Deep Learning based Intrusion Detection System for Vehicular Ad-Hoc Network

Rohit Pravinkumar Vedpathak[1], Suraj Shivaji Redekar[2]

[1]M.Tech. Student, Department of Computer Science and Engineering, Ashokrao Mane Group of Institutions, Kolhapur, Maharashtra, India.
[2]Asst. Professor, Department of Computer Science and Engineering, Ashokrao Mane Group of Institutions, Kolhapur, Maharashtra, India.

---***---

Abstract - *Vehicular Ad-hoc Networks (VANETs) provides wireless communication among Vehicles and Infrastructures. Connected vehicles are promising in Intelligent Transportation Systems (ITS) and smart cities. The main objective of VANET is to spice up safety, comfort, driving efficiency, and waiting time on the road. VANET is different from other ad-hoc networks, due to its unique properties and a high degree of mobility. However, it's at risk of various security attacks like DoS attacks, Fuzzy attacks, and Impersonation Attacks, due to the shortage of centralized infrastructure. This poses a big threat to vehicle safety. In CAN bus there's no information about the source and destination address for authentication. The attacker can easily inject any message. This could cause bugs. During this work, we used Simple RNN, RNN with LSTM, and RNN with attention mechanism based on Deep Learning for IDS to integrate and differentiate the intrusions in VANET. The intrusion detection technique depends on the offset ratio analysis and also the measure between the messages request and also the response within the CAN. Intrusion detection system plays an important role in ensuring safety, security, and, therefore, the key technology is to accurately detect a variety of attacks on the network. The RNN-IDS model provides a completely new method of research, intrusion detection, and improves the accuracy of intrusion detection.*

***Key Words*: CAN Bus, Intrusion Detection, IDS, Deep Learning, RNN, LSTM, VANET, Neural Networks**

1. INTRODUCTION

The rapid expansion of the transmission of information between a variety of devices and protocols has led to serious problems for the protection increases the importance of the development of modern intrusion detection systems (IDS). In today's world, many people are actively making use of cars and other personal vehicles. A crucial problem that each person has faced every day is the increasing number of accidents occurred on road and this transportation safety problem continues to worsen because of population growth and the increase in the number of vehicles in urban areas.

A Vehicular ad-hoc network (VANET) refers to a network in a very special way, where the different moving vehicles and other devices connect to a wireless carrier, and the exchange of information, at least on their own for a variety of reasons, that is the most important task for the improvement of road safety. A small network is formed simultaneously with cars and other devices that behave as a mode network. The communication system of an intelligent vehicle is usually referred to as a vehicle to everything or it is also called a VANET which means Vehicular Ad-Hoc Network. An Ordinary VANET, the communication system is normally responsible for 3 main types of communication to be considered on the smart automobile. Those types are vehicle to vehicle, vehicle to infrastructure, and vehicle to roadside. There is a major advancement in-vehicle system has been made with integrating a number of computing devices called ECU. Different types of communication protocols are designed to support communication. CAN is the simple communication protocol supporting attaching sensors and actuators with ECUs.

In this work, we introduce IDS based on RNN's deep learning strategies to integrate and differentiate intrusions in VANET.

1.1 CAN Bus

The Controller Area Network (CAN) a bus is a communication protocol that will be used as a standard for efficient and reliable transfer of data between the vehicle and the nodes in real-time. In such a network, broadcast messages must transmit from one node to a different on the bus and there's no information about the source and therefore the destination address for the validation. This security hole results in inject any message by an attacker that may cause system malfunction.

The reliability of inter-vehicle communication is one of the most important aspects is to ensure the widespread adoption of unified transport systems. Inter-Vehicle communication vehicles to exchange messages over a short range. There are a variety of different protocols established for the service of communication. The primary protocol is Controller Area Network (CAN) protocol. This is a bus, a serial communication protocol for connecting sensors and actuators to the ECU. The ECU can be hacked and compromised. An attacker can get remote code execution of the Electronic Control Unit (ECU) in automotive vehicles through interfaces like Bluetooth. The attacker can influence the behavior of vehicles like display, steering, braking, and acceleration, etc. As mentioned earlier, due to the weakness of the vehicle, the attackers are able to use the CAN Bus vulnerability. In this work, we investigated three types of attacks that occur in-vehicle: DoS, Fuzzy, and Impersonation attacks. We proposed an Intrusion Detection System to detect these kinds of attacks.

A] DoS (Denial-of-Service) Attack

In a DoS attack, the network server is overloaded with too many requests. Since VANETs use wireless technology, and it's very easy for attackers to launch a denial of service attack. As a result, receivers of VANET services may not be receiving the real-time delivery of the service and will lead to catastrophic results.

In DoS attacks, the attacker can inject high-priority messages in an exceedingly very short period of time into the bus. Not only that, it's easy to achieve control of a node within the network by the attacker. Therefore, it's easy to send the best priority identifiers. Thus, the network is flooded very quickly and eventually will lead to accidents and so on.

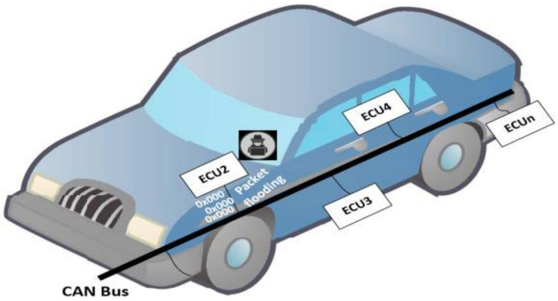


Fig-1.1: DoS Attack Scenarios on In-Vehicle Network

B] Fuzzy Attack

In a fuzzy attack, the attacker injects messages of arbitrary spoofed identifiers with random data. As a result, all nodes of the network receive lots of functional messages and it's going to cause malfunction of the network. This may lead to mal behavior in vehicles.

To launch a fuzzy attack, the attacker observed in-vehicle messages and also the chosen target identifiers. This might cause unexpected behaviors.

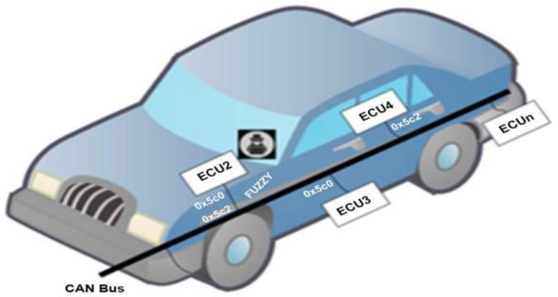


Fig-1.2: Fuzzy Attack Scenarios on In-Vehicle Network

C] Impersonation Attack

In an Impersonation Attack, the attacker updates the message and claims that the message comes from an original authenticated source and the identity of the legitimate node is stolen by an attacker.

In an impersonation attack, the attacker intentionally sending false information to the network. The aim of this kind of message is to send to form confusion within the communication or to call selfish behavior of node to induce some facility.

2. RELEVANT WORK

In [1], the authors focus on the IDS system which is classified into four layers namely, data collection, feature identification, model training, and execution of the classification model. The dataset used is CICIDS2017. The authors present an efficient method using anomaly detection by LSTM based neural network with an accuracy of 96% to determine distributed denial of service attacks.

