messages and 1,079,368 injected messages—make up the Fuzzy dataset. Here, we conducted two different CAN bus attacks. A DoS attack injects CAN messages (such as the "0x000" CAN ID packet) in short cycles every 0.3 milliseconds, whereas a Fuzzy attack injects random messages in short cycles every 0.5 milliseconds.

In "Table 1," the following characteristics of this dataset are described: CAN packets contain a timestamp, CAN ID, data length code (DLC), flag, and an 8-bit data field (DATA [0]-DATA [7]).

TABLE 1 Data Attributes

Attributes	**Features**
Timestamp	Recorded time (s)
CAN ID	identifier of CAN message in HEX
DLC	Number of bytes, from 0 to 8

Flag	T or R T indicates injected message R indicates normal message
DATA [0] -DATA [7]	Data value by bytes

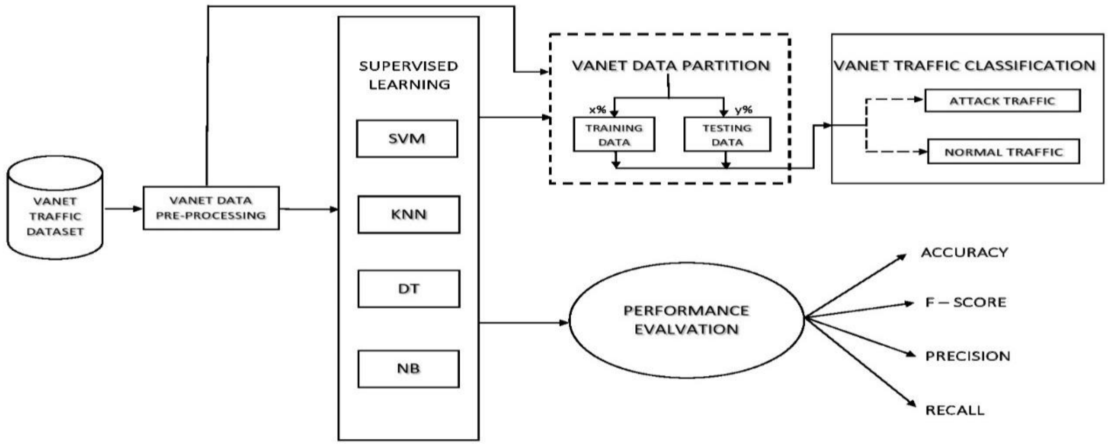

FIGURE 1 Architecture for Vanet traffic dataset analysis and classification

Our suggested model's operation is shown in Figure 1. The VANET Dataset from the HCRL Research Lab was initially pre-processed using techniques including data type compatibility, data formatting, and others. The dataset must then be cleaned by deleting unnecessary columns of data and any missing data. Our car hacking dataset's "timestamp" attribute was removed, which constrained the range of our investigation. The hexadecimal data is then converted to decimal data format. Four supervised machine learning algorithms—KNN, SVM, Decision Tree (DT), and Naive Bayes—were conducted on the extracted VANET dataset (NB). Several training and testing data ratios have been applied to the dataset. The data has been assigned the labels 1 and 0, respectively, for regular and injected messages. The machine learning model was trained using the aforementioned algorithms. Lastly, we utilise four metrics—Accuracy, Precision, Recall, and F-Score—to evaluate how well machine learning models perform for different training/testing dataset ratios.

4.1 K-Nearest Neighbour

Similar things are thought to be near together. Because the method memorises the training cases rather than learning a model, it is also known as an instance-based learner. This algorithm's drawback is that as the amount of data being used increases in size, the algorithm becomes noticeably slower. It acknowledges the presence of related things nearby. Hence, similar things are situated next to one another.

The KNN Algorithm:
1. Load the data
2. Set k = 1.
• Measure the separation between each training data row and each test data row.
We will use Euclidean distance as our distance metric in this case because it is the most often utilised approach. Other metrics, such as Chebyshev and cosine, may also be employed.
• Sort the calculated distances in ascending order using the distance values.
• Pull the first k rows of the sorted array.
• Identify the row's most prevalent class.
• Provide the anticipated class.

4.2 Support Vector Machine (SVM)

An approach for supervised machine learning called SVM can be used to solve classification or regression issues [22]. However, because it can conduct classification in a non-linear manner by utilising the kernel method, classification issues are where it is most frequently used. The SVM is equipped to recognise spoofing and network intrusion assaults. SVM is based on the idea of decision planes, which specify decision boundaries, as opposed to other machine learning methods. It uses some sort of graphical method.

