On the Robustness of Network Intrusion Detection System in In-Vehicular Networks

SIL 765 Project Report, Semester II, 2022-2023

Isha Pali
Computer Science & Engineering
Indian Institute of Technology Delhi
qiz228047@iitd.ac.in

ABSTRACT

This article explores the vulnerability of the controller area network (CAN), which is a common bus system used in in-vehicle networks, to message injection attacks. Since CAN relies on a broadcast-based communication mechanism with no authentication or encryption, it is vulnerable to cyber-attacks, which poses a serious risk to the safety of drivers. To identify suspicious communications over CAN, intrusion detection systems (IDS) have been developed, but recent research has shown that these IDS are vulnerable to adversarial attacks. The article proposes to assess the robustness of these IDS against hostile attacks by generating attack messages that could not be easily detected by the IDS, by changing a small portion of the original attack packets of the dataset. The article suggests using a delay as the mutation operation to be performed, and the objective is to investigate whether or not there are any shifts in the performance matrices of NIDS as a result of adding delay to the malicious packets contained in the initial traffic. The study's novelty lies in its actual adversarial attack carried out in the CAN network domain, making it a fruitful line of inquiry to pursue.

KEYWORDS

Controller Area Network, Cyber Attacks, Adversarial Attack, Intrusion Detection System, NIDS, ,

1 Introduction

1.1 Area

Modern advancements in the automobile sector have made our vehicles more sophisticated and connected. As a result, a typical vehicle is made up of hundreds of electronic control units (ECU), and all these ECUs communicate with each other through the bus system [1,2]. For In-vehicle networks, controller area network (CAN) is a typical bus system which facilitates effective communication between ECUs. Nevertheless, since CAN relies on a broadcast-based communication mechanism with no authentication or encryption, it is vulnerable to message injection attacks through the on-board diagnostic (OBD-II) port [3], and remote network channels [4]. Cyber-attacks over CAN must be avoided because they pose a serious risk to the safety of drivers. To identify suspicious communications over CAN, many intrusion detection systems (IDS) [5] have been developed.

1.2 Problem

The CAN bus system uses ML/DL models as network intrusion detection systems. However recent research, such that in [6], has revealed that adversarial attacks, which are made by intentionally produced perturbations over normal inputs, may be able to exploit these models. Therefore, our aim is to assess the robustness of these IDS against hostile attacks. A wide range of adversarial attacks have been created against ML/DL algorithms in a number of different contexts, including images, the real world [7], and malware [8]. The majority of adversarial attacks on NIDS have concentrated on directly changing network properties. However, because feature-level adversarial attacks only alter traffic features, which cannot be directly replayed in the network to carry out the planned malicious operations, they are not practical. Therefore, effective adversarial attacks should take into account packet space modifications that target NIDS.

.

1.3 Solution

In order to perform the attack, the aim is to generate attack messages that could not be easily detected by the IDS, by changing a small portion of the original attack packets of the dataset. Prior studies have only made a little dent in the development of effective packet-level adversarial attacks and that too in the core network. Early packet-level attacks frequently rely on arbitrarily applying preset mutations with a lot of trial-and-error to come up with a workable solution, which offers little to no theoretical direction and understanding. In recent publications there is only one other research work done for CAN network, which has generated the attack samples by perturbating directly on the ECU ID vector. As a result, their approach does not verify the actual impact of the adversarial attack samples created. The outcome of the attack could be impacted by changes in the values of ECU IDs of CAN messages. Hence, there is a need to validate the accuracy and reliability of adversarial attack samples in a car's network before converting attacks in feature space into genuine attack messages that can be sent to the CAN in a vehicle.

In this project, I plan to implement adversarial attack like Liuer Mihou but in the CAN network. As Liuer Mihou first trains a NIDS that mimics the decision boundaries of the target NIDS, my aim is to create a similar NIDS for CAN network. Then finding an optimal set of CAN packet mutations to fool the NIDS. Due to the limited time the scope of this project remains the creation of an DL or ML based NIDS and to see the effect of mutation performed on the input features on the NIDS.

1.4 Evaluation

To evaluate the target NIDS performance matrices, such as accuracy, precision, and F1 score are calculated. It is recommended, however, that an attack be tried on a real vehicle or a testbed to determine whether or not it is successful in order to have a realistic idea of how effective it is. Only then can the efficacy of an assault be practically determined. But the scope of this work lies within the software based NIDS not the actual system on a real vehicle.

Here, **delay** is chosen as the mutation operation to be performed. In light of this, our objective is to investigate whether or not there are any shifts in the performance matrices of NIDS as a result of adding delay to the malicious packets contained in the initial traffic.

1.5 Takeaways

There have been a number of studies conducted on adversarial attacks in the image domain and the core network domain. However, in this study an actual adversarial attack is carried out in the CAN network domain, which is a unique aspect of this work and represents its importance. CAN is a safety-critical application that makes use of the most recent innovations in technology, like as machine learning and deep learning, to provide better services. However, the most recent threats to these technologies also give birth to new attacks, such as adversarial attacks. As a result, this is a fruitful line of inquiry to pursue. Hence, this is a promising area to work on.

