Fast - IDSGAN

CMPE 258, Team 3

Poojitha Vurtur Badarinath (014505660)

Shivani Suryawanshi (014490944)

Sughosh Krishnamurthy (014370954)

Abstract

Ubiquitous cyber-intrusions endanger the security of our devices constantly. They may bring irreversible damages to the system and cause leakage of privacy. Thus, Intrusion detection systems (IDS), as one of the important security solutions, are used to detect network attacks. With the extensive applications of traditional machine learning algorithms in the security field, intrusion detection methods based on machine learning techniques have been developed rapidly. However, Intrusion Detection Systems are weak against adversarial attacks, and research is being done to prove the ease of breaking these systems. To improve the detection system, more potential attack approaches should be researched. Our goal in this project is to design a framework called Fast - Intrusion Detection System Generative Adversarial Network (F-IDSGAN) to create adversarial attacks, which can deceive and evade any IDS. Based on the CICIDS2017 dataset, the developed model is able to generate different attacks and excellent results are achieved.

Introduction

The intrusion detection system (IDS) is an essential tool in cybersecurity which detects and defends against malicious network traffic. The main goal of IDS is to classify normal and malicious network records. Machine-learning algorithms have been widely used in the improvement of intrusion detection systems due to their high flexibility and achieved good results. However, IDS fails to classify adversarial malicious traffic. And the Generative Adversarial Networks (GAN) are the potential chosen method for such adversarial attacks.

In this project, we implement a GAN which generates adversarial attacks against IDS. The goal of the model is to generate malicious traffic that can pass through the IDS. Extra Tree Classifier is used as the black box IDS, which classifies malicious and normal traffic. We design and improve the generator and the discriminator based on Wasserstein GAN.

This is how the F-IDSGAN framework operates; a generator generates malicious traffic and mixes it in with normal traffic, and sends the traffic to the IDS and the discriminator. The discriminator and generator train in parallel. The discriminator tries to identify each piece of traffic it encounters and compares its decision with the IDS, and adjusts its weights accordingly. Meanwhile, the generator learns how it is performing based on the decision the discriminator made. Eventually, the generator learns how to disguise a malicious input and creates a near-perfect model, which completely fools the discriminator.

We used the CICIDS2017 dataset which contains benign and the most up-to-date common attacks, which resembles the true real-world data. Attacks include Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS. The feature set contains 78 network flow features.

In summary, our primary contributions are the following:

- We design F-IDSGAN, to generate malicious traffic to attack IDS. In the generation, the modification will not affect the attack functionality of the original attack.
- To mimic IDS in the real world, ensemble learning technique - extra tree classifier is used.
- F-IDGAN performed well in the experiments. The detection rate to adversarial examples approaches 0 in the IDS model. It indicates that most of the adversarial attacks can pass through the detection of black box IDS.

Related Work

Over the last decade or so there has been a rapid development of machine learning algorithms in the field of cyber security. When posed as a classification problem there are plenty of state of the art methods that caters towards this purpose. The models vary from classical machine learning framework to deep neural networks that achieve a good performance in classification tasks.

With development of Generative Adversarial Network (GAN) by Goodfellow et al.[1] has led to rapid usage of GANs across all domains of deep learning and one such area is cyber security to generate adversarial samples that break the detection system in order to develop more robust intrusion detection system. Subsequently there has been plenty of research being done towards generating adversarial samples that break the detection system. Grosse et al.[2] proposed to apply the algorithm based on the forward derivative of the attacked neural networks to craft adversarial Android malware examples with the function of the malware preserved. Anderson et al.[3] proposed a reinforcement learning algorithm with a set of the functionality preserving operations was used for generating adversarial malware examples. Rosenberg et al.[4] generated the adversarial examples combining API call sequences and static features with an end-to-end attack generation framework. Zhou et al.[5] crafted the adversarial spam with the adversarial SVM model and researched how to construct a more robust spam filter. Hu et al. [6] proposed a GAN framework to generate adversarial malware examples for the black-box attacks. They also leveraged a new model to generate some adversarial API sequences which would be inserted into the original API sequences of malware to form the attacks, aiming at bypassing Recurrent Neural Networks (RNN) detection systems.