4.3 Decision Tree (DT)

A straightforward example of categorising inputs is a decision tree, in which data is continuously divided based on a particular parameter. High-dimensional data can be handled accurately by decision trees. Choosing which value to utilise to split a decision tree's node is the main challenge in decision tree construction.

4.3.1 The Decision Tree Algorithm:

• Using attribute selection measures (ASM) to divide the dataset records, the best attribute is chosen (i.e., the best internal node in the tree). The knowledge that provides the most information from a classification standpoint will be the best attribute.
• Subdivide the dataset into smaller subsets using that property as a decision node.
• Build the tree by repeatedly going through this method for each child (internal node) up until one of these conditions is met:
- The same attribute value encompasses the entirety of tuples.
- There are no more attributes or instances left.

The split mechanism mentioned above is used to build the tree shape. In order to score each property, ASM explains the dataset that is provided. The best scoring factor will be selected as a splitting factor. Split points for branches must be stated when a property with a continuous value is present. The Gini Index, Gain Ratio, and Information Gain are the three most popular selection metrics.

4.4 NAIVE BAYES

A supervised learning algorithm, but not just one algorithm, but rather a collection of algorithms that share a same fundamental principle. It is largely used in text categorization, a field with a sizable and intricate training dataset. The Naive Bayes model is simple to construct when the features in the datasets are unrelated to one another since it is both a quick classifier and insensitive to unrelated features. It is ideal for binary scenarios. The three varieties of the Naive Bayes Model are Bernoulli, Multinomial, and Gaussian.

4.4.1THE NAIVE BAYES ALGORITHM:

• Data
This stage pre-processes the text before splitting the dataset into a train set and a test set.
Using Naive Bayes to Analyse the Training Data

After pre-processing, fit the Naive Bayes model to the Training set.
Gaussian NB classifiers can be used to train on this dataset.
estimating the test's outcome:
Make predictions using the predict function after creating a new predictor variable.
Using a confusion matrix generated during the experiment, the Naive Bayes classifier should be evaluated for accuracy.
The experiment's results are represented visually

5 PERFORMANCE EVALUATION

We are use the open-source Python tool "Scikit-Learn" for our machine learning studies. It is quick and simple to utilise this data mining and analysis tool. Together with methods for classification, regression, and clustering, this package also provides features for model selection, dimensionality reduction, and data pre-processing. The following configurations are necessary for our technical test bed in order to conduct our research: 13GB of RAM, two 2.30GHz cores, and several graphics processing units on a Xeon processor (GPUs).
Using a constructed DoS and Fuzzy dataset that contains information from both an attack and regular traffic, our system is put to the test. Scikit-learn uses cross-validation as part of supervised machine learning by dividing the available dataset into a training (Y test) and a testing (X test) set. Now, train the model using the classifier. Using a confusion matrix, any binary classifier's accuracy is evaluated. The outcomes are measured using True positives (TPs), True negatives (TNs), and false positive/false negative (FP/FN) indicators, with the data specifically divided between attacks and normal traffic (FNs).We can calculate Accuracy, F-score, Precision, and Recall using these values, as shown below:

Accuracy = TP+TN / TP+FP+FN+TN (1)

Precision = TP / TP+FP (2)

Recall = TP / TP+FN (3)

F-score = 2TP / 2TP+ FP+FP (4)

When false positives and false negatives have similar values, accuracy performs well. We can take into account both Precision and Recall if the values of False Positives and False Negatives are considerably different from one another.
The pre-processed dataset has been divided into multiple ratios of training and testing data (60%-40%, 70%-30%, 80%-20%, and 90%-10%) in order to evaluate our suggested approach. Later, the evaluation of four performance metrics using various ML algorithms is observed. We observed that the accuracy for the KNN and SVM algorithms [1] remains high at more than 96% for both datasets. If we increase the proportion of training datasets, the accuracy will be significantly better. When comparing KNN to SVM, KNN performs better with decent accuracy, and the F-score value stays essentially the same for all different types of data partition.
In our work, we test various training and testing data splits using both DoS and fuzzy datasets. The standard data splitting ratio of 70% for training data and 30% for testing data is taken into account for the best result analysis. Using DoS and fuzzy datasets, FIGURE 2 and 3 displays our performance evaluation of the Decision Tree algorithm and Nave Bayes.