2. Background and Related Work

2.1 Background

ECUs in current automobiles communicate via the Controller Area Network (CAN) protocol. Modern car architecture relies on the CAN system to allow ECUs to manage engine operation, braking, and steering. The CAN communication protocol lacks encryption and authentication, making it vulnerable to cyberattacks.

CAN is used in various vehicles ranging from cars, trucks, buses, to boats, airplanes, and agricultural machinery. The below Fig. 1 shows the CAN bus system of a real vehicle.

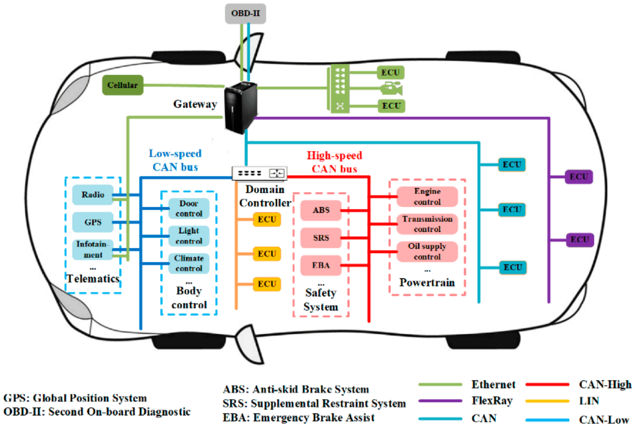


Fig.1 CAN bus system [13]

The Controller Area Network (CAN) is a message-based broadcast protocol that is a de facto standard for in-vehicle networks. It is a reliable and cost-effective communication protocol used for transferring data between different electronic control units (ECUs) in modern vehicles. The CAN bus is connected through a twisted pair of wires, where CAN-high and CAN-low carry the differential signal. The signal on the CAN-high wire is dominant when it is at 5V, while the signal on the CAN-low wire is dominant when it is at 0V. On the other hand, both wires carry the same voltage of 2.5V for recessive signals. These voltage levels are used to transmit data over the CAN bus between different ECUs. Despite its popularity, the CAN bus is also known to have several security vulnerabilities that can be exploited by attackers, making it essential to use intrusion detection systems to detect and prevent potential cyberattacks.

Some of the possible security threats to the CAN bus include message injection attacks, message alteration attacks, and message replay attacks. Message injection attacks involve an attacker sending messages that can trigger unintended actions, while message alteration attacks involve an attacker modifying messages on the bus to cause system malfunction or misbehavior. Message replay attacks involve an attacker recording a sequence of messages from the bus and then replaying them to produce an unintended effect.

Network Intrusion Detection Systems (NIDS) protect in-vehicle networks. NIDS employs machine learning methods to detect suspect CAN bus data. CAN bus NIDS have been designed using deep neural networks, LSTM models, and CNNs. NIDS can identify known cyberattacks but may not detect unexpected or hostile assaults. Thus, researchers have tested NIDS's resilience against adversarial assaults, which actively design attack signals to avoid detection. This research detects NIDS flaws and build more secure IDS for in-vehicle networks.

2.2 Related work on CAN IDSs

In the Controller Area Network (CAN), electronic control units (ECUs) are essential for proper functioning as they need to communicate with each other. Unfortunately, the current CAN communication protocol lacks security features such as encryption or authentication, which makes it vulnerable to cyber-attacks. Therefore, several measures have been proposed to prevent such attacks on the CAN.

In their research, Seo et al. [10] suggested an intrusion detection system (IDS) that utilized a generative adversarial network. The researchers aimed to construct a one-class classification model that could detect unseen attacks by relying solely on normal data. To train the model, they utilized a dataset generated from the CAN messages of an actual vehicle. Song et al. [11] proposed an intrusion detection system (IDS) that utilized a deep convolutional neural network (DCNN). They developed a data assembly module that could effectively convert the CAN bus data into a grid-like structure that matched the DCNN. However, recent studies have shown that deep learning models can be susceptible to adversarial attacks. The common approach involves adding a small amount of well-crafted perturbation to the input, causing the target classifier to misclassify the modified input.

The article [12] discusses the need for intrusion detection systems (IDS) in connected cars due to their communication capabilities and third-party applications. Deep neural network models have been proposed to detect attacks on the controller area network (CAN) bus, but it is unclear if they can withstand adversarial attacks. The authors use a genetic algorithm to generate adversarial CAN attack messages for Denial-of-Service (DoS), fuzzy, and spoofing attacks to test the effectiveness of the state-of-the-art IDS. The results show that the IDS is not effective in detecting the generated adversarial CAN attack messages, with detection rates significantly decreasing.

There are several other works which has developed the IDS for CAN mentioned in Table I below.