Although the adversarial technology has been widely applied in malware detection, there is little academic research about the adversarial malicious traffic examples against IDS. One such research was done by Z Lin et al.[7] where their proposed method generates adversarial traffic which attacks the IDS in an attempt to break it. On similar ground our project model constructs the architecture of the generative adversarial networks for the adversarial attack examples targeting at IDS and successfully attacks black- box IDS model.

In comparison our model is much faster than IDSGAN proposed by Z Lin et al.[7] as our model effectively converges with less than 30 epochs with each epoch taking <5mins and one key reason for this is modified Wasserstein loss that is implied in our Fast-IDSGAN model. Also, our F-IDSGAN model has the capacity to generate 14 different attack types viz a viz 4 different attack types as proposed by Z Lin et al.

Dataset and Preprocessing

The CICIDS-2017 dataset used in this project is generated by the “Canadian Institute of Cybersecurity”. This dataset is generated over a span of 5 days and contains up to date attack categories which resembles the real world normal and attack data. The dataset has both packet-based and bidirectional flow-based format. This consists of 79 attributes which includes one target attribute as well. Target attribute consists of one normal and 14 attack class labels totaling 15. This dataset consists of 2.8 million records which is generated over 5 days.

Fig 1: Showing all the attack types covered in dataset

Data Preprocessing

- The data generated on all the 5 days is merged into one large dataset
- All the NULL values which few of the attributes had are replaced with zeros
- Few of the outputs with INF values are replaced with MAX value to let the model continue with computation and not result in nan output
- Unfriendly characters are removed in the target label
- Target label is encoded as -1 for normal samples and +1 for attack samples
- Eight irrelevant columns below are removed from the dataset
    - Fwd Avg Bytes/Bulk
    - Fwd Avg Packets/Bulk
    - Fwd Avg Bulk Rate
    - Bwd Avg Bytes/Bulk
    - Bwd Avg Packets/Bulk
    - Bwd Avg Bulk Rate
    - Bwd PSH Flags
    - Bwd URG Flags
- Random under sampling of the real data is performed to bring the number of records in real samples close to the records in the attack data to balance the dataset. After undersampling, the number of normal samples are reduced to half a million records which is close to the number of attack records
- Features are divided into functional and non-functional based on whether the change in that feature’s value invalidates the attack type or not. 25 features are considered

functional and rest are considered nonfunctional. Only the nonfunctional features are modified to generate new attacks

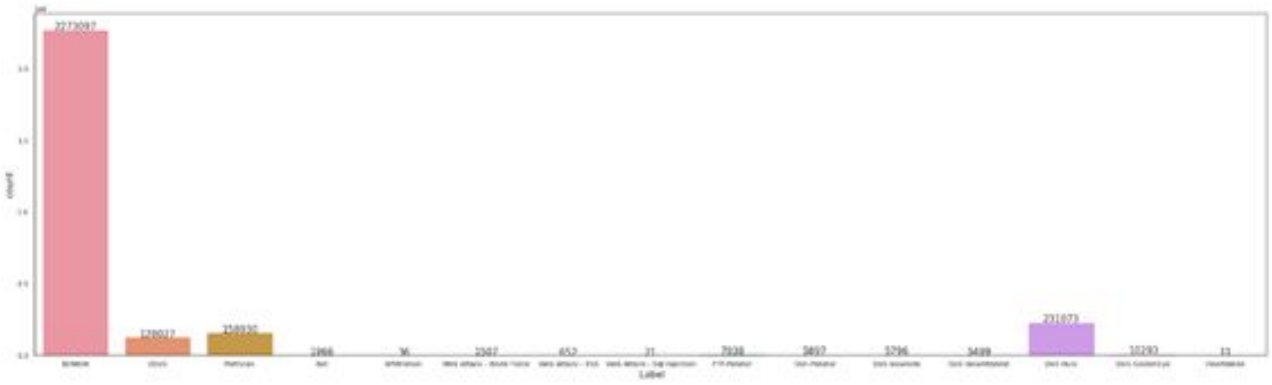

Fig 2: Showing the number of records of each attack type and the normal data

- For evaluation purposes the resampled - balanced malicious samples and normal samples are bifurcated into training data and cross validation data with the ratio of 4 : 1. In essence the training data contained ~400000 samples of original normal and attack samples and ~100000 samples of each cross validation data.