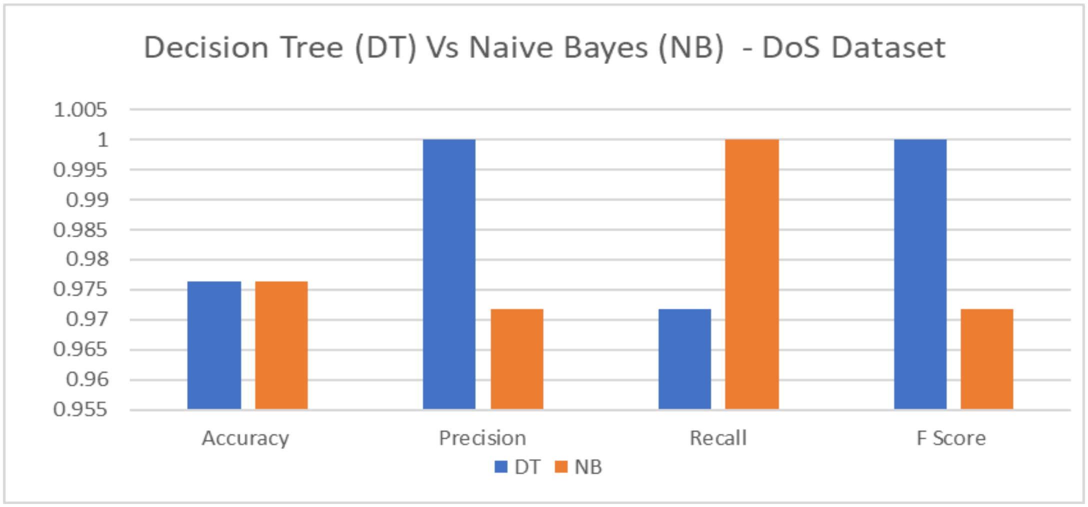

FIGURE 2 Comparison between Decision Tree and Naive Bayes for DoS dataset

The accuracy for both the Decision Tree and Naive Bayes algorithms maintains at 97.5% based on the graph above (Figure 2). For Decision Tree and Naive Bayes, the output precision was close to 100% and 97%, respectively. Yet, the recall performance factor's results appear to be the opposite of precise. With Decision Tree, the recall is still 97%, however for Naive Bayes, it is over 100%. For the Decision Tree, the F-Score clearly displays a 100% outcome, but for Naive Bayes, it is less than 98%.

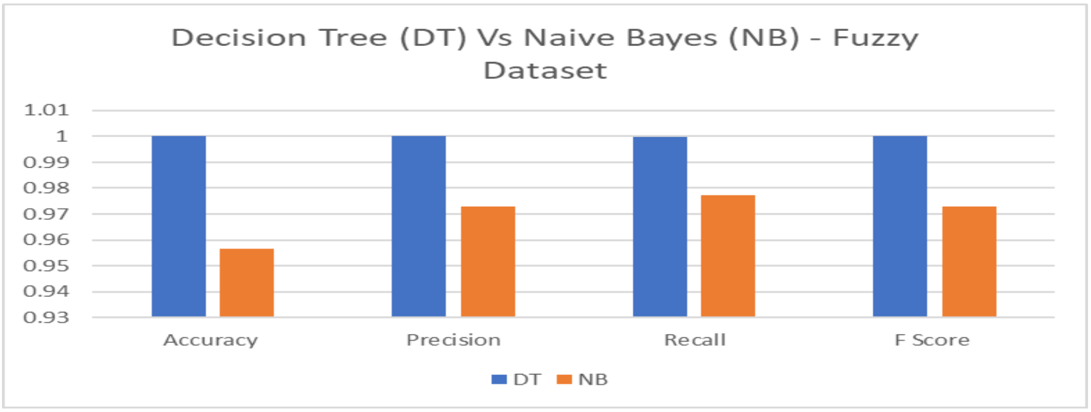

FIGURE 3 Comparison between Decision Tree and Naive Bayes for Fuzzy dataset

Figure 3 demonstrates that while Naive Bayes algorithms are less than 96% accurate, Decision Tree techniques are almost 100% correct. Naive Bayes' precision was over 97% whereas Decision Tree's was almost 100%. On the other hand, the precision performance factor's results and those of the recall performance factor are fairly similar. The recall rate for Decision Tree is still very close to 100% when compared to Naive Bayes. Naive Bayes has an F-Score of almost 97 percent, while the Decision Tree has a score of 100 percent. In conclusion, the DoS dataset's accuracy for the Decision Tree and Naive Bayes algorithms remains at 97%. All four performance indicators display high values when the fuzzy dataset is classified using a decision tree. From the statistics shown above, we can conclude that Decision Tree performs better in both datasets.Four performance measures have been examined for their effects on DoS and fuzzy attack traffic using DT and Nave Bayes algorithms. The inquiry involved several experiments, with the following findings as a result.