In [2], the authors have proposed the RNN intrusion detection system model not only features a strong modeling ability for intrusion detection but also has high accuracy in both binary and multiclass categorization. Compared with traditional classification methods, such as naive bayesian, and random forest, the performance obtains the next detection rate and accuracy rate with a low false positive rate, especially under the multiclass classification on the NSL-KDD dataset. The model can functionally improve both the accuracy of intrusion detection and also the ability to detect the intrusion type.

In [3], the authors propose an intrusion detection system that determines the intrusions in vehicles. They use two algorithms based on SVM and KNN to detect the DoS and Fuzzy attacks. For analysis, they use two sets of car hacking data provided by HCRL (Hacking and Countermeasures Research Lab). In classification, they used two algorithms of the most popular methods of classification: Support Vector Machine and K-Nearest Neighbor. First, the authors made a pre-processing for data. Then, extracted the features of each dataset. After that, they implemented KNN and SVM algorithms.

In [4], the authors analyze the response performance of nodes to detect whether a vehicle is under attack or not. OTIDS can better detect message injection attacks and which node attacks potentially more dangerous attacks on vehicles. In addition, OTIDS can detect which types of messages were injected during a message injection attack, and which node was damaged during an impersonating node attack. The authors believe their detection method contributes to enhancing vehicle security without changing the CAN protocol.

In [5], the authors have applied their own implementation of the LSTM recurrent neural network classifier to intrusion detection data. The results show that the LSTM classifier offers superior performance as compared to the results of the KDD Cup '99 challenge and moreover other tested strong static classifiers. The strengths are in the detection of `DoS' attacks and network probes, which both produce a specific time series of events. The performance on the attack classes that produce only some events is like the results of the other tested classifiers. Performance is measured in terms of mean-squared error, accuracy, ROC curve, confusion matrix, and the corresponding AUC value. And that they finally conclude that LSTM is incredibly suitable for classifying high-frequency attacks. For low-frequency attacks, the advantage of using LSTM vanishes. Although we state that the results achieved by LSTM are very competitive. This is the first report demonstrating the effective use of LSTM recurrent neural networks, for intrusion detection.

In prior studies, a number of approaches based on traditional machine learning, including SVM [7], [8], K-Nearest Neighbor (KNN) [9], ANN [10], Random Forest (RF), [11], and others [12], [13], are proposed and have achieved success for an intrusion detection system.

In [14], the authors utilize a deep learning approach based on a deep neural network for flow-based anomaly detection, and also the experimental results show that deep learning is applied for anomaly detection in software-defined networks.

In [15], the authors propose a deep learning-based technique using self-taught learning (STL) on the benchmark NSL-KDD dataset in a network intrusion detection system. When comparing its performance with those observed in previous studies, the technique is shown to be more effective. However, this category of references focuses on the feature reduction ability of deep learning. It mainly uses deep learning techniques for pre-training, and it performs classification through the standard supervision model. The use of deep learning techniques to perform classification directly is not common, and there is an absence of study of the performance in multiclass classification.

In [16], the authors proposed a method for in-vehicle intrusion detection based on the investigation of the rate of messages. Due to the number of messages on the CAN bus that includes the sum of the normal and attacks messages; they analyzed the rates of messages per second in order to detect anomalous message rates.

In [17], the authors introduced a method for anomaly detection. The proposed technique proved that there is no false-positive error, but if the attacker injects messages and could not outcome and break and affect the CAN, then their algorithm cannot detect the attack at all.

[18], the authors proposed an effective misbehavior detection model based on machine learning techniques. The method has four stages: data acquisition, data sharing, analysis, and decision making. They used Artificial Network (ANN) methods using the feedforward and the backpropagation algorithms. It works by classifying and training based on historical data from both malicious and normal data. They used a real traffic dataset called (NGSIM), so that's making their model more effective.

With the continuous development of big data and computing power, deep learning methods have grown rapidly, and are widely used in a variety of fields.

3. PROPOSED INTRUSION DETECTION SYSTEM

We propose an intrusion detection system for vehicular ad-hoc networks that determine the intrusions in vehicles. We have used three algorithms are RNN, RNN with LSTM, and RNN with Attention Mechanism to detect the DoS, Fuzzy, and Impersonation attacks. Through the analysis, we have used three car-hacking datasets: “DoS dataset”, “Fuzzy dataset” and “Impersonation dataset”, which are provided by the Hacking and Countermeasure Research Lab (HCRL). [6] These datasets are taken from the actual vehicles, by connecting the CAN traffic through the OBD-II port. OBD II stands for On-Board Diagnostic II. OBD-II is an on-board computer that monitors emissions, mileage, speed, and other information of the vehicle. Then, they got the performance of the message injection attacks. The DoS dataset has 656,579 numbers of messages. It has 12 columns. The Fuzzy dataset has 591,990 numbers of messages and 12 columns. The Impersonation dataset has 995,472 numbers of messages and 12 columns.

Datasets:

I. DoS Attack: Inject '0x000 'CAN ID messages in a short cycle.
II. Fuzzy Attack: Inject messages of randomly selected CAN values and DATA values.
III. Impersonation Attack: Inject Impersonating Node messages, arbitration ID = '0x164'.

Data Attributes:

Timestamp, CAN ID, DLC, and DATA [0], DATA [1], DATA [2], DATA [3], DATA [4], DATA [5], DATA [6], DATA [7].

I. Timestamp: Recorded time (s)
II. CAN ID: CAN message identifier in HEX (e.g. 043f)
III. DLC: number of data bytes, from 0 to 8.
IV. DATA [0 ~ 7]: Amount of data (byte)

The structure of the three datasets is similar, though they represent different types of attacks. The DoS attack dataset represents DoS attacks, where it involves injecting message of “0000” CAN ID every 0.3 milliseconds, we observe that “0000” is that the most dominant CAN ID. Injecting messages of totally random CAN ID and Data values every 0.0 milliseconds represents fuzzy attacks in fuzzy attack dataset. We note that, if RTR bit = 0 and All data values are not 0 then we have a fuzzy attack. The impersonation attack dataset represents impersonation attacks, where it involves injecting messages of “0x164” CAN ID. We note that the impersonating node injects a message of “0x164” CAN ID and the difference between current and previous timestamp is more than 250 seconds.

We mentioned before that these datasets are similar, so we applied the same preprocess to them. First, we added appropriate header names to every dataset as they are unmarked with headers. Then, we removed unnecessary columns from DoS and Fuzzy attack dataset, which were the Timestamp as we do not have a time-series analysis. But, we have used the timestamp column of the impersonation attack dataset because the difference between current and previous timestamps is required to detect impersonation attacks. We removed the missing data as well. We also converted hexadecimal data into decimal format. We distinct the normal messages with 1 and the injected messages with 0. We split the dataset into Training and Testing data randomly in the ratio of 70:30. For intrusion detection, we used three algorithms. 1. Simple RNN, 2. RNN with LSTM and 3. RNN with Attention. First, we made a preprocessing on the dataset as we mentioned above. Then, we extracted the features of each dataset. After that, we implemented Simple RNN, RNN with LSTM, and RNN with Attention and we will explain how they work in the next step.

3.1 Simple RNN

Recurrent neural networks (RNN) are a category of neural networks that are useful for modeling data sequencing. Derived from feed forward networks, RNNs express similar behavior to how human brains function. To put it simply: the recurrent neural network produces predictive results in the following data that the other algorithms do not.