Methodology

Our goal of this project is to implement a fast deep neural network which will generate malicious samples that passes through the Detection System undetected. In order to achieve this objective we have implemented a Generative Adversarial Network to generate adversarial malicious samples that ensures the sanctity of attack data by keeping the functional features unaltered that can pass through the Detection System. This implementation comprises three different models, an intrusion detection system(IDS), discriminator model(D) and generator model(G).

The IDS is a pre-trained classification model that has a well defined ability to predict malicious and normal traffic. For this purpose we have implemented an ExtraTree classification model that has a near 100% accuracy in detecting traffic, providing a strong incentive towards breaking the detection system. However, in real world scenarios the IDS can be any state of the art intrusion detection system and, for this any type of classification model that achieves good accuracy can be built with fundamental machine learning principles of classification.

Theoretically speaking, Generative Adversarial Network is a zero sum game between two neural network models where one model learns to generate samples (Generator) that fools the other model (Discriminator). On this grounds in our project we have designed two neural networks models to achieve this functionality, here the generator learns to generate malicious samples by taking in actual attack samples along with noise that can pass through the IDS. For the generator to learn to deceive the IDS discriminator plays a vital role by providing feedback of the predicted traffic samples by the IDS. This feedback basically ensures how the generator should learn to generate malicious samples which shall pass through the IDS and based on this architectural design the model is optimized over their gradients to learn and achieve our goal. Architectural design and data flow of the model implementation is given below.

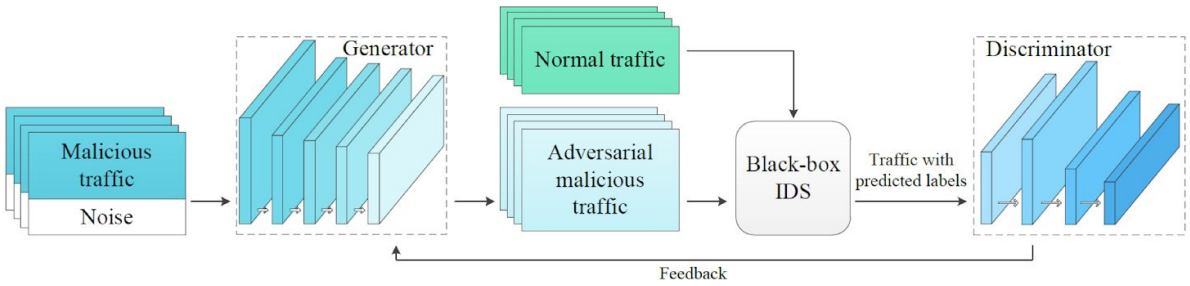

Fig 3: Design Architecture. The traffic data with different attack types is divided into malicious and normal data. The original malicious traffic with noise is used for training of the generator and the predicted traffic labels of the adversarial traffic and normal data by the IDS is trained with discriminator to provide feedback to the generator to improve the adversarial attacks.

Experiments and Algorithm

In this section of the report we shall thoroughly go through each of the models along with their implementation algorithm. The basic structure of our F-IDSGAN is explained in the above section and in order to achieve this we have implemented something on similar grounds to Wasserstein-GAN (W-GAN). Theoretically, W-GAN gives us the measure of how fake/real are the adversarial samples and this theoretical concept helps us design our loss functions in order to achieve our objective. Our F-IDSGAN core design architecture has three models, intrusion detection system, generator and discriminator which are sequentially explained below.