The accuracy, precision, recall, and F-Score performance parameters of the machine learning method are examined in Figures 4 to 8. This paper aims to investigate the impact of these performance indicators on different training and testing datasets. Researchers employ data partitions based on training and testing data to evaluate these performance aspects. 90/10, 80/20, 70/30, and 60/40 are these ratios.

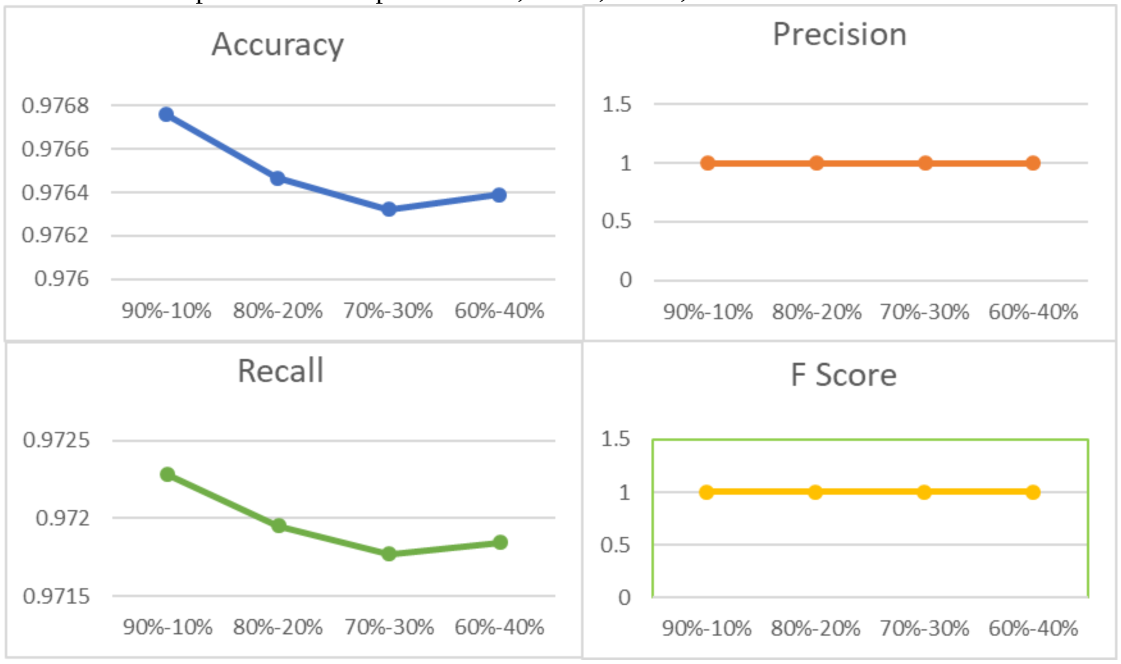

FIGURE 4. Performance metrics of DT algorithm over DoS attack traffic with varied training-testing dataset

The performance metrics of the DT algorithm (Accuracy, Precision, Recall, and F-Score) over DoS attack traffic on various testing and training data partitions are shown in FIGURE 4. It has been found that every performance directly affects how much training data is necessary to create a machine learning model. For all of the data divisions, the accuracy is still close to 97%, but 90:10 displays the highest accuracy. The data divisions are not significantly affected by the decision. The accuracy values are quite close to 100%. Recall metrics for the 70:30 data split are seen to be relatively poor. Eventually, the accuracy and F-Score yield findings that are comparable.