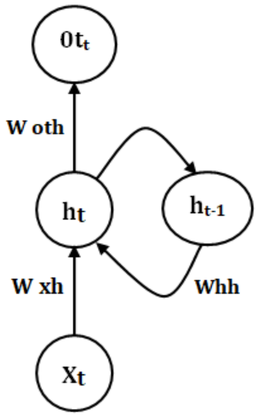


Fig-3.1: Simple RNN Architecture

A Recurrent Neural Network works on the goal of saving the output of a particular layer and feeding this back to the input so on forecast the output of the layer. Recurrent means the output at the present time step becomes the input to the subsequent time step. In RNN the information is being processed by means of a loop. When it makes a decision, it considers the current input and also what it has learned from the inputs it receives formerly.

We have used the following parameters when implementing the Simple RNN; The number of passes of the entire training dataset the algorithm has completed is 20, The number of training examples utilized in one iteration is 32, The Dropout is 0.1 which prevents the model from overfitting, the optimizer is adam, The loss function is binary cross-entropy and metrics is accuracy. We used the activation function tanh i.e. hyperbolic tangent activation.

3.2 RNN with LSTM

Long Short Term Memory Network is the extension of Simple RNN. An extra layer is that the LSTM layer. An LSTM includes a similar control flow as a recurrent neural network. It processes data passing on the information because it propagates forward. There are differences in operations in the LSTM's cell. These operations are wont to allow the LSTM to stay or forget information.

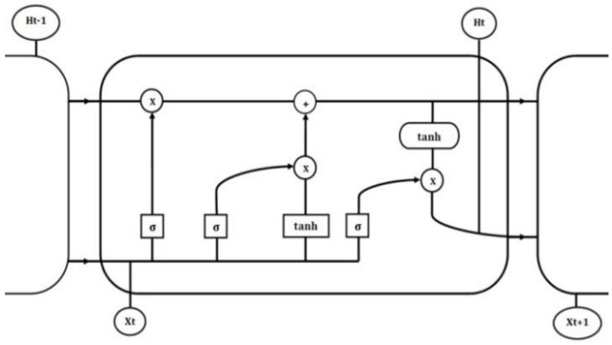


Fig-3.2: RNN with LSTM Architecture

LSTM has a three-step process: Each LSTM module has three gates named Forget Gate, Input Gate, and Output Gate. Input gate - It identifies which value from input should be used to modify the memory. Forget gate - It determines what details are to be discarded from the block. This is determined by the sigmoid function. Output gate - The input and also the memory of the block are used to decide the output.

We have used the following parameters when implementing the RNN with LSTM; The number of passes of the entire training dataset the algorithm has completed is 20, The number of training examples utilized in one iteration is 32, The Dropout is 0.1 which prevents the model from overfitting, the optimizer is adam, The loss function is binary cross-entropy and metrics is accuracy. We used the activation function tanh i.e. hyperbolic tangent activation and hard sigmoid recurrent activation.

3.3 RNN with Attention

The attention mechanism in deep learning relies on the concept of directing your focus, and information it pays inordinate attention to certain factors, and when processing the data. In wide terms, Attention is one component of a network's architecture and is responsible for managing and quantifying the interdependence: Between the input and output elements and within the input elements.

The attention mechanism with RNN allows focusing on specific parts of the input sequence where it predicts the selected part of the high-quality output sequence and enables easy learning. The merging of attention mechanisms enabled improved performance in numerous tasks making it an integral part of latest RNN networks.

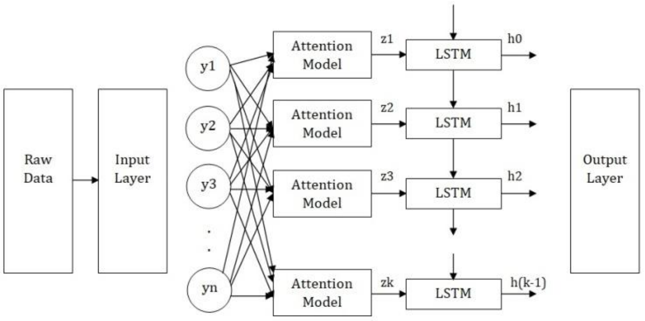


Fig-3.3: RNN with Attention Architecture

We have used the following parameters when implementing the RNN with Attention; The number of passes of the entire training dataset the algorithm has completed is 20, The number of training examples utilized in one iteration is 32, The Dropout is 0.1 which prevents the model from overfitting, The optimizer is adam, The loss function is binary cross-entropy and metrics is accuracy. We used the activation functions tanh i.e. hyperbolic tangent activation and softmax.

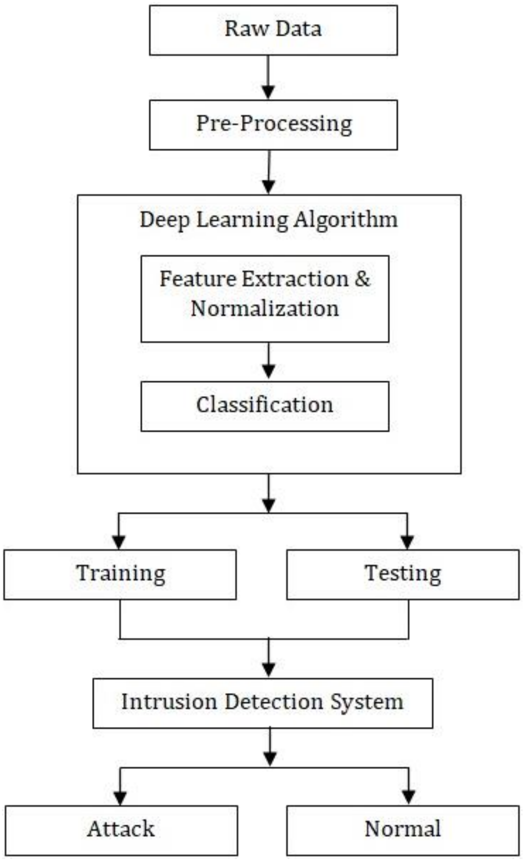


Fig-3.4: Block Diagram of Proposed System

The block diagram shows the process of the proposed methodology.

1) Raw Data: The raw data contain input data and training datasets. The HCRL's CAN dataset for Intrusion Detection will use to train the model, which includes DoS attacks, fuzzy attacks, and impersonation attacks. The input data is test data that will test in the proposed system.

2) Pre-processing: Input data and training data will then give to the pre-processing, this block is used for the process of the data output of this block is given as an input to the feature extraction block.

3) Feature Extraction & Normalization: In these blocks, various features can be extracted from the given input data or from the training dataset. In the proposed methodology for feature extraction deep learning algorithms namely Simple RNN, RNN with LSTM, RNN with Attention are used. In a deep learning algorithm, feature extraction can be done automatically.

4) Classification: A deep learning classifier is used to classify the input data as an attack or normal data.

5) Intrusion Detection System: This block gives output whether the input data is normal or attack. For the proposed methodology, deep learning algorithms are used to detect intrusions or attacks on CAN Bus.