Intrusion Detection System

Intrusion detection system (black box IDS) is a simple classification model which is pre-trained to detect malicious and normal traffic. For this purpose, we have implemented an ExtraTree classifier and trained the model using both original attack samples - T_{ATTACK} and original normal samples - T_{NORMAL}. To increase the robustness and accuracy of the model it was trained with augmented T_{ATTACK} data and the augmentation technique adopted was increased redundancy of T_{ATTACK} samples. This augmentation was mainly adopted because the model was underperforming when it was tested individually with T_{ATTACK} and T_{NORMAL} samples, thus this augmentation technique simply involved training the classifier with two times more T_{ATTACK} samples by keeping the number of T_{NORMAL} samples unchanged. This technique delivered optimum performance when it was tested with individual T_{ATTACK} and T_{NORMAL}.

Our ExtraTree classifier was trained with around 900,000 T_{ATTACK} samples and 250,000 T_{NORMAL} samples with following training parameters:

bootstrap=False, ccp_alpha=0.0, class_weight=None,

criterion='gini', max_depth=None, max_features='auto',

max_leaf_nodes=None, max_samples=None,

min_impurity_decrease=0.0, min_impurity_split=None,

min_samples_leaf=1, min_samples_split=2,

min_weight_fraction_leaf=0.0, n_estimators=100,

n_jobs=None, oob_score=False, random_state=0, verbose=0,

warm_start=False

To evaluate the performance of our IDS model we used two scoring metrics F1 score and MSE for individual T_{ATTACK} and T_{NORMAL} samples which gave a near 0 MSE for both. In our F-IDSGAN the adversarial A_{ATTACK} samples along with the original T_{NORMAL} samples are passed to the IDS to obtain the predicted P_{ATTACK} and P_{NORMAL} labels of the samples.

Generator

The main objective of the generator is to learn to generate adversarial attack samples - $A_{ATTACKS}$ that can pass through the IDS undetected. For this purpose we designed a multi-layer neural network model using Keras sequential that consists of 5 Dense layers with each layer batch normalization. The model summary is given in Fig 3.

To transform original T_{ATTACK} samples into adversarial A_{ATTACK} samples, the generator is trained with m-dimensional original T_{ATTACK} samples added with m-dimensional noise - N. The original T_{ATTACK} samples are preprocessed and to be consistent, the elements in N samples are composed of random numbers in uniform distribution within the range [0, 1]. To make training faster there are no activation layers for any of the layers and to make sure the output of the generator retains m-dimension as original T_{ATTACK} samples dot product is taken wrt m-dimension in the last dense layer. In addition to implementing F - IDSGAN there are several tricks involved in processing the adversarial A_{ATTACK} samples. To restrict the output elements into the range of [0, 1], the element which is above 1 is set as 1 and the element which is below 0 is set as 0. Also to ensure the sanctity of the adversarial A_{ATTACK} samples they are multiplied with non functional feature vector NF_V and are added to the product between original T_{ATTACK} samples and functional feature vector F_V.

$$A_{ATTACK} = (T_{ATTACK} \times F_V) + (G(T_{ATTACK,}, N) \times NF_V).....(1)$$

Discriminator

The key functionality of the discriminator in our F-IDSGAN is to learn to imitate IDS in order to provide feedback to the generator based on the predicted malicious samples P_{ATTACK} and predicted normal samples P_{NORMAL}. This imitation helps the generator training by enabling it to learn generate adversarial A_{ATTACK} samples that can bypass the IDS by classifying the output of the generator and supplying it as feedback. For this purpose a multi-layer neural network was designed with 3 layers and 1 fully connected layer with each layer having ReLU activation by using Keras sequential model whose network design is as given in fig 4.

In order to imitate the IDS, the predicted labels P_{ATTACK} and P_{NORMAL} of the adversarial A_{ATTACK} samples and original T_{NORMAL} samples are compared with the discriminator classification of the adversarial A_{ATTACK} samples and original T_{NORMAL} samples by using predicted labels P_{ATTACK} and P_{NORMAL} as target labels for the discriminator to learn. The procedure for making the discriminator imitate the IDS is also depicted in the below figure.