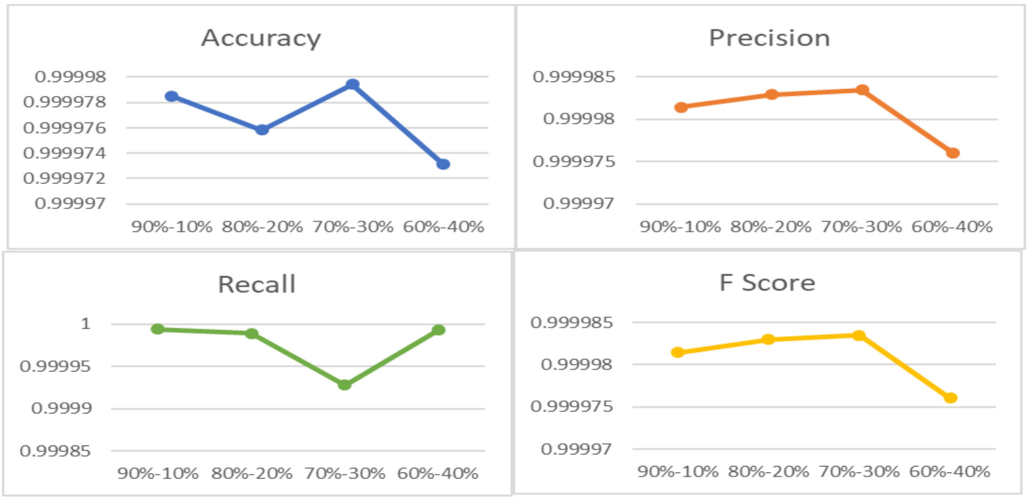

FIGURE 5 Performance metrics of DT algorithm over Fuzzy attack

FIGURE 5 shows the performance metrics for the DT algorithm over Fuzzy attack traffic on various testing and training data partitions (Accuracy, Precision, Recall, F-Score). It has been shown that utilising a 70%–30% data division for training and testing data results in good performance across the board. It has been found that the accuracy for the 70:30 data split is good. The data division from 90:10 to 70:30 yields good results according to the precision. Values greater than 99% are seen. With the exception of 70:30, recall results indicate good performance for all partitions. F-score performance is almost comparable to precision metrics.

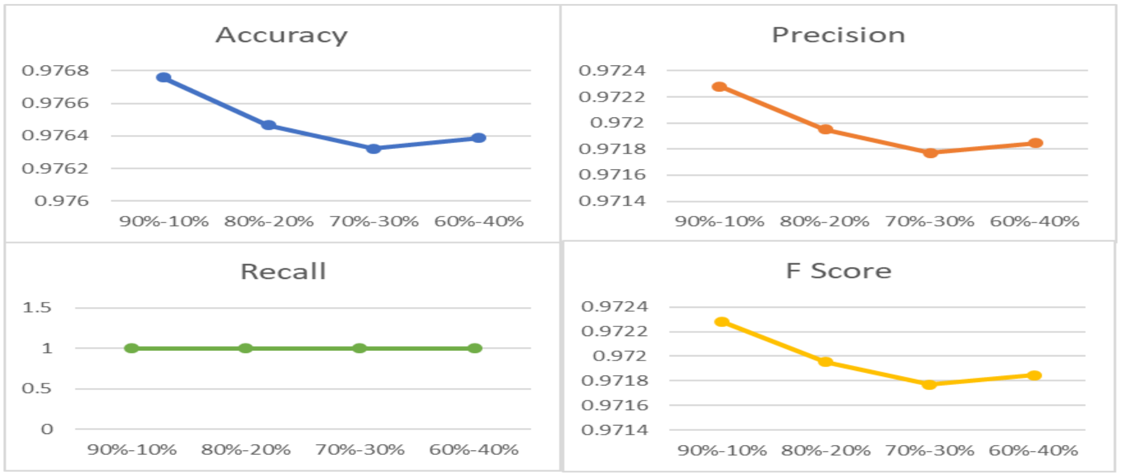

FIGURE 6 Performance metrics of NB algorithm over DoS attack traffic with varied training-testing dataset

FIGURE 6 illustrates the NB algorithm's performance metrics (Accuracy, Precision, Recall, and F-Score) under DoS attack traffic on various testing and training data partitions. All performance metrics except recall are observed to be significantly lower when training / testing data is partitioned into 70% -30% data partitions. For 90:10 data partitioning, the accuracy, precision, recall, and F-score all perform admirably. The research indicates that data partitioning has no discernible effect on recall metrics. For the 90:10 data partition, the accuracy, precision, F-score, and recall are all very near to 100%. A small improvement in accuracy, precision, and F-score is seen when the data partitioning ratio is adjusted from 70:30 to 60:40. FIGURE 7 shows the accuracy, precision, recall, and F-score performance metrics of the NB method over fuzzy attack traffic on various testing and training data partitions. It has been found that all performance indicators drastically vary when the proportion of training and testing datasets is marginally altered. i.e., 80% -20% to 70% -30%.