3. Problem Statement

Fig. 2 depicts the framework proposed for creating adversarial CAN messages. Our framework is intended to modify existing attack messages in feature space so that the target IDS cannot detect them. Thus, the proposed framework does not alter standard communications. However, for a given sequence of attack messages, attackers can inject fake messages or modify portions of attack messages so long as these modifications do not alter the attack's effects. The framework modifies a given set of attack messages iteratively using mutations i.e. delay. Possibly, the applied modifications would not be sufficient to circumvent the IDS. To validate the efficacy of attack messages, the modified attack messages could be resent to the target IDS. When the target IDS does not detect them, the proposed framework's modifications would allow them to circumvent the target IDS.the matter of any benefits which are generally available to the employees like health care, maternity leave, and pension.

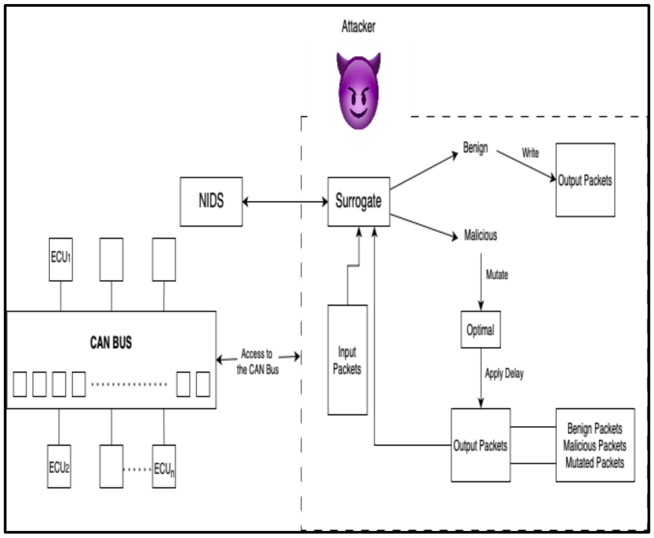


Fig. 2 System Model

3.1 System Model

The graphic that follows provides an illustration of the system model for the CAN bus. All of the ECUs are connected to a CAN bus, and the CAN bus is used to broadcast messages to the other components. The message is received and processed by the ECU that it was meant for. There is also a Network Intrusion Detection System (NIDS) present, which is likewise linked to the bus as a node, but it operates on a server that has greater capabilities than the ECUs. There is neither authentication nor an encryption method built into the CAN bus. The ML/DL-based NIDS that is being utilised is susceptible to attacks launched by attackers. The NIDS might be a complicated model that the attacker is not aware of; but, because it is operating in a grey box context, the attacker can have some concept of the model that is being utilized, which it then uses for the purpose of developing a surrogate model.

The basic pointers which for the system model are:

1. No Encryption in CAN bus
2. No Authentication in CAN bus
3. Each ECU broadcast the messages on the CAN bus
4. The NIDS is connected to the CAN bus a separate node

3.2 Threat Model

The attacker is assumed to have a few capabilities which are defined below.

Goal – The attacker wants the malicious packets to be classified as benign.

Knowledge – The attacker operates under a grey box setting where it does not have complete knowledge of the network features but knows the possible model used.

Capabilities

• The attacker can sniff both normal and malicious traffic since they have access to the CAN Bus. Because CAN lacks encryption and authentication, packets can be easily intercepted.

Author	**Model(s)**	**Dataset(s)**	**Features**	**Attack type**	**Performance**
Kang et al.	DNN using pre-trained DBN	Generated by OCTANE	CAN ID, DATA field	Normal/attack	95%
Loukas et al.	DL (RNN)	testbed: robotic vehicle	•Network Incoming • Network Outgoing • CPU • Disk Data • Encoder •Accelerometer • Power • Current • Attack label	Denial-of-Service (DoS), command injection, and malware (Net) attacks.	86.9% - validation
Seo et al.	GAN	Hyundai's YF Sonata vehicle	patterns of CAN IDs. Converted extracted CAN IDs into a simple image by encoding with one-hot-vector.	DoS, Fuzzy, RPM, and Gear attacks	95%
Lokman et al.	Deep Contractive Auto-encoders	CAN data collected from Toyota Camry, Hyundai Elantra, and Perodua Axia.	CAN ID, CAN Data	DoS, Fuzzy, Impersonate attack	91.0%
Zhang et al.	DL (anomaly detection)	Collected data on the real intelligent vehicle CAN bus using KvaserCAN Leaf Light V2	Extracted features using Busmaster from CAN log file (timestamp, ID, rpm, speed, throttle etc.)	spoofing and replay attacks.	97.0% and 98.0%
Zhu et al.	LSTM (anomaly based)	Collected from real vehicle	Time, data	spoofing, replay, and flooding attacks	80%
Yang et al.	RNN-LSTM	simulated ECU analog signals based on Multisim Version10.0	transceiver fingerprints of 50 ECUs, time sequence	Spoofing	98.76%
Song et al.	DCNN InceptionResnet	Created and publicly available HCRL dataset	transforms a CAN ID sequence into a 2-D grid data frame.	DoS, fuzzing, gear, and RPM	80%
ADVERSARIAL ATTACKS RELATED NIDS					
WISA (Choi, J et al.)	Inception Resnet	hcrl car-hacking dataset	transforms a CAN ID sequence into a 2-D grid data frame.	DoS, Fuzzy, Spoofing	Other measures
LM (anonymous)	AutoEncoder	Self-generated	Network traffic features	DoS, Fuzzy, Spoofing, Impersonation	

Table I. Literature Review on CAN IDSs.

