Generative Adversarial Networks For Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems

Muhammad Usama*, Muhammad Asim*, Siddique Latif†, Junaid Qadir*, Ala-Al-Fuqaha‡
*Information Technology University, Punjab, Pakistan.
†University of Southern Queensland, Australia.
‡Hamad Bin Khalifa University, Qatar.
Email: *(muhammad.usama, muhammad.asim, junaid.qadir)@itu.edu.pk, †siddique.latif@usq.edu.au, ‡aalfuqaha@hbku.edu.qa

***Abstract*—Intrusion detection systems (IDSs) are an essential cog of the network security suite that can defend the network from malicious intrusions and anomalous traffic. Many machine learning (ML)-based IDSs have been proposed in the literature for the detection of malicious network traffic. However, recent works have shown that ML models are vulnerable to adversarial perturbations through which an adversary can cause IDSs to malfunction by introducing a small impracticable perturbation in the network traffic. In this paper, we propose an adversarial ML attack using generative adversarial networks (GANs) that can successfully evade an ML-based IDS. We also show that GANs can be used to inoculate the IDS and make it more robust to adversarial perturbations.**

***Index Terms*—Adversarial machine learning, GAN, IDS**

I. Introduction

In cybersecurity, network anomaly detection is an important task. Rapid growth in network traffic and cyber attacks on networked applications have made intrusion detection systems (IDSs) a crucial component of the network security suite. Given network traffic samples, IDSs are expected to precisely classify which sample is unusual and malicious without suffering from a high false positive rate.

Recent advances in machine learning (ML) and deep learning (DL) techniques have revolutionized vision, language, and speech processing. The classification performance in these areas has nearly surpassed the human level performance. Motivated by the success of the ML/DL in these areas, IDSs have also adopted ML/DL for performing classification tasks to determine anomalous behavior in network traffic. However, recently linear and non-linear ML/DL classifiers are exposed to be vulnerable to adversarial examples created for fooling the classifiers in reporting malfunctioned classification reports. Since IDSs have adopted ML/DL techniques for classification there integrity has also become questionable.

Adversarial ML examples are the worst-case domain shifts of the input samples, arising from a fundamentally flawed assumption in ML/DL models that the distribution followed by the training data will also be encountered at inference time, Which is not the case in the real world. Adversaries exploit this shortcoming and use local search, combinatorial optimization or convex programming to find the adversarial perturbation which compromises the integrity of ML/DL performance.

In this work, we utilize generative adversarial networks (GANs) [5] for creating an adversarial ML attack on ML/DL-based IDSs where the details of ML/DL technique used in the IDS are unavailable to the adversary. GANs belong to a family of generative models based on differentiable generator networks. The core idea of GANs is to pit a generator network against a discriminator network in an interactive game theory like setting. The goal of the generator network is to learn the best approximation of the training data whereas the goal of the discriminator network is to distinguish between samples from original data and generated data.

We employ GANs to introduce a strategic adversarial perturbation in network data to compromise the IDS performance, and then to counter this adversarial perturbation, we introduce a GAN model in the IDS ML model to ensure robustness. Our results indicate that we can successfully evade ML/DL-based IDSs using the adversarial perturbation generated through GANs. Interestingly, GAN technology cannot only be used for attacking IDSs but also for empowering them. Our results show that by using a GAN-based defense, IDSs can be made robust against previously seen as well as unseen adversarial perturbations.

The major **contributions** of this paper are:

- We propose and validate a GAN-based adversarial ML attack on a black-box IDS. Our proposed GAN-based attack is the first adversarial attack that can successfully evade IDSs while ensuring that the functional behavior of the network traffic is preserved;
- We propose a GAN-based training mechanism for defense purposes, which improves the robustness of the IDS against adversarial perturbations; During the attack and the defense procedures, the functional behavior of the network traffic is ensured by only altering the non-functional characteristics of the network traffic.