4. PERFORMANCE EVALUATION & DISCUSSION

Through this analysis, we used Python, and the main reason we used python is that its libraries help a lot with deep learning. The libraries we used are Scikit-Learn, Numpy, Pandas, Matplotlib, and Keras. Actually, Scikit is formulated to interoperate with the Python numerical and scientific library like NumPy. The Scikit-Learn library focuses on data modelling but not on loading, and summarizing data. NumPy library is used for working with arrays. It also has functions for working within the domain of algebra, Fourier transform, and Matrices. Pandas library is used for working with data sets. It includes functions for data analyzing, cleaning, exploring, and manipulating data. Matplotlib is a low-level graphics library in python, which is used as a visualization tool. Keras is a high-level neural network API/Library written in Python which hides the complexity of neural networks from users and provides an abstraction. It runs on top of other Deep Learning Frameworks such as TensorFlow, Theano, and CNTK. Generally, TensorFlow used as a backend to Keras API. It is the quick and fastest way to develop an application with minimum coding. It provides API for FNN, CNN, RNN, and many more, etc. In order to detect the intrusive data, we use deep learning algorithms - Simple RNN, RNN with LSTM, and RNN with Attention Mechanism. We implemented RNN by using Keras API. In the proposed system, we have used Confusion Matrix, and Classification Report, ROC Curve to measure the performance of the RNN-IDS model.

4.1 Confusion Matrix

Confusion Matrix is a performance calculation for a classification problem where the output will be two or more classes. In our proposed system, we used a confusion matrix because the output of our model is in two classes - normal and attack. It's used to assess the performance of the classification model. It is a table with four various combinations of predicted and actual values. It's extremely useful for measuring Recall, Precision, Recall, and F1 Score.

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Fig-4.1: Confusion Matrix

True Positive (TP): The actual value was positive and the model predicted positive value.

True Negative (TN): The actual value was negative and model predicted negative value.

False Positive (FP): The actual value was negative but the model predicted positive value.

False Negative (FN): The actual value was positive but the model predicted negative value.

4.2 Classification Report

A Classification report is used to measure the quality of predictions from classification models or algorithms. The classification report displays the precision, recall, f1, support, and accuracy scores for the model.

a] Precision

It also called Positive predictive value. It is the ratio of correct positive predictions to the total predicted positives.

$$Precision = \frac{TP}{TP + FP}$$

b] Recall

It is also called Sensitivity, Probability of Detection, and True Positive Rate. The ratio of the corresponding positive predictions for the total number of positive examples.

$$Recall = \frac{TP}{TP + FN}$$

c] F1-Score

It is harmonic mean of precision and recall. It considers both precision and recall. The maximum score is 1 and the minimum 0.

$$F_1\text{–score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

d] Accuracy

It is simply equal to the proportion of predictions that the model accurately classified.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

4.3 Receiver Operating Characteristics Curve

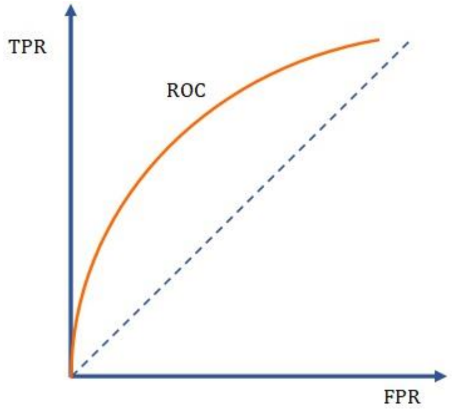


Fig-4.2: Receiver Operating Characteristics Curve

Is useful tool when predicting the probability of a binary outcome is the Receiver Operating Characteristic curve or ROC Curve. It's a plot of the false positive rate (X-Axis) versus the true positive rate (Y-Axis) for a number of dissimilar threshold values in-between 0.0 and 1.0

True Positive Rate is the number of observations that are correctly predicted to be positive.

$$\textbf{TPR} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

The false positive rate is the number of observations that are incorrectly predicted to be positive.

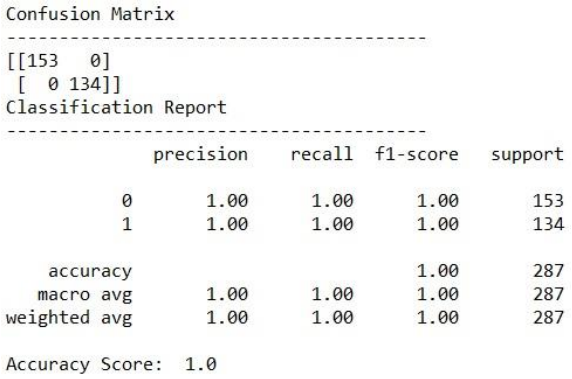
$$\textbf{FPR} = \frac{False\ Positive}{True\ Negative + False\ Positive}$$

For different threshold values, we will get different True Positive Rate and False Positive Rate. Thus, in order to visualize which threshold is best suited for the classifier, we have used the ROC curve. The above figure 4.2 shows what a typical ROC curve looks like.

The ROC curves that fall below the area in the upper left corner indicate good performance levels; while the ROC curves that fall elsewhere in the lower right corner indicate poor performance levels. A merging of two straight lines both moving away from the baseline towards the top-left corner is a perfect classifier of ROC Curve.

5. RESULTS

The dataset contains a large number of messages as we mentioned in the proposed system. Each dataset needs from 30 - 40 minutes of CAN traffic. So we used the first 1000 rows from the DoS Attack dataset, the Impersonation Attack dataset, and the first 5000 rows from the Fuzzy Attack dataset to checking the proposed intrusion detection system within less time and to obtain the results from the proposed intrusion detection system. We chose 5000 rows of the Fuzzy Attack dataset because the Fuzzy Attack dataset has fewer attacks than DoS and Impersonation attack datasets.



```
Confusion Matrix
----------------------------------------
[[153   0]
 [  0 134]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       153
           1       1.00      1.00      1.00       134

    accuracy                           1.00       287
   macro avg       1.00      1.00      1.00       287
weighted avg       1.00      1.00      1.00       287

Accuracy Score:  1.0
```

Fig-5.1: Confusion Matrix and Classification Report of DoS Attack Detection using Simple RNN

In Figure 5.1, we used the confusion matrix to describe the performance of a simple RNN model and to visualize important predictive statistics such as recall, accuracy, and precision. The accuracy score in figure 5.1 is 1.0. It indicates that the Simple RNN model classifies 100% of intrusions correctly.

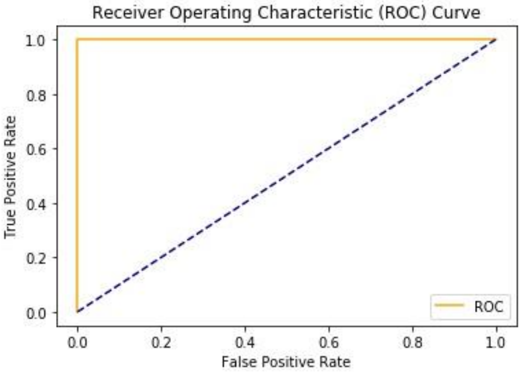
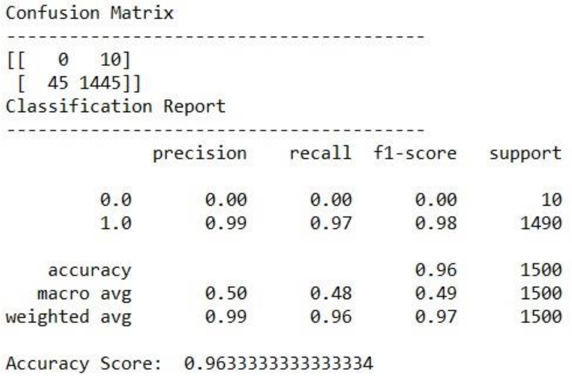