• The attacker has the ability to change adverse traffic, insert forged packets, and then replay the changed traffic throughout the network.
• It is presumed that the attacker has had access to the on-board diagnostic (OBD-II) port or other remote network channels utilizing the vehicle's OBU for an infinite period of time.
• A surrogate NIDS that closely resembles the target NIDS's decision boundaries can be created in the attack scenario's grey box setting even when the attacker is unaware of the target NIDS's specifics.
• It is believed that the target NIDS will also categorise the packet as malicious or benign if the surrogate model has classified it as such. This is the attribute of transferability.
categorised by the platforms as either "driver/delivery partners" or "independent contractors." Workers as a result do not benefit from labour laws governing wages, hours, working conditions, and the ability to engage in collective bargaining.

3.3 Key Challenges

1. Creating adversarial input in such a way that it directly affects the input CAN traffic, making it a feasible attack. When it comes to the image domain, there is no dependence between the individual pixels that make up the image. Therefore, altering any one of the pixels will result in the production of a legitimate picture. However, when it comes to network traffic, whether it be a core network packet or a Can network packet, changing one element of the packet might have repercussions on other aspects of the packet. The relationships between the qualities are dependent on one another. It is possible that this will result in the creation of an invalid packet that is meaningless from a practical standpoint. Those packets could not be played back under any circumstances.

2. Selecting a surrogate model in such a way that it closely mimics the decision boundaries of the the target NIDS which could be done in a trial-and-error method.

3. The replaying of CAN messages is not feasible as per the scope of this project, so the adversarial packet generated may or may not be practical.

4. Proposed Solution

Fig. 3 shows the actual implementation steps to perform the mutations to generate the adversarial example. To perform the above steps, the code is written in python. The input to the model i.e. data.csv is the actual CAN traffic logged as a CSV file. There are four basic attributes/features of a real CAN traffic i.e. Timestamp, Arbitration_ID, DLC and Data. The below Fig. 4 shows the input features to our model. Apart from the basic four attributes the other three columns are added based on the traffic captured. The dataset used in this project is the online available dataset (HCRL - Car Hacking: Attack & Defense Challenge 2020 (hksecurity.net).

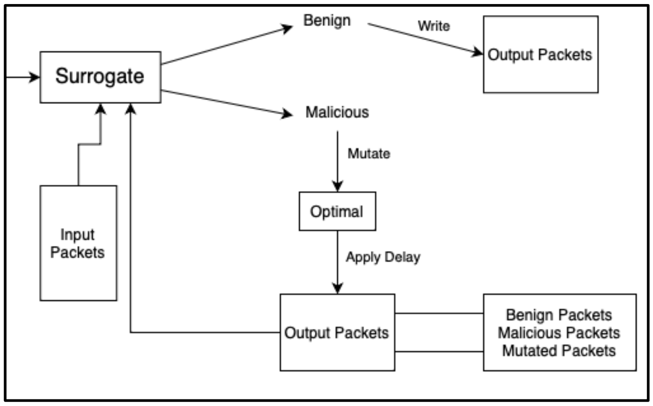
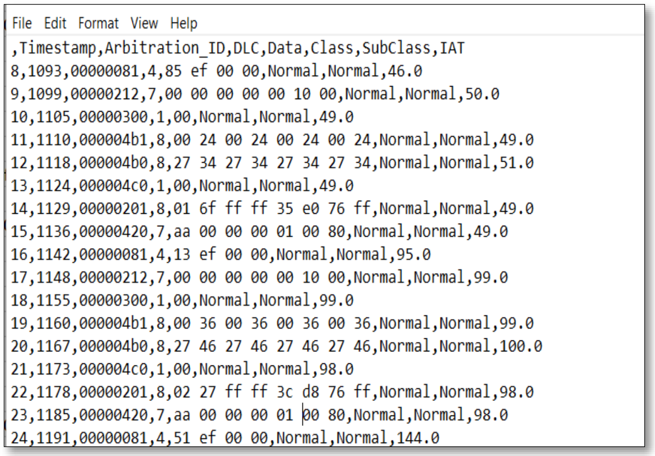