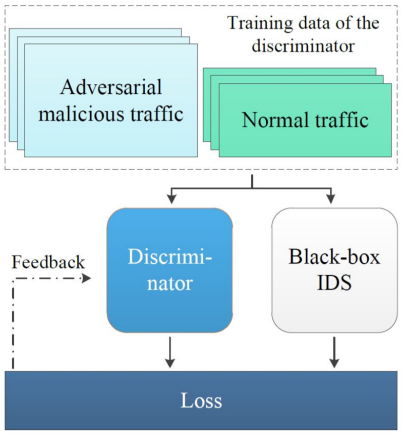

Fig 4: Discriminator Training

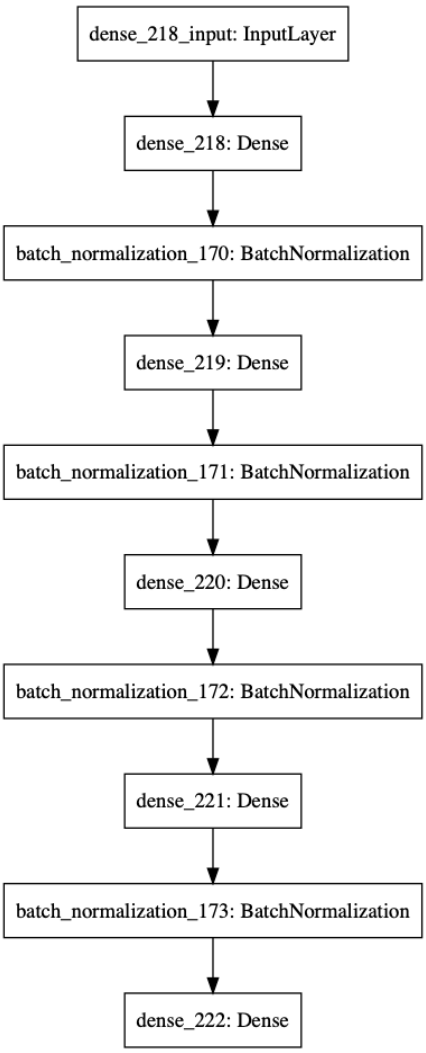

Fig 5: Generator Model

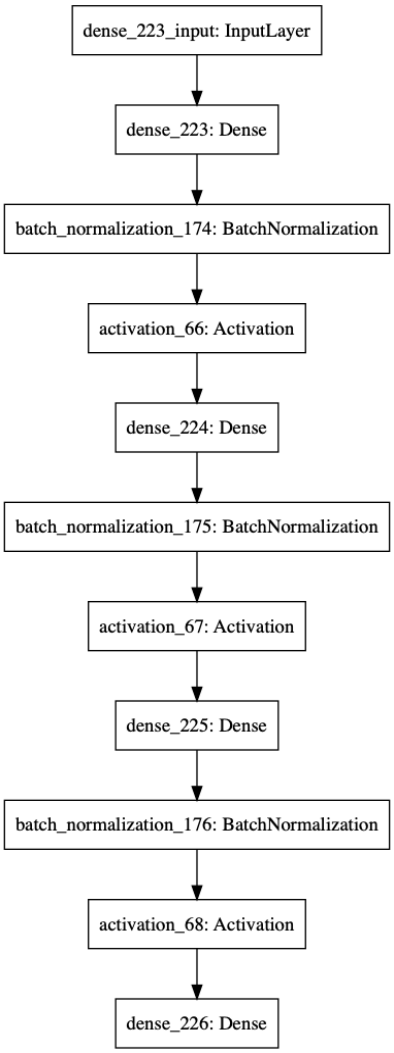

Fig 6: Discriminator Model

F-IDSGAN Training

Training GANs are like running on thin ice as it requires parallel training of two different neural net models with different functionality. Thus, making it hard for loss propagation to each model by ensuring neither of the models overpowers making losses to explode. For this purpose the training of the entire F-IDSGAN network was completely done using tensorflow v2.