Fig-5.2: ROC Curve of DoS Attack Detection using the Simple RNN

In figure 5.2 shows the output of Simple RNN applied to the DoS attack dataset. ROC Curve is fallen under the area of the top left corner hence it indicates the best performance of Simple RNN in intrusion detection in the DoS attack dataset.



```
Confusion Matrix
----------------------------------------
[[   0   10]
 [  45 1445]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        10
         1.0       0.99      0.97      0.98      1490

    accuracy                           0.96      1500
   macro avg       0.50      0.48      0.49      1500
weighted avg       0.99      0.96      0.97      1500

Accuracy Score:  0.9633333333333334
```

Fig-5.3: Confusion Matrix and Classification Report of Fuzzy Attack Detection using Simple RNN

The accuracy score in figure 5.3 is 0.96. It indicates that the Simple RNN model classifies 96% of intrusions correctly.

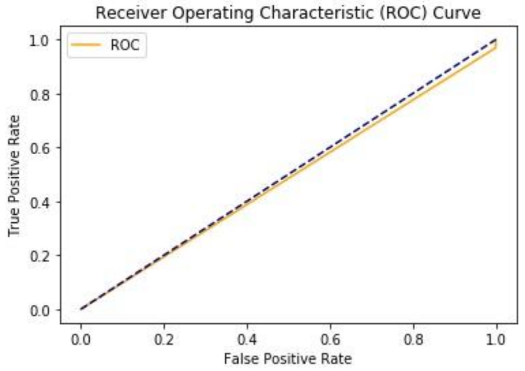
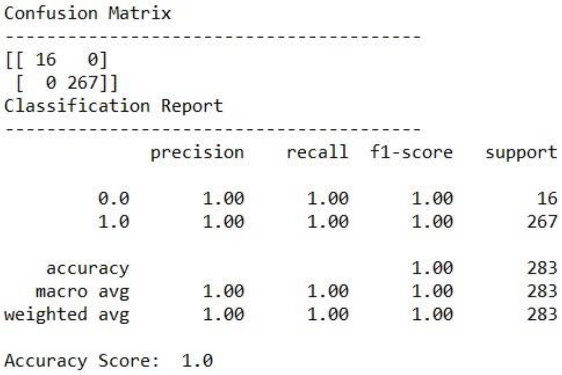


Fig-5.4: ROC Curve of Fuzzy Attack Detection using the Simple RNN

The dotted blue and orange color lines show in the above figure. In figure 5.4 both lines are similar because random spoofed attacks are very few in the fuzzy attack dataset. It indicates the good performance of the Simple RNN model in intrusion detection in the fuzzy attack dataset.



```
Confusion Matrix
----------------------------------------
[[ 16   0]
 [  0 267]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        16
         1.0       1.00      1.00      1.00       267

    accuracy                           1.00       283
   macro avg       1.00      1.00      1.00       283
weighted avg       1.00      1.00      1.00       283

Accuracy Score:  1.0
```

Fig-5.5: Confusion Matrix and Classification Report of Impersonation Attack Detection using Simple RNN

The accuracy score in figure 5.5 is 1.0. It indicates that the Simple RNN model classifies 100% of intrusions correctly.

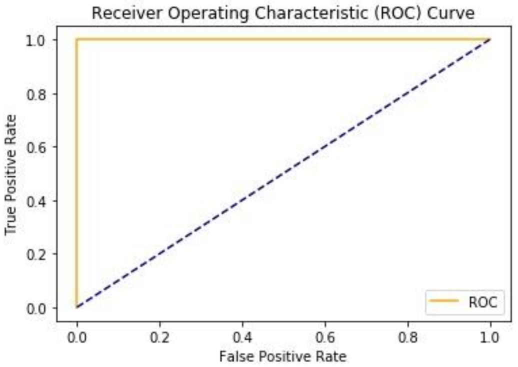
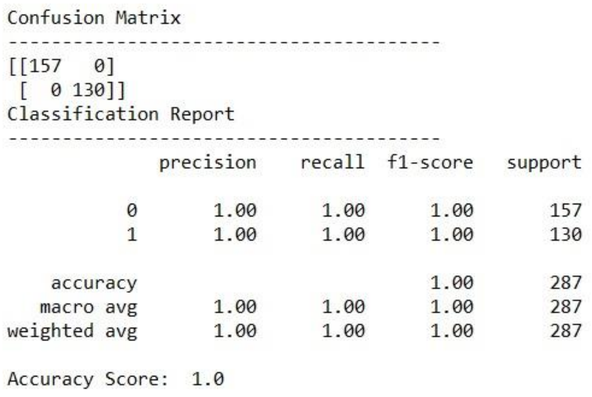


Fig-5.6: ROC Curve of Impersonation Attack Detection using Simple RNN

In figure 5.6 shows the output of Simple RNN applied to the impersonation attack dataset. ROC Curve fallen under the area of the top left corner hence it indicates the best performance of Simple RNN in intrusion detection in impersonation attack dataset.



```
Confusion Matrix
----------------------------------------
[[157   0]
 [  0 130]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       157
           1       1.00      1.00      1.00       130

    accuracy                           1.00       287
   macro avg       1.00      1.00      1.00       287
weighted avg       1.00      1.00      1.00       287

Accuracy Score:  1.0
```

Fig-5.7: Confusion Matrix and Classification Report of DoS Attack Detection using RNN with LSTM

The accuracy score in figure 5.7 is 1.0. It indicates that the RNN with LSTM model classifies 100% of intrusions correctly.

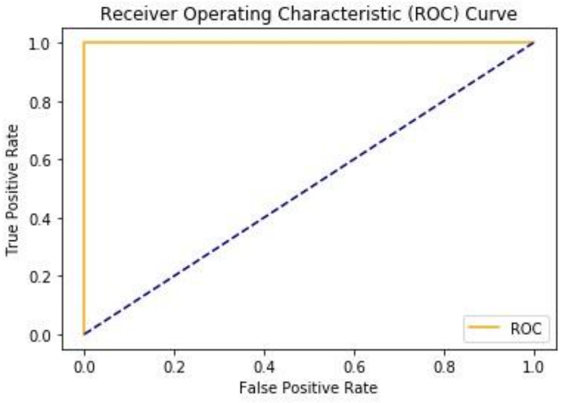
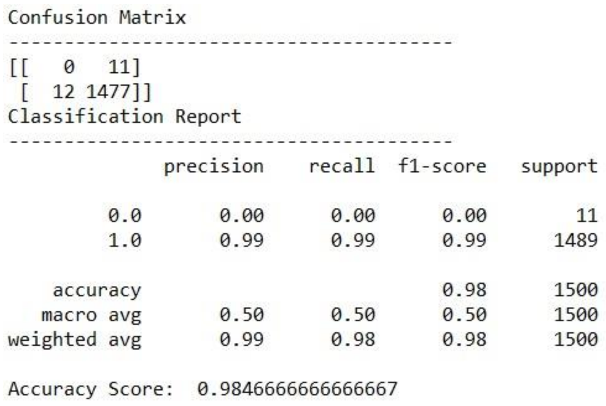


Fig-5.8: ROC Curve of DoS Attack Detection using RNN with LSTM

In figure 5.8 shows the output of RNN with LSTM applied to the DoS attack dataset. ROC Curve fallen under the area of the top left corner hence it indicates the best performance of RNN with LSTM in intrusion detection in the DoS attack dataset.



```
Confusion Matrix
----------------------------------------
[[   0   11]
 [  12 1477]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00        11
         1.0       0.99      0.99      0.99      1489

    accuracy                           0.98      1500
   macro avg       0.50      0.50      0.50      1500
weighted avg       0.99      0.98      0.98      1500

Accuracy Score:  0.9846666666666667
```

Fig-5.9: Confusion Matrix and Classification Report of Fuzzy Attack Detection using RNN with LSTM

The accuracy score in figure 5.9 is 0.98. It indicates that the RNN with LSTM model classifies 98% of intrusions correctly.