Fig. 3 Overview of the framework



```
,Timestamp,Arbitration_ID,DLC,Data,Class,SubClass,IAT
8,1093,00000081,4,85 ef 00 00,Normal,Normal,46.0
9,1099,00000212,7,00 00 00 00 00 10 00,Normal,Normal,50.0
10,1105,00000300,1,00,Normal,Normal,49.0
11,1110,000004b1,8,00 24 00 24 00 24 00 24,Normal,Normal,49.0
12,1118,000004b0,8,27 34 27 34 27 34 27 34,Normal,Normal,51.0
13,1124,000004c0,1,00,Normal,Normal,49.0
14,1129,00000201,8,01 6f ff ff 35 e0 76 ff,Normal,Normal,49.0
15,1136,00000420,7,aa 00 00 00 01 00 80,Normal,Normal,49.0
16,1142,00000081,4,13 ef 00 00,Normal,Normal,95.0
17,1148,00000212,7,00 00 00 00 00 10 00,Normal,Normal,99.0
18,1155,00000300,1,00,Normal,Normal,99.0
19,1160,000004b1,8,00 36 00 36 00 36 00 36,Normal,Normal,99.0
20,1167,000004b0,8,27 46 27 46 27 46 27 46,Normal,Normal,100.0
21,1173,000004c0,1,00,Normal,Normal,98.0
22,1178,00000201,8,02 27 ff ff 3c d8 76 ff,Normal,Normal,98.0
23,1185,00000420,7,aa 00 00 00 01 00 80,Normal,Normal,98.0
24,1191,00000081,4,51 ef 00 00,Normal,Normal,144.0
```

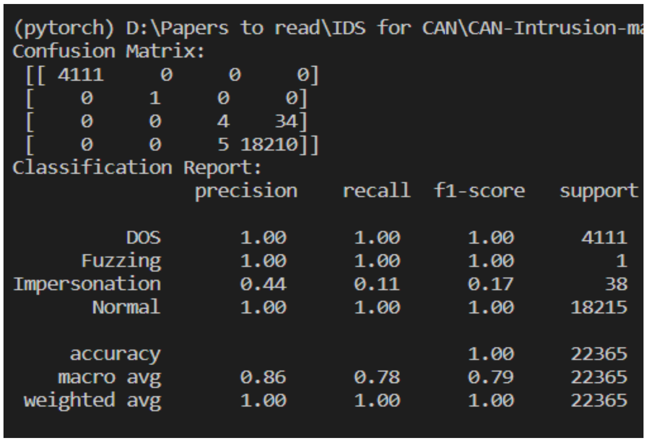
Fig. 4 Input features to the NIDS

1. This above dataset is then passed to the NIDS in our case. The model used is Decision Tree classifier. Also, the other models such as Random Forest, LSTM, MLP are used to compare the results which could be the potential Target NIDS.
2. The training of the DT model has been done using the above dataset and 80%-20% split is performed to create the test data.
3. The test file contains both benign and malicious packets. The next step is to select the malicious traffic only and add delay, thus making it an adversarial input.
4. Next step is to find a scheme based on which delay has to be added to the malicious packet’s timestamp field. I started with the “random delay “within a range to be used as mutation.
5. Then I selected a specific CAN ID on which the mutation will be performed.
6. The delay to be added with the timestamp field of CAN traffic, is chosen randomly between (low, high) value where low = Mean inter-arrival time of all the traffic of an ID in benign setting and high = Mean Inter arrival time of packets of an ID in the presence of attack traffic.

7. Then this mutated traffic (delay added in original traffic) is sent to NIDS again, to check the effect of random delay addition on the decision of NIDS.

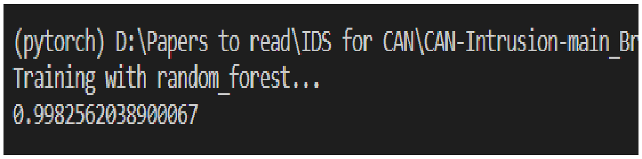
5. Completed Evaluation

Fig. 5 shows the performance of different ML model which could be used as potential target NIDSs. However, for the experimentation, Decision Trees and Random Forest is used.



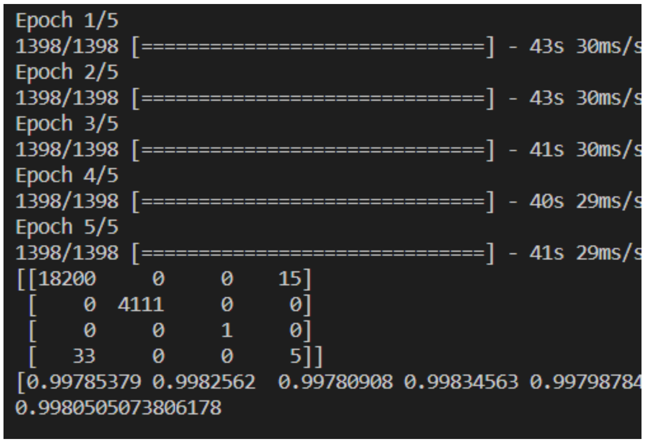
```
(pytorch) D:\Papers to read\IDS for CAN\CAN-Intrusion-ma
Confusion Matrix:
 [[ 4111     0     0     0]
 [    0     1     0     0]
 [    0     0     4    34]
 [    0     0     5 18210]]
Classification Report:
                precision    recall  f1-score   support

         DOS       1.00      1.00      1.00      4111
     Fuzzing       1.00      1.00      1.00         1
Impersonation       0.44      0.11      0.17        38
      Normal       1.00      1.00      1.00     18215

    accuracy                           1.00     22365
   macro avg       0.86      0.78      0.79     22365
weighted avg       1.00      1.00      1.00     22365
```