The rest of the paper is organized as follows. In the next section, we will provide a brief review of the related research that focuses on producing adversarial examples for IDSs using GANs. We discuss some preliminaries in Section III in which we provide the problem formulation, assumed threat model, considered dataset, and the constraint of preserving func-

978-1-5386-7747-6/19/$31.00 ©2019 IEEE

tional behavior while launching adversarial attacks. Section IV introduces our GAN-based adversarial attack methodology and presents the results of our attack technique. Section V describes the details of the proposed defense mechanism against adversarial examples and highlights that the proposed defense increases the robustness of IDS against adversarial perturbations. Finally, the paper is concluded.

II. Related Work

Intrusion detection is a process of detecting malicious activities in the network traffic. Many ML/DL-based intrusion detection schemes have been proposed over the years [15], [4], [13], [3] but most of these schemes have suffered from high false positive rates and class imbalance issues. Another major issue associated with ML/DL-based IDSs is their vulnerability to the adversarial examples, where an adversary adds an imperceptible perturbation to a legitimate traffic sample to create a false sense of security. This aspect of ML/DL-based IDSs has been relatively unexplored and requires immediate attention.

Most of the recent works in developing adversarial perturbation for network security applications are focused on malware classification and portable executable (PE) classification. Grosse et al. [6] proposed an adversarial perturbation for deep neural networks (DNNs) based malware classifier where they have used fast gradient sign method (FGSM) and Jacobian based saliency map attack (JSMA) for creating adversarial malware examples. In our previous work [16], we have explored the adversarial attacks and defenses in cognitive self-organizing networks where we have performed FGSM, JSMA, and basic iterative method (BIM) attack on malware classifiers to highlight that future ML/DL-based networks will be very vulnerable to adversarial perturbations. Anderson et al. [1] proposed an adversarial perturbation for PE by using reinforcement learning technique to evade DNN based classifiers. Similarly, Kolosnjaji et al. [9] proposed a gradient-based adversarial perturbation that evades the malware PE classifier by only perturbing 1% of the total bytes. Xu et al. [17] proposed an adversarial perturbation using genetic programming to produce an adversarial portable document file (PDF) to evade the DNN classifier while ensuring the semantic properties of the PDF.

The fundamental idea of the GAN is to generate adversarial examples, which have its basis (like all generative models) in learning the true underlying distribution of the data. There are few examples where GANs have also been used for creating adversarial examples in malware, PE, and IDSs. Hu et al. [7] used GANs for making adversarial examples for DL-based black-box malware classifier but their model does not ensure the functional behavior of the malware executable. Similarly, Lin et al. [12] proposed Wasserstein GAN [2] based adversarial example generating mechanism for a ML/DL-based black-box IDS, although they claim that the functional behavior of the network traffic was preserved but they have altered two functional features of the network traffic which invalidate their claim on the preservation of functional behavior of adversarial network traffic.

In this paper, we have proposed a GAN-based adversarial example crafting technique which successfully evades the ML/DL-based black-box IDS and also ensures the preservation of functional behavior of adversarial network traffic. We have also proposed a very effective defense against adversarial ML examples using GANs to improve the robustness of the ML/DL-based IDSs.

III. Preliminaries

A. Problem Definition

Suppose $\mathcal{X}$ is the feature set with n number of features, and let (x_i, y_i) is a sample where $x_i \in X$ is a legitimate network traffic sample and $y_i \in \mathcal{Y}$ is true class label where $\mathcal{Y}$ represent the number of classes. The IDS aims to learn a classifier $f : X \longrightarrow \mathcal{Y}$ which is a true representation of the incoming network traffic. The goal of the adversary is to generate an imperceptible adversarial perturbation δ which when added to a legitimate sample x_i constitute an adversarial example x^* and gets classified as $f(x_i + \delta) \neq y_i$. We present a framework by using GANs to construct x^* that successfully evade a black-box IDS. We have also proposed a GAN-based defense to improve the robustness of ML/DL techniques used in IDSs against adversarial examples.