In training of the generator, the adversarial A_{ATTACK} samples are passed through the discriminator and the gradient update is done based on the classified A_{ATTACK} samples by the discriminator. The loss function for the generator is as below.

$$L_G = E_{T \in T_{ATTACK}} D(A_{ATTACK}).....(2)$$

where, E represents the labels of traffic samples belonging to original malicious traffic samples.

For the discriminator training, adversarial A_{ATTACK} samples and original T_{NORMAL} samples are the training data for the discriminator. The discriminator loss is calculated by classification labels of the discriminator and the predicted labels obtained of the IDS. Thus formally given by the below equation.

$$D_G = (P_{ATTACK} \times D(A_{ATTACK})) - (P_{NORMAL} \times D(T_{NORMAL}))........(3)$$

where, P_{ATTACK} and P_{NORMAL} are IDS predicted labels and A_{ATTACK} are adversarial traffic generated by the generator and T_{NORMAL} are original normal traffic samples.

For optimizing the gradients of these losses RMSProp optimizer was used as similar to Wasserstein GAN and the key difference from W-GAN optimization are weight clipping, loss computation and training the discriminator twice for each epoch. The training algorithm of our F-IDSGAN is as below.

Algorithm: F-IDSGAN Training; epochs: 30, l_rate: 5e-6, batch_size = 1024

Input:

m-dimensional noise samples - N ;

Original traffic signals T_{NORMAL} and T_{ATTACK}

Output:

Generated Adversarial traffic samples A_{ATTACK}

Steps:

1. Initialize IDS, Generator (G) and Discriminator (D) and generate noise samples N;
2. for epochs in total_epochs:
3. for batches in steps_per_epoch:
4. Generate A_{ATTACK} sample from eq(1)
5. Update G gradients using eq(2)
6. Classify A_{ATTACK} and T_{NORMAL} of the D w.r.t IDS predictions
7. Update D gradients using eq(3)
8. end for
9. end for

Results and Evaluation

We note that the parameter settings and configurations for F-IDSGAN training are described in the Experiments and Algorithm section. With those parameters, the generator and discriminator were able to achieve -59.3153 and -31.7972 loss respectively. Fig(7) Illustrates that loss of generator and discriminator decreases but the loss of IDS increases after each iteration.

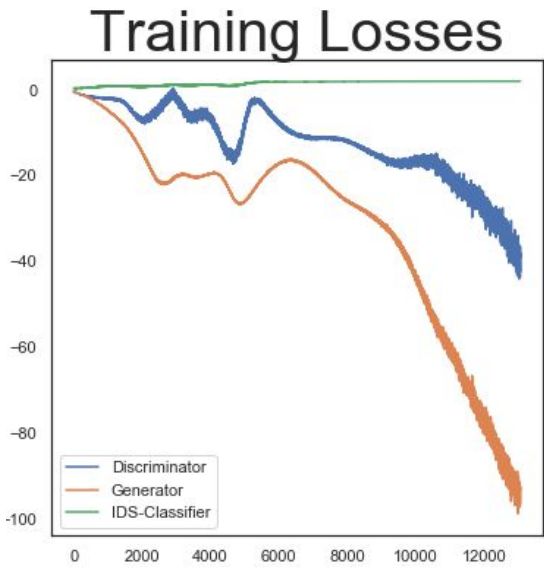

Fig 7: Training Loss

For the evaluation of F-IDSGAN, the detection rate and the evasion increase rate is measured. The detection rate (DR) reflects the proportion of correctly detected malicious traffic records by the black-box IDS to all of those attack records detected, directly showing the evasion ability of the model.

$$DR = \frac{No.\ of\ correctly\ detected\ attacks}{No.\ of\ all\ the\ attacks}$$

The evasion increase rate(EIR) as the rate of the increase in the undetected adversarial malicious traffic examples by IDS compared with the original malicious traffic examples. The original detection rate and the adversarial detection rate represent the detection rate to the original malicious traffic records and that to the adversarial malicious traffic records, respectively.

$$EIR = 1 - \frac{Adversarial\ detection\ rate}{Original\ detection\ rate}$$

A lower detection rate means more malicious traffic evades the black-box IDS, directly reflecting the stronger ability of F-IDSGAN. On the contrary, a lower evasion rate reflects more

adversarial examples that can be detected by the black-box IDS. So, the motivation for F-IDSGAN is to obtain a lower detection rate and a higher evasion increase rate.