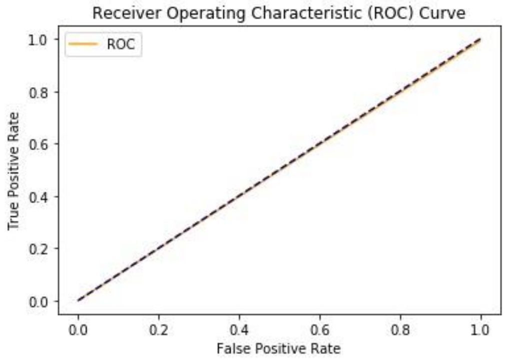
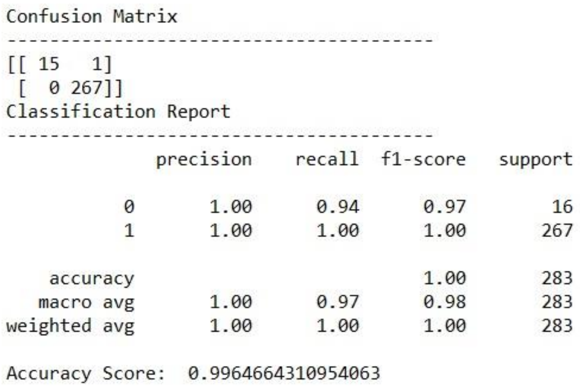


Fig-5.10: ROC Curve of Fuzzy Attack Detection using RNN with LSTM

The dotted blue and orange color lines show in above figure 5.10. In the figure, both lines are similar because random spoofed attacks are very few in the fuzzy attack dataset. It indicates the good performance of RNN with the LSTM model in intrusion detection in the fuzzy attack dataset.



```
Confusion Matrix
----------------------------------------
[[ 15   1]
 [  0 267]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

           0       1.00      0.94      0.97        16
           1       1.00      1.00      1.00       267

    accuracy                           1.00       283
   macro avg       1.00      0.97      0.98       283
weighted avg       1.00      1.00      1.00       283

Accuracy Score:  0.9964664310954063
```

Fig-5.11: Confusion Matrix and Classification Report of Impersonation Attack Detection using RNN with LSTM

The accuracy score in figure 5.11 is 0.99. It indicates that the RNN with LSTM model classifies 99% of intrusions correctly.

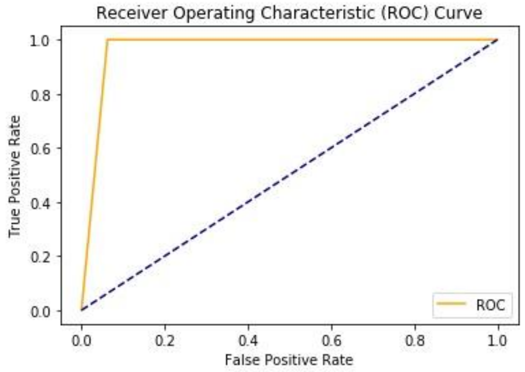
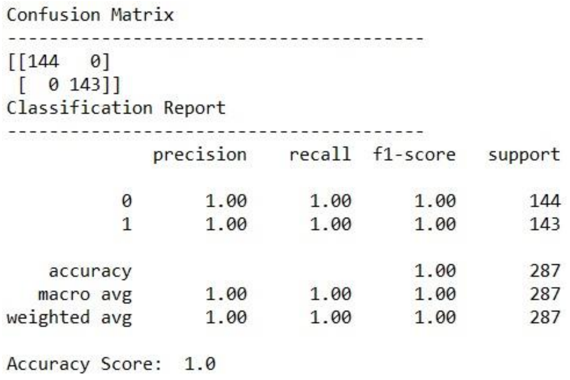


Fig-5.12: ROC Curve of Impersonation Attack Detection using RNN with LSTM

In figure 5.12 shows the output of RNN with LSTM applied to the impersonation attack dataset. ROC Curve fallen under the area of the top left corner hence it indicates the best performance of RNN with LSTM in intrusion detection in the impersonation attack dataset.



```
Confusion Matrix
----------------------------------------
[[144   0]
 [  0 143]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       144
           1       1.00      1.00      1.00       143

    accuracy                           1.00       287
   macro avg       1.00      1.00      1.00       287
weighted avg       1.00      1.00      1.00       287

Accuracy Score:  1.0
```

Fig-5.13: Confusion Matrix and Classification Report of DoS Attack Detection using RNN with Attention

The accuracy score in figure 5.13 is 1.0. It indicates that the RNN with Attention model classifies 100% of intrusions correctly.

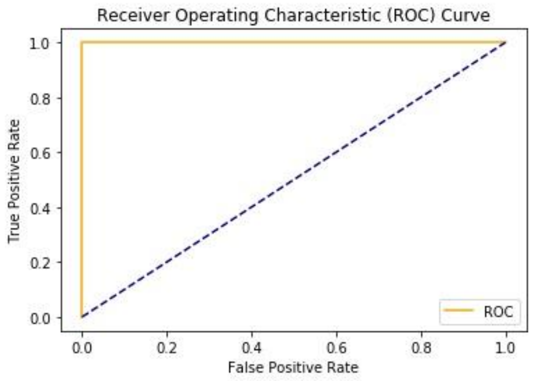
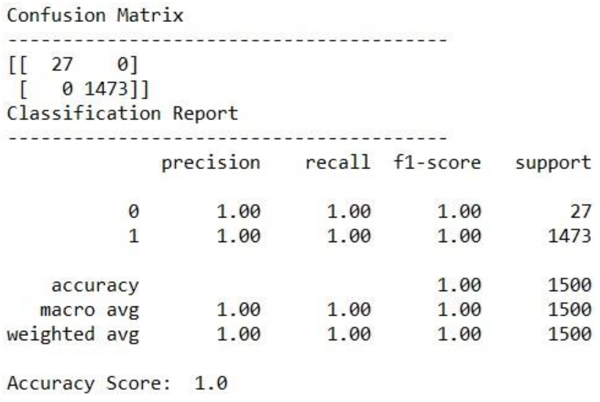


Fig-5.14: ROC Curve of DoS Attack Detection using RNN with Attention

In figure 5.14 shows the output of RNN with Attention applied to the DoS Attack Dataset. ROC Curve is fallen under the area of the top left corner hence it indicates the best performance of RNN with Attention in intrusion detection in the DoS attack dataset.



```
Confusion Matrix
----------------------------------------
[[  27    0]
 [   0 1473]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       1.00      1.00      1.00      1473

    accuracy                           1.00      1500
   macro avg       1.00      1.00      1.00      1500
weighted avg       1.00      1.00      1.00      1500

Accuracy Score:  1.0
```

Fig-5.15: Confusion Matrix and Classification Report of Fuzzy Attack Detection using RNN with Attention

The accuracy score in figure 5.15 is 1.0. It indicates that the RNN with Attention model classifies 100% of intrusions correctly.