B. Threat Model

1) Adversarial Capabilities: In this paper, we assume black-box settings where the adversary can only query the IDS for related labels. We further assume that adversary has prepared an oracle by simultaneously querying the IDS. Rest of the information about the IDS is not accessible to the adversary.

2) Adversarial Goal: The goal of the adversary is to generate such adversarial examples which evade and compromise the integrity of the deployed IDS.

C. Dataset

We have evaluated proposed GAN-based adversarial examples on KDD99 dataset widely used for bench-marking IDS performance [8]. The dataset consists of five classes, namely: *Normal, Probe, DoS, U2R*, and *R2L* representing the different types of intrusion traffic for IDSs to evaluate. Each record in the dataset consists of 41 features, 34 of them are continuous and 7 are categorical features. One-hot representation is used to encode the categorical features. Details of the dataset's features and its relation with the attack types are provided in Table I.

D. Constraint of Preserving Functional Behavior

A very important constraint on adversarial ML examples is to preserve the functional behavior of the perturbed examples. For computer vision, the adversary has to maintain the visual appearance of the adversarial examples, for language processing examples adversary has to preserve the semantic meaning while creating adversarial text examples, for malware and PE adversary has to ensure that an dversarial perturbation does not alter the executability of the malware or PE. For network traffic features the adversary has to ensure that an adversarial perturbation does not invalidate the network traffic features.

TABLE I
DIVISION OF THE FEATURES ON THE BASIS OF THEIR MEANING IN THE NETWORK TRAFFIC

Feature Type	Features
Intrinsic	duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent
Content	hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, s_host_login, s_guest_login
Time-based traffic features	count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate
Host-based traffic features	dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate

TABLE II
RELATION OF EACH FEATURE SET WITH DIFFERENT NETWORK ATTACKS IN THE DATASET.

	Attack Types			
Feature Sets	**Probe**	**DoS**	**U2R**	**R2L**
Intrinsic	✓	✓	✓	✓
Time-based	✓	✓		
Content			✓	✓
Host-based	✓			

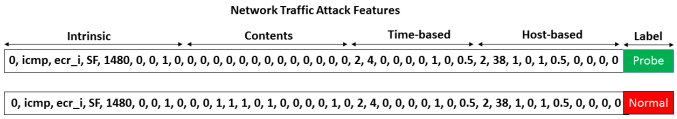

Fig. 1. An illustration of how our proposed GAN-based adversarial attack can lead to an instance of the *Probe* attack class being classified by the DNN-based black-box IDS as *Normal* while keeping the functional attributes of the *Probe* class unchanged.

To completely understand the preservation of functional behavior of network traffic features especially network attack traffic, we need to take a step back and understand the feature extraction method from *tcpdump* records. Feature extraction for IDSs from *tcpdump* records are extensively explained in [11] where they have devised a four-layer feature extraction scheme. This scheme is based on the nature of the attack in network traffic flows.

These four steps of feature extraction are as follows:

1) Firstly, *intrinsic* features from the network traffic flow are extracted. These features are necessary for any network traffic validity. Any alteration in these features will invalidate the network traffic.
2) Secondly, *time-based* features were extracted. These features provides time-based characteristics of normal and malicious traffic. These features along with intrinsic features are necessary for identifying *Probe* and *DoS* attacks. *U2R* and *R2L* attacks do not require time-based network traffic statistics as these attacks are embedded in contents of the packets. So any change in the time-based features of the network traffic flow features will invalidate the network traffic characteristics.
3) Thirdly, once the intrinsic and time-based features are extracted from network traffic then *content* features from flow traffic were extracted to detect *U2R* and *R2L* attacks. *Content* features are not required for *Probe* and *DoS* attack detection. Any alteration in these features will not invalidate the *Probe* and *DoS* attack. In this paper, we have used *content* features to generate adversarial examples.
4) Lastly, *host-based* traffic features were extracted, these features along with *intrinsic* and *time-based* features are necessary for *slow-probe* attack detection. Any alteration in these features will not only change the host-based information but also invalidates the traffic flow.