The low detection rate and the high evasion increase rate obtained in the test reflect that IDSGAN shows its great capacity for the adversarial attack in the experiment. Fig 8. shows that the detection rate for adversarial attacks is 0, meaning generated attacks are able to pass through IDS undetected. Also, Evasion increase rate is 1 means 100% of the adversarial attacks can evade the detection of IDS model.

```
Few Original Malicious attacks Samples as Predicted by the IDS:
 ----------------------------------------------------------------------
 [['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']
 ['MALICIOUS ATTACK!!']]
 ----------------------------------------------------------------------
 Detection System MSE:  0.0
 Detection Rate:  1.0


Converted Adversial Samples as Predicted by the IDS:
 ----------------------------------------------------------------------
 [['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']
 ['NORMAL DATA']]
 ----------------------------------------------------------------------
 Detection System MSE:  2.0
 Detection Rate:  0.0

 Evasion Increase Rate (EIR):  1.0
```

Fig 8: Comparison of detection rate for adversarial attacks and normal attacks

IDS achieves 99% accuracy on the normal test dataset. When adversarial attacks are passed along with the test data accuracy of IDS decreases to 69%. IDS fails to identify the adversarial attacks and classifies them as benign. A total of 111530 adversarial attacks are passed to the IDS, and only around 9000 attacks are detected. Fig 8. shows that the true positive rate is around 0.5 and the area under the curve is 0.75. Fig 9. illustrates that the IDS model can classify more accurately with around 100% accuracy.

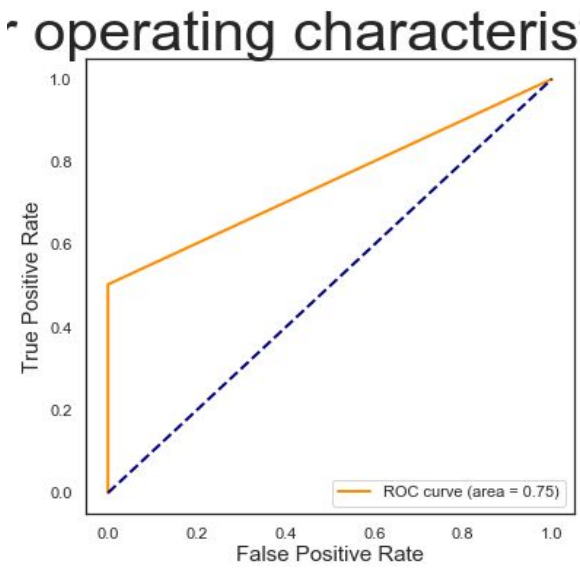

Fig 8: ROC for test data including adversarial attacks

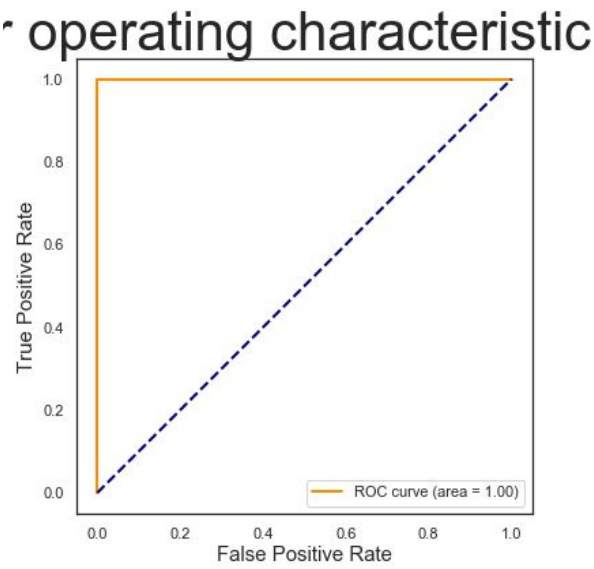

Fig 9: ROC for normal test data

From the above figures, it is clear that the IDS fails to identify adversarial attacks. Thus, the results of the experiments show that F-IDSGAN can generate adversarial examples that can evade detection.