(a) Decision Trees



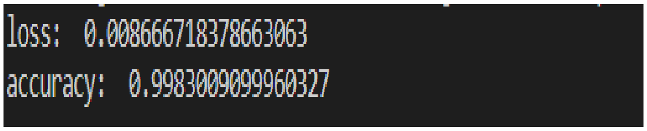
```
(pytorch) D:\Papers to read\IDS for CAN\CAN-Intrusion-main_Br
Training with random_forest...
0.9982562038900067
```

(b) Random Forest



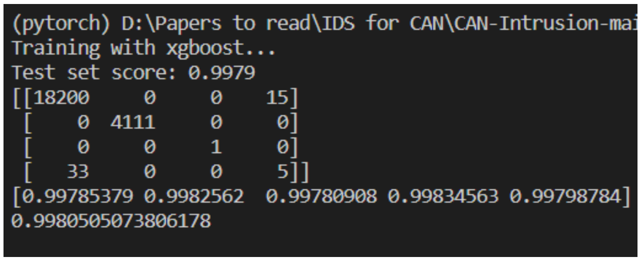
```
Epoch 1/5
1398/1398 [==============================] - 43s 30ms/s
Epoch 2/5
1398/1398 [==============================] - 43s 30ms/s
Epoch 3/5
1398/1398 [==============================] - 41s 30ms/s
Epoch 4/5
1398/1398 [==============================] - 40s 29ms/s
Epoch 5/5
1398/1398 [==============================] - 41s 29ms/s
[[18200     0     0    15]
 [    0  4111     0     0]
 [    0     0     1     0]
 [   33     0     0     5]]
[0.99785379 0.9982562  0.99780908 0.99834563 0.99798784
0.9980505073806178
```

(c) MLP



```
loss:  0.008666718378663063
accuracy:  0.9983009099960327
```

(d) LSTM



```
(pytorch) D:\Papers to read\IDS for CAN\CAN-Intrusion-mai
Training with xgboost...
Test set score: 0.9979
[[18200     0     0    15]
 [    0  4111     0     0]
 [    0     0     1     0]
 [   33     0     0     5]]
[0.99785379 0.9982562  0.99780908 0.99834563 0.99798784]
0.9980505073806178
```

(e) XGBoost

Fig. 5 Performance of ML classifier (a to e)

After adding delay, the above-mentioned accuracies were affected differently.

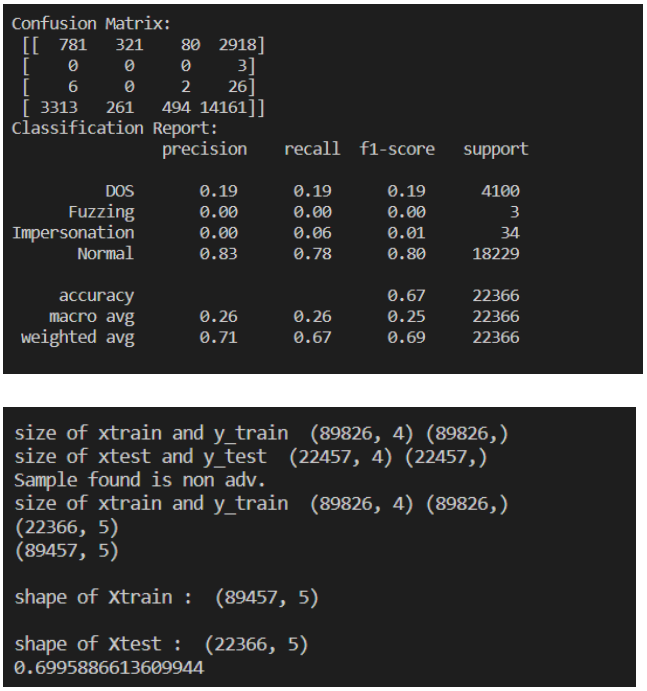
5.1 Mutation Operations

In this work, the mutation operation selected is the delay. Delay is applied by changing the inter arrival times of the CAN packets. To apply this mutation, two approaches are taken.

1. Random Delay
2. Range based Delay
3. IAT based Delay

Random Delay is a basic approach used to introduce delays where each of the malicious packet (DOS for experimentation) is selected and a random delay value is added to the timestamp of those packet. Then this modified dataset is again passed to the NIDS to find, and the results shows that there is no change in the performance of the NIDS. Adding random delays does not yield the desired results.

Range based Delay is the modified version of Random delay in which the delay value is calculated between a fixed range. This range is calculated based on the average IAT of different types of attack and normal packets. For example, if the normal periodicity of an ECU with ID 00000210 is 12ms and during the attack phase it become 50ms, then a random value between 12 and 50 is chosen to be added as delay. This approach has reduced the accuracy of DT and RF models. It dropped from 1.0 and 0.99 to 0.67 and 0.7 respectively as shown in Fig. 6. This shows that if the attacker tweaks the input set by adding delay to the input set, the NIDS will not perform good, it may allow malicious packet to bypass the system.