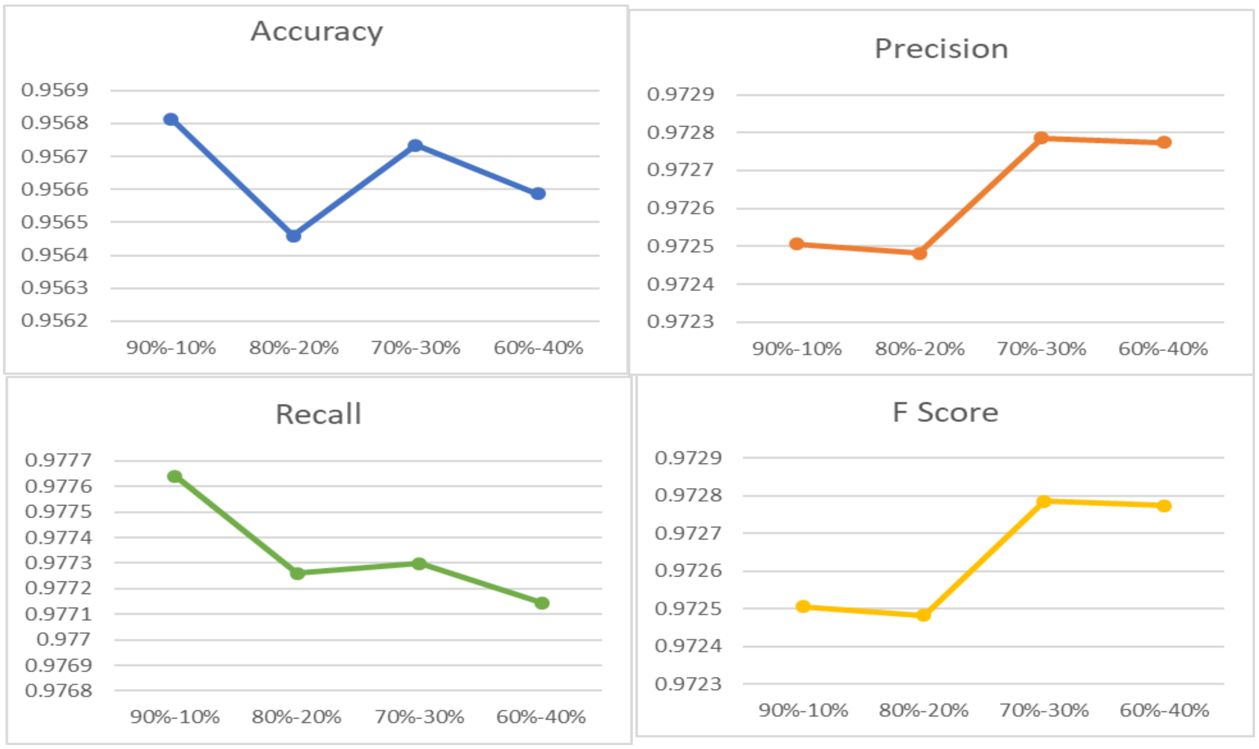

FIGURE 7 Performance metrics of NB algorithm over Fuzzy attack traffic with varied training-testing dataset

For this dataset, the accuracy metrics trend was erratic. The graph makes it quite evident that the accuracy values for the 90:10, 80:20, 70:30, and 60:40 data partitions are out of range. When compared to the 90:10 and 80:20 data split ratios, the precision output for 70:30 and 60:40 indicates a significant improvement. When there is a sizable amount of training data compared to testing data for the model, recall performance is seen to be strong. Values of F-Score are comparable to precision.