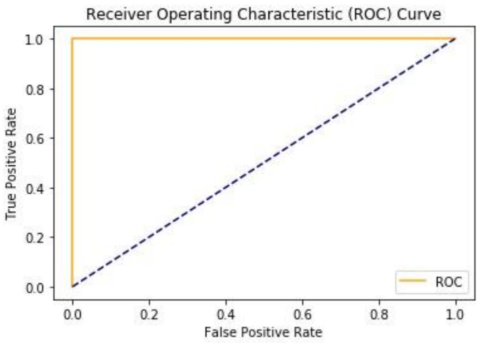
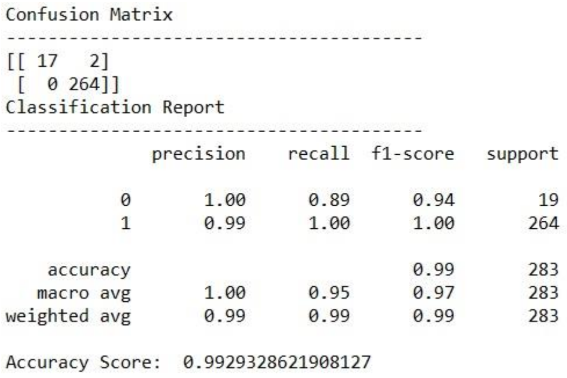


Fig-5.16: ROC Curve of Fuzzy Attack Detection using RNN with Attention

In figure 5.16 shows the output of RNN with Attention applied to the fuzzy attack dataset. ROC Curve is fallen under the area of the top left corner hence it indicates the best performance of RNN with Attention in intrusion detection in fuzzy attack dataset.



```
Confusion Matrix
----------------------------------------
[[ 17   2]
 [  0 264]]
Classification Report
----------------------------------------
              precision    recall  f1-score   support

           0       1.00      0.89      0.94        19
           1       0.99      1.00      1.00       264

    accuracy                           0.99       283
   macro avg       1.00      0.95      0.97       283
weighted avg       0.99      0.99      0.99       283

Accuracy Score:  0.9929328621908127
```

Fig-5.17: Confusion Matrix and Classification Report of Impersonation Attack Detection using RNN with Attention

The accuracy score in figure 5.17 is 0.99. It indicates that the RNN with Attention model classifies 99% of intrusions correctly.

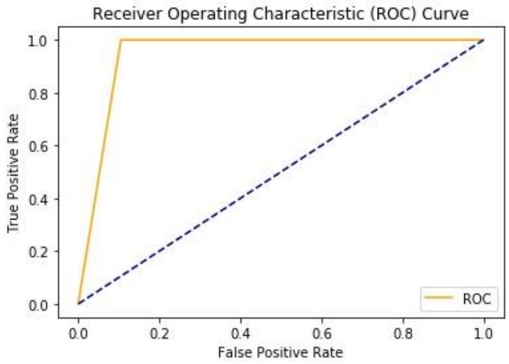


Fig-5.18: ROC Curve of Impersonation Attack Detection using RNN with Attention

In figure 5.18 shows the output of RNN with Attention applied to the impersonation attack dataset. ROC Curve fallen under the area of the top left corner hence it indicates the best performance of RNN with Attention in intrusion detection in the impersonation attack dataset.