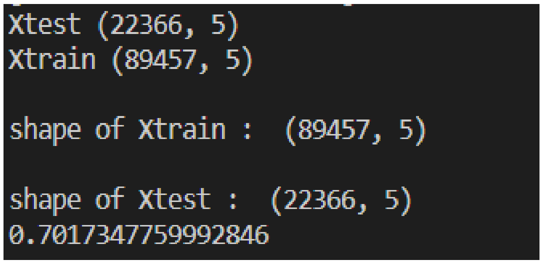
```
Confusion Matrix:
 [[  781   321    80  2918]
 [    0     0     0     3]
 [    6     0     2    26]
 [ 3313   261   494 14161]]
Classification Report:
                precision    recall  f1-score   support

         DOS       0.19      0.19      0.19      4100
     Fuzzing       0.00      0.00      0.00         3
Impersonation      0.00      0.06      0.01        34
      Normal       0.83      0.78      0.80     18229

    accuracy                           0.67     22366
   macro avg       0.26      0.26      0.25     22366
weighted avg       0.71      0.67      0.69     22366
```

```
size of xtrain and y_train  (89826, 4) (89826,)
size of xtest and y_test  (22457, 4) (22457,)
Sample found is non adv.
size of xtrain and y_train  (89826, 4) (89826,)
(22366, 5)
(89457, 5)

shape of Xtrain :  (89457, 5)

shape of Xtest :  (22366, 5)
0.6995886613609944
```

Fig. 6 Performance of DT and RF models for Range based Delay

The next approach taken is the IAT based delay. In this method, the mean inter arrival time of each type of packet of each ID is calculated and according to the difference in the values, delay is added. For example, the normal packet of ID 00000081 has mean IAT of 49.5 ms and the same traffic in attack scenario has mean IAT of 8ms. Then a delay of (50-8 = 42) is required to make an attack packet look like a normal packet. Thus, in this way each type of traffic is mutated for a certain ECU ID. To see the effect, packets of all ECUs are not mutated together in one go.

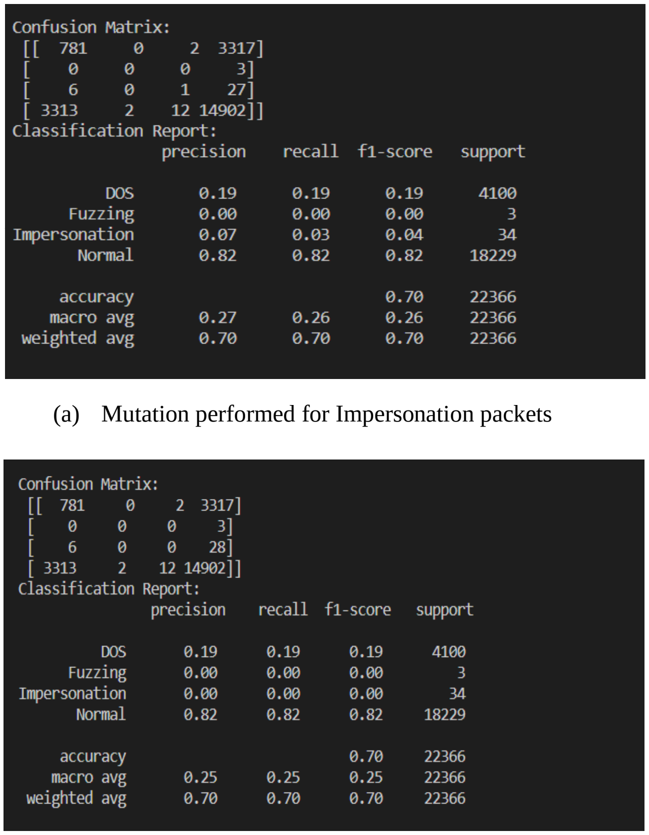
Again, in the IAT approach, the mutations are not performed on all types of attacks together. Firstly, appropriate delay has been added to only DoS packet, then only to impersonation packets and lastly to both together. It has been observed that accuracies have been reduced for each type of attack mutation. For the all the cases, the NIDS earlier showed 100% accuracy which later reduced to 70% in case of Decision Tree as the target model. Similarly, for RF model, it decreases by 30% as shown in Fig. 7. Thus, the model is not able to predict well.



```
Xtest (22366, 5)
Xtrain (89457, 5)

shape of Xtrain :  (89457, 5)

shape of Xtest :  (22366, 5)
0.7017347759992846
```

Fig. 7 Performance of RF for IAT based Delay

Apart from the accuracies, other metrics such as precision, recall and F1scores are also calculated for DT model as shown in the Fig. 8.



```
Confusion Matrix:
 [[  781     0     2  3317]
 [    0     0     0     3]
 [    6     0     1    27]
 [ 3313     2    12 14902]]
Classification Report:
                precision    recall  f1-score   support

         DOS       0.19      0.19      0.19      4100
     Fuzzing       0.00      0.00      0.00         3
Impersonation      0.07      0.03      0.04        34
      Normal       0.82      0.82      0.82     18229

    accuracy                           0.70     22366
   macro avg       0.27      0.26      0.26     22366
weighted avg       0.70      0.70      0.70     22366
```