6 CONCLUSIONS

This article suggests a framework for detecting DoS and fuzzy attacks in VANET systems due to the openness of autonomous vehicles to various attacks. IDS is a reasonably priced method of protecting vehicle networks. Using the HCRL dataset for both attack and regular traffic, we compare four machine learning algorithms. Accuracy, Precision, Recall, and F-Score are the four essential performance indicators that we include in our research. The decision tree algorithm has the highest accuracy rate of all machine learning algorithms, according to the performance evaluation of our suggested work. On the Fuzzy dataset, the Decision Tree algorithm achieves a 99 percent accuracy rate, and on the DoS dataset, a 97 percent accuracy rate. With the DoS dataset, Naive Bayes is 97% accurate, and for the Fuzzy dataset, it is 95% accurate. Naive Bayes receives an F-score of 97 percent for both the DoS and Fuzzy datasets, while Decision Tree receives an F-score of 100 percent for the DoS dataset and 99 percent for the Fuzzy dataset. Furthermore, our research shows that data partitioning affects the various machine learning algorithm performance metrics. Our results indisputably show how data splitting is important for assessing the effectiveness of machine learning algorithms. This will help future researchers in this area think about the optimal data partitioning to use with machine learning algorithms for intrusion detection systems in a VANET setting in order to get the best results. The analysis of performance indicators using a range of different samples from the data partition may be included in the future scope of this project. Modifying the random state values used by machine learning algorithms could achieve this.

REFERENCES

1. Alshammari, A., A. Zohdy, M., Debnath, D. and Corser, G. (2018) Classification Approach for Intrusion Detection in Vehicle Systems. *Wireless Engineering and Technology*, **9**, 79-94. doi: 10.4236/wet.2018.94007
2. Song, Hyun Min, Jiyoung Woo, and Huy Kang Kim. "In-vehicle network intrusion detection using deep convolutional neural network." Vehicular Communications 21 (2020): 100198.
3. Hossain, D.; Inoue, H.; Ochiai, H.; Fall, D.; Kadobayashi, Y. LSTM-Based Intrusion Detection System for In-Vehicle Can Bus Communications. IEEE Access 2020, 8, 185489–185502.
4. Seo, Eunbi, Hyun Min Song, and Huy Kang Kim. "GIDS: GAN based Intrusion Detection System for In-Vehicle Network." *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2018.'
5. Sakiz, F. and Sen, S. (2017) A Survey of Attacks and Detection Mechanisms on Intelligent Transportation Systems: VANETs and IoV. Ad Hoc Networks, 61, 33-50. https://doi.org/10.1016/j.adhoc.2017.03.006.
6. Olufowobi, H.; Young, C.; Zambreno, J.; Bloom, G. SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing. IEEE Trans. Veh. Technol. 2019, 69, 1484–1494.
7. Tuohy, S., Glavin, M., Hughes, C., Jones, E., Trivedi, M. and Kilmartin, L. (2015) Intra-Vehicle Networks: A Review. IEEE Transactions on Intelligent Transportation Systems, 2, 534-554. https://doi.org/10.1109/TITS.2014.2320605.
8. Al-Saud, M.; Eltamaly, A.M.; Mohamed, M.A.; Fard, A.K. An Intelligent Data-Driven Model to Secure Intravehicle Communications Based on Machine Learning. IEEE Trans. Ind. Electron. 2019, 67, 5112–5119
9. Marchetti, M. and Stabili, D. (2017) Anomaly Detection of CAN Bus Messages through Analysis of ID Sequences. IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 1577-1583.

10. Gao, Y, Wu, H, Song, B, et al. A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network. IEEE Access 2019; 7: 154560–154571.
11. Lyamin, N, Kleyko, D, Delooz, Q, et al. Real-time jamming DoS detection in safety-critical V2V C-ITS using data mining. IEEE Communication Lett 2019; 23(3): 442–445.
12. O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, and A. Mouzakitis, “Intrusion Detection Systems for Intra-Vehicle Networks: A Review,” IEEE Access, vol. 7, pp. 21266–21289, 2019
13. Mani, R., Jayaraman, S., &Ellappan, M. (2020). Hybrid seagull and thermal exchange optimization algorithm-based NLOS nodes detection technique for enhancing reliability under data dissemination in VANETs. International Journal of Communication Systems, 33(14), e4519.
14. Dr.E.Mohan, Dr.A.Annamalai “Distributed Attack Detection For Wireless Sensor Networks ” International Journal of Engineering & Technology, Volume 7 ,issue 6, 465-468 , 2018, (ISSN: 2227-524X).

15. Mani, R., Jayaraman, S. and Ellappan, M., 2022. Hybrid invasive weed optimization and squirrel search algorithm-based NLOS nodes localization mechanism for improving reliable data dissemination in VANETs. *International Journal of Communication Systems*, *35*(10), p.e5171.