Conclusion and Future Scope

With this approach we were successfully able to generate adversarial malicious samples which held the sanctity by maintaining the attack characteristics of traffic data. The results achieved from our F-IDSGAN model are extraordinary. As mentioned in the results and evaluation section our model was evaluated on validation data which both the GAN network and IDS model had not seen before and our main scoring criteria to evaluate the generated adversarial samples is evasion increase rate (EIR). Due to high performance of the IDS model it makes hard for our F-IDSGAN model to break the detection system in accordance with the latter, implementation of modified Wasserstein loss has helped us achieve this objective. The Detection Rate (DR) score for original malicious traffic samples is 1.0 implying that all the malicious samples were caught by the IDS. In contrast, when the original attack samples were converted into adversarial samples none of the adversarial attack samples were caught by the IDS and the DR score dropped to 0 with 100% increase in EIR score. Thus, the results achieved from our F-IDSGAN model are quite impressive.

Apart from this we also experimented the validity of our validation results of our F-IDSGAN, upon obtaining the adversarial attack samples as per equation (1) in Experiments and Algorithm section the non functional vector NF_V was replaced with functional vector F_V and as expected all the samples were caught by the IDS implying that our F-IDSGAN had successfully learned to generate non functional features without being affected by the functional features that holds the very basic nature of attack data which also ensure that the adversarial attacks are for definite attack signals which are failed to be detected by the IDS.
Note: This method is not implemented in the code; it was just used as a test case to evaluate our F-IDSGAN model.

Although we were able to achieve good results towards generating adversarial traffic samples that breaks a robust detection system. It would be good experimental research practice for our F-IDSGAN to be tested on various other detection system models in order to draw a comparative analysis on the EIR score of other IDS models.

Also, one interesting research would be to implement our F-IDSGAN as an active learning network where the detection system is simply another neural network which can be made trainiable based on EIR threshold. This research would require another loss computation for the neural net IDS obtained by generated samples to the discriminator classification. This field of active learning neural networks are relatively new and would require a bit of literature survey. But this theoretical idea of active learning of F-IDSGAN definitely has a positive impact in the field of cyber security.

NOTE: Implementation code of the above described F-IDSGAN can be found in the following github repository https://github.com/Poojithavb/CMPE258-Project/blob/master/F_IDSGAN.ipynb

References

[1]. [Goodfellow et al. 2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley,D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672– 2680.

[2]. [Grosse et al. 2016] Grosse, K.; Papernot, N.; Manoharan, P.; Backes, M.; and McDaniel, P. 2016. Adversarial perturba- tions against deep neural networks for malware classifica- tion. *arXiv preprint arXiv:1606.04435*.

[3]. [Anderson et al. 2017] Anderson, H. S.; Kharkar, A.; Filar, B.; and Roth, P. 2017. Evading machine learning malware detection. *Black Hat*.

[4]. [Rosenberg et al. 2018] Rosenberg, I.; Shabtai, A.; Rokach, L.; and Elovici, Y. 2018. Generic black-box end-to-end at- tack against state of the art api call based malware classifiers. *arXiv preprint arXiv:1804.08778*.

[5]. [Zhou et al. 2012] Zhou, Y.; Kantarcioglu, M.; Thuraising- ham, B.; and Xi, B. 2012. Adversarial support vector ma- chine learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1059–1067. Beijing, China: ACM.

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

[7]. [Z Li et al. 2018] Lin, Z.; Shi, Y.; Xue, Z.; IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. *arXiv:1809.02077v3.*