(a) Mutation performed for Impersonation packets

```
Confusion Matrix:
 [[  781     0     2  3317]
 [    0     0     0     3]
 [    6     0     0    28]
 [ 3313     2    12 14902]]
Classification Report:
                precision    recall  f1-score   support

         DOS       0.19      0.19      0.19      4100
     Fuzzing       0.00      0.00      0.00         3
Impersonation      0.00      0.00      0.00        34
      Normal       0.82      0.82      0.82     18229

    accuracy                           0.70     22366
   macro avg       0.25      0.25      0.25     22366
weighted avg       0.70      0.70      0.70     22366
```

(b) Mutation performed for DoS packets

Fig. 8 Performance of DT for IAT based Delay

For attacker, the goal is to increase the false negatives in order to successfully fool the NIDS. Hence, False negative rate (FNR) is a better measure to know the effects of adding delay as a mutation.

FNR = False Negative / (False Negative + True Positive)
In case of before mutation, DoS shows FNR of 0, Fuzzing shows FNR of 0.33, Impersonation shows 0.82 and a very small FNR of 0.0004 in case of normal messages. After applying the range based delays, the DoS FNR increases from 0 to 0.79. For normal packets, there is an increase of 0.2 and for fuzzing and impersonation FNR slightly increased. This shows that the NIDS is incorrectly classifying the DoS packets as benign after adding delay to it.
Similarly for the IAT based delays, FNR for DoS came out to be 0.80, for impersonation 0.96 and for normal packets its, 0.51. This shows that the NIDS generating more False Negatives in case of IAT based delays than the Range based for DOS mutation and a similar result is seen for impersonation mutation.

6. Conclusion and Future Work

.

Adding delay as mutation definitely shows the reduction in the performance of the NIDS. The delay, when applied randomly, does not show any significant change in the FNR rate. In case of Range based delay, the FNR rate shows that the model is now misclassifying the malicious packets as benign. However, the FNR increases most kn case of IAT based delay. The accuracies do not decreases much after the mutation fro DoS and Impersonation as only a single ECU is used to perform the attack, which might be learned by the model. But if distributed DoS or Impersonation occurs, then it may reduce the accuracies as well.

The above results would be more practical if the generated adversarial traffic is again replayed. However, due to the constraints such as limited time and the unavailability of a working testbed, this could not be a part of this project. To further extend this work, replaying could be a better approach. This step will help to ensure that the adversarial traffic is effective in compromising the security of the CAN network. By replaying the traffic on a testbed, the impact of the adversarial attack can be measured and assessed in a realistic environment. This will provide valuable insights into the effectiveness of the proposed approach and the robustness of the anomaly detector.

7. REFERENCES

[1] T.J. Park, C.S. Han, and S.H. Lee, "Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system," Mechatronics, vol. 15, no. 8, pp. 899-918, 2005.

[2] F. Yu, D.F. Li, and D. Crolla, "Integrated vehicle dynamics control—state-of-the art review," in Proceedings of the Vehicle Power and Propulsion Conference, IEEE, 2008.

[3] Cyber-attacks over CAN must be avoided because they pose a serious risk to the safety of drivers. To identify suspicious communications over CAN, many intrusion detection systems (IDS) have been developed [3].

[4] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle."

[5] R. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," Computer, vol. 35, no. 4, pp. 27-30, 2002.

[6] O. Ibitoye, R. Abou-Khamis, A. Matrawy, and M.O. Shafiq, "The threat of adversarial attacks on machine learning in network security-a survey," arXiv preprint arXiv:1911.02621, 2019.

[7] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," arXiv preprint arXiv:1607.02533, 2016.

[8] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," arXiv preprint arXiv:1702.05983, 2017.

[9] M.J. Kang and J.W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," PloS One, vol. 11, no. 6, article no. e0155781, 2016.

[10] Seo, E., Song, H. M., & Kim, H. K. (2018, August). GIDS: GAN based intrusion detection system for in-vehicle network. In 2018 16th Annual Conference on Privacy, Security and Trust (PST) (pp. 1-6). IEEE.

[11] H.M. Song, J. Woo, and H.K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," Vehicular Communications, vol. 21, article no. 100198, Mar. 2020.

[12] J. Choi and H. Kim, "On the Robustness of Intrusion Detection Systems for Vehicles Against Adversarial Attacks," in Information Security Applications: 22nd International Conference, WISA 2021, Jeju Island, South Korea, August 11–13, 2021, Revised Selected Papers 22, Springer International Publishing, 2021, pp. 39-50.

[13] H. Zhang, X. Meng, X. Zhang and Z. Liu, "CANsec: A Practical In-Vehicle Controller Area Network Security Evaluation Tool," in Sensors, vol. 20, no. 17, p. 4900, 2020.