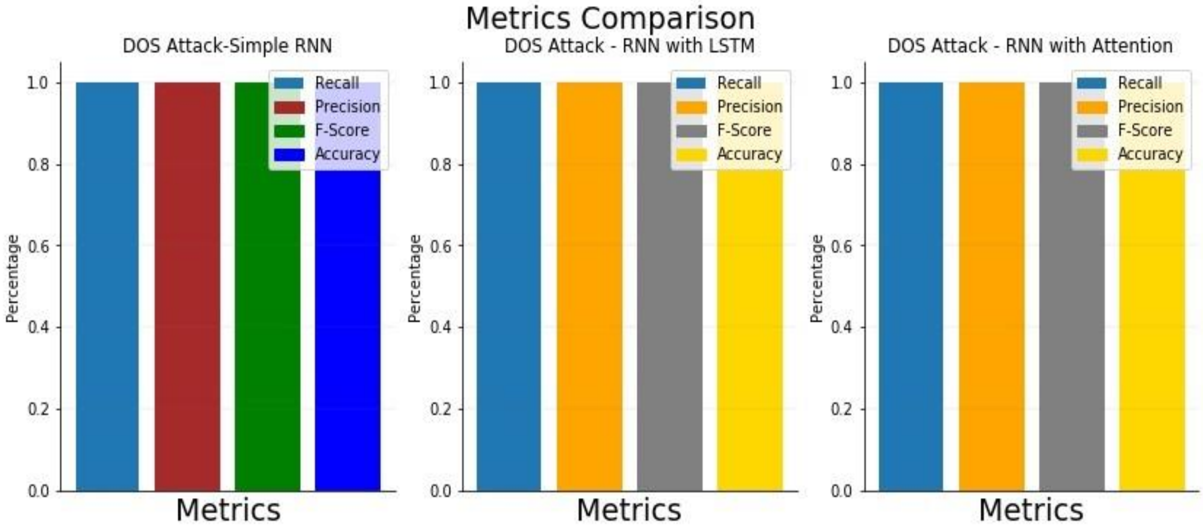


Fig-5.19: Metrics Comparison of DoS Attack Detection

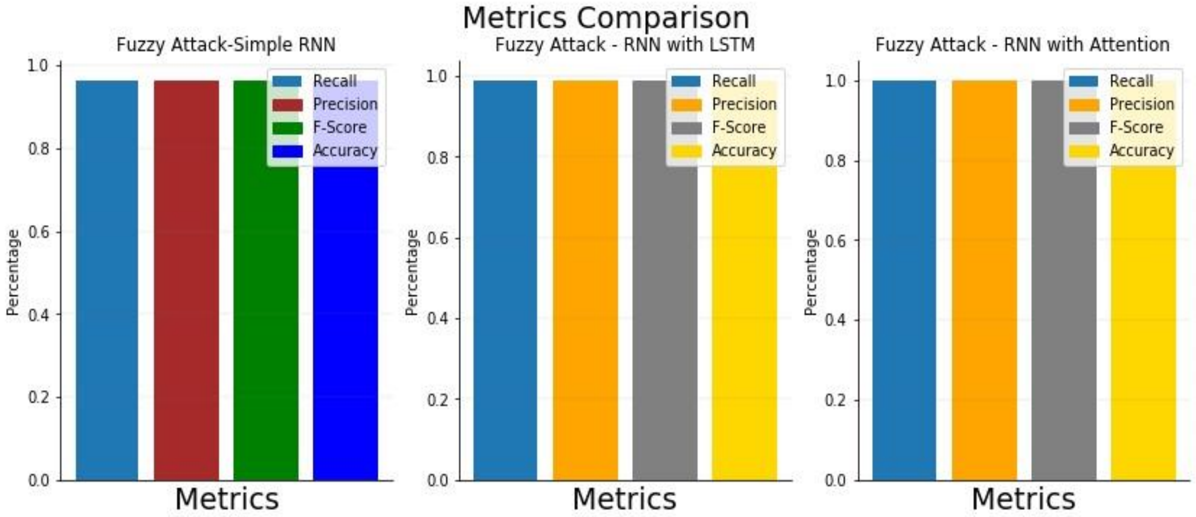


Fig-5.20: Metrics Comparison of Fuzzy Attack Detection

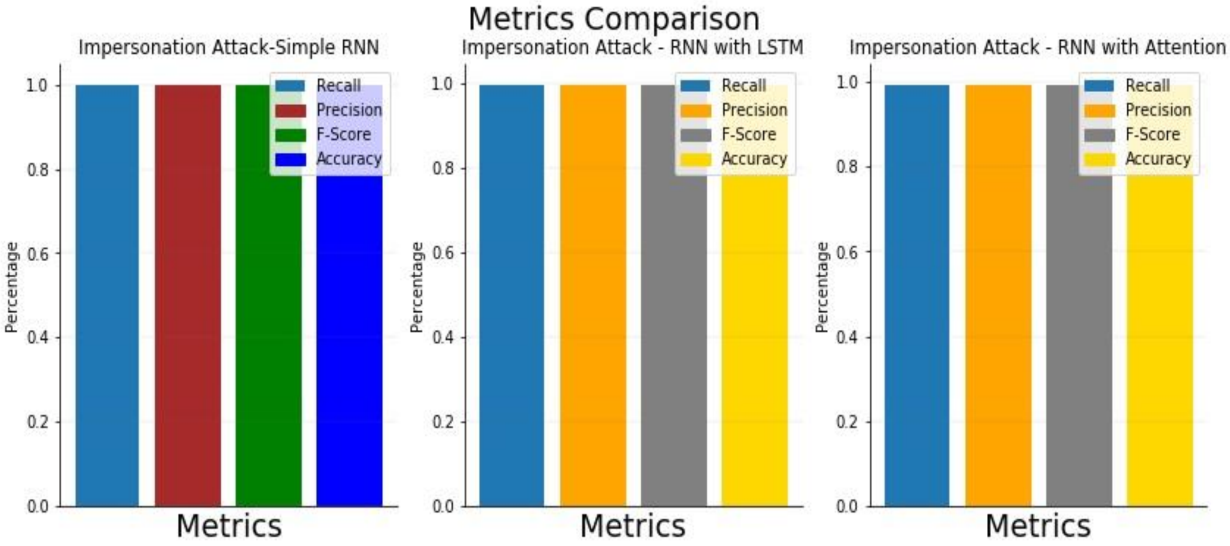


Fig-5.21: Metrics Comparison of Impersonation Attack Detection

6. CONCLUSION

In modern systems like connected vehicles, intelligent intrusion detection systems became a significant security application. These vehicles are targeted to different types of attacks which result in effects on the vehicles' performance, threats to public and personal property, and road safety. In this work, we propose an intrusion detection method for CAN Bus Intrusion Detection System in vehicles. It has the ability to detect DoS, Fuzzy, and Impersonation attacks that occur on CAN Bus. We use three data sets, one for DoS Attack, Fuzzy Attack, and another one for Impersonation Attack which is created by HCRL. We preprocessed the data and, then implemented the RNN, RNN with LSTM, and RNN with Attention algorithms. All of them provided great results; however, RNN with attention gave better performance than RNN and RNN with LSTM. Hence our intrusion detection system in the vehicular ad-hoc networks is successfully achieved through different technologies that involve the use of RNN, RNN with LSTM, RNN with Attention algorithms that clubbed with neural networks, and deep learning techniques.

7. REFERENCES

[1] S. Nayyar, S. Arora and M. Singh, "Recurrent Neural Network Based Intrusion Detection System," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 0136-0140

[2] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in IEEE Access, vol. 5, pp. 21954-21961, 2017.

[3] Alshammari, A., Zohdy, M.A., Debnath, D. and Corser, G. (2018) Classification Approach for Intrusion Detection in Vehicle Systems. Wireless Engineering and Technology, 9, 79-94.

[4] H. Lee, S. H. Jeong and H. K. Kim, "OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame," 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 2017, pp. 57-5709.

[5] Ralf C. Staudemeyer. (2015) "Applying Long Short-Term Memory Recurrent Neural Network for Intrusion Detection." IEEE Research Article - SACJ No. 56.

[6] Hacking and Countermeasure Research Lab (2017) CAN-Intrusion-Dataset. http://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

[7] F. Kuang, W. Xu, and S. Zhang, ``A novel hybrid KPCA and SVM with GA model for intrusion detection,'' Appl. Soft Comput., vol. 18, pp. 178_184, May 2014.

[8] R. R. Reddy, Y. Ramadevi, and K. V. N. Sunitha, ``Effective discriminant function for intrusion detection using SVM,'' in Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI), Sep. 2016, pp. 1148_1153.

[9] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, ``A new intrusion detection sys-tem based on KNN classification algorithm in wireless sensor network,'' J. Elect. Comput. Eng., vol. 2014, Jun. 2014, Art. no. 240217.

[10] B. Ingre and A. Yadav, ``Performance analysis of NSL-KDD dataset using ANN,'' in Proc. Int. Conf. Signal Process. Commun. Eng. Syst., Jan. 2015, pp. 92_96.

[11] N. Farnaaz and M. A. Jabbar, ``Random forest modeling for network intrusion detection system,'' Procedia Comput. Sci., vol. 89, pp. 213_217, Jan. 2016.

[12] J. A. Khan and N. Jain, ``A survey on intrusion detection systems and classification techniques,'' Int. J. Sci. Res. Sci., Eng. Technol., vol. 2, no. 5, pp. 202_208, 2016.

[13] A. L. Buczak and E. Guven, ``A survey of data mining and machine learning methods for cyber security intrusion detection,'' IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153_1176, 2nd Quart., 2016.

[14] A. Javaid, Q. Niyaz,W. Sun, and M. Alam, ``A deep learning approach for network intrusion detection system,'' presented at the 9th EAI Int. Conf. Bio-inspired Inf. Commun. Technol. (BIONETICS), New York, NY, USA, May 2016, pp. 21_26.

[15] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, ``Deep learning approach for network intrusion detection in software defined networking,'' in Proc. Int. Conf.

[16] Hoppe, T., Kiltz, S. and Dittmann, J. (2008) Security Threats to Automotive CAN Networks—Practical Examples and Selected Short-Term Countermeasures. International Conference on Computer Safety , Reliability , and Security , Newcastle upon Tyne, 22-25 September 2008, 235-248.

[17] Muter, M., Groll, A. and Freiling, F.C. (2010) A Structured Approach to Anomaly Detection for In-Vehicle Networks. 6th Information Assurance and Security , Atlanta, GA, 92-98.

[18] Ghaleb, F.A., Zainal, A., Rassam, M.A. and Mohammed, F. (2017) An Effective Misbehavior Detection Model Using Artificial Neural Network for Vehicular Ad Hoc Network Applications. IEEE Conference on Application, Information and Network Security, Miri, 13-14 December 2017, 13-18.