We have provided the taxonomy of the feature sets and their relation with the functional behavior of the attack traffic in Table II. Our GAN-based adversarial attack only adds perturbation to content features to ensure at the functional behavior is preserved. This also highlights that to perform an adversarial attack, the attacker must have complete domain knowledge. This is also the reason why many adversarial attacks (e.g., FGSM, BIM, JSMA) are not applicable to network traffic as these attacks do not ensure the preservation of the functional behavior of network traffic features.

An illustrative example of how the functional behavior of the network traffic features is preserved is presented in Figure 1. As the proposed attack is creating adversarial examples of *Probe* class, the adversary has to ensure that functional behavior of the intrinsic, time-based, and host-based network traffic features are preserved. Figure 1 shows that adversarial examples produced by our proposed GAN-based framework have only altered the *content* features which fool ML/DL based classifier in classifying the *Probe* attack traffic as *Normal* traffic.

IV. GAN-BASED ATTACK METHODOLOGY

In this section, we will provide the framework designed for creating adversarial examples to evade ML/DL-based IDSs.

A. Adversarial Attack Using GANs

GANs consist of two neural networks namely: a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. Provided the input examples $X = \{x_1, x_2, ..., x_n\}$, $\mathcal{G}$ tries to generate counterfeit examples, ideally from the underlying data distribution $p(x)$, that deceives the $\mathcal{D}$ in accepting them as original samples from set X. Meanwhile, $\mathcal{D}$ learns to discriminate between legitimate examples from X and counterfeited examples from $\mathcal{G}$. This learning process is formulated as a mini-max game between $\mathcal{G}$ and $\mathcal{D}$. The optimization function describing this adversarial game is given by [5]

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \; \mathbb{E}_{p(x)} \log \mathcal{D}(x) + \mathbb{E}_{p(z)} \log(1 - \mathcal{D}(\mathcal{G}(z))), \qquad (1)$$

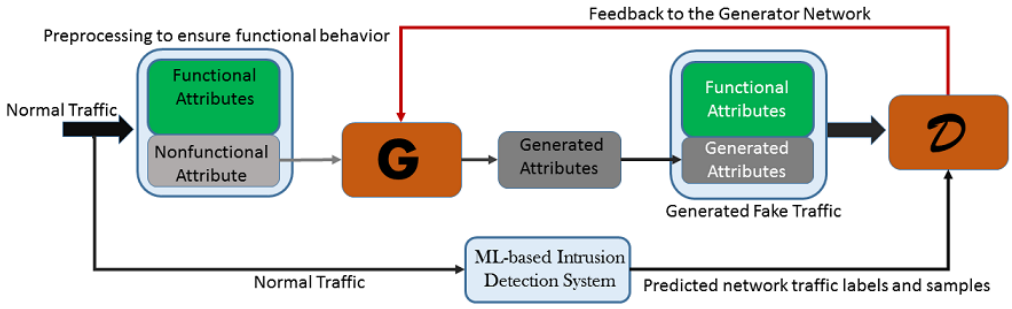

Fig. 2. GANs for evading ML/DL-based-IDSs, highlighting the procedure of generator and discriminator network training.

where $p(z)$ is the distribution of latent random variables z, and is usually defined to be a known and simple distribution such as $\mathcal{N}(0, I)$ or $\mathcal{U}(a, b)$. Training $\mathcal{G}$ and $\mathcal{D}$ is performed by taking alternative gradient steps to ensure that $\mathcal{G}$ can learn to deceive the $\mathcal{D}$ and $\mathcal{D}$ can learn to detect the counterfeit examples.

Figure 2 depicts the overall architecture opted for generating adversarial examples. The proposed GAN framework consists of three components, namely: a generator network $\mathcal{G}$, a discriminator network $\mathcal{D}$, and a black-box ML/DL-based classifier f. The input $x_i \in X$ is divided into two portions; namely, functional and non-functional attributes. This division is performed on the basis of attribute contributions to the functional behavior of the network traffic. $\mathcal{G}$ takes non-functional attributes of the data as an input and generates a perturbation δ of the size of non-functional attributes of the input. Then we concatenate the functional portion of the original traffic x and generated δ, this concatenation is given as $x||G(x)$. The concatenated samples are fed to $\mathcal{D}$, which is responsible for classifying between original and counterfeited examples.

D is trained to mimic the behavior of IDS. This is accomplished by feeding both malicious and normal traffic to both IDS and discriminator and predictions from IDS are used as labels for the training of D. Contrary to D, G is trained specifically on malicious data $\mathcal{M}$ such that D (a proxy for IDS) is fooled. The designed adversarial loss for $\mathcal{G}$ and $\mathcal{D}$ are provided in equations 2 and 3

$$\mathcal{L}_{\mathcal{G}} = \min_{\mathcal{G}} \mathbb{E}_{p(z)} \log(1 - \mathcal{D}(\mathcal{G}(z)||x)) \tag{2}$$

$$\mathcal{L}_{\mathcal{D}} = -\min_{\mathcal{D}} \mathbb{E}_{p(x)} \mathbb{f}(x) \log(\mathcal{D}(x)) \tag{3}$$

The goal of the $\mathcal{G}$ is to generate such counterfeited examples, known to be malicious, that are indistinguishable from the legitimate traffic x and the goal of the $\mathcal{D}$ is to distinguish between legitimate and counterfeited examples. While training $\mathcal{G}$ the $\mathcal{D}$ is considered fixed and reversible. The complete loss function used in the proposed GAN framework is given as:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{p(z)} \log(1 - \mathcal{D}(\mathcal{G}(z)||x)) + \mathbb{E}_{p(x)} \mathbb{f}(x) \log(\mathcal{D}(x)) \tag{4}$$

The training procedure for the GAN is provided in Algorithm 1 whereas the details of the GAN architecture used and its hyperparameters are shown in Table III.

Algorithm 1 GAN Training Algorithm

Input: G,D,$\mathcal{X}$,$\mathcal{M}$
Output: G,D
for $t = 1, 2, ...$ **do**
  for $t_G = 1, 2, ...G_{steps}$ **do**
    $x_{batch} \leftarrow \mathcal{M}$
    $\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} \mathcal{L}_G(x_{batch})$ (Adam update)
  end for
  for $t_D = 1, 2, ...D_{steps}$ **do**
    $x_{batch} \leftarrow \mathcal{X}$
    $y_{batch} \leftarrow f(x_{batch})$
    $\theta_D \leftarrow \theta_D - \eta \nabla_{\theta_D} \mathcal{L}_D(x_{batch}, y_{batch})$ (Adam update)
  end for
end for

TABLE III
GAN ARCHITECTURE AND HYPERPARAMETERS

Operation	Units	Non-linearity	Dropout
Generator (G)			
Dense	50	ReLU	0.0
Dense	20	ReLU	0.0
Dense	13	ReLU	0.0
Discriminator (D)			
Dense	50	ReLU	0.0
Dense	20	ReLU	0.2
Dense	01	Sigmoid	0.0
Hyperparameters			
Optimizer	Adam		
Learning Rate	0.001		
Batch Size	128		
Latent Dimensions	13		
Iterations	100		
Weight Initialization	Xavier initializer		

B. Results of GAN-based Adversarial Attack

The proposed GAN-based adversarial attack is used to construct the adversarial examples for the *Probe* class. These generated examples are subjected to ML/DL-based black-box IDSs which gets fooled in believing *Probe* attack traffic as *Normal* class traffic. The functional behavior of the attack traffic is ensured by following the procedure provided in Section III. We have only considered classification between *Normal* and *Probe* class network traffic but the provided framework is applicable to other network traffic classes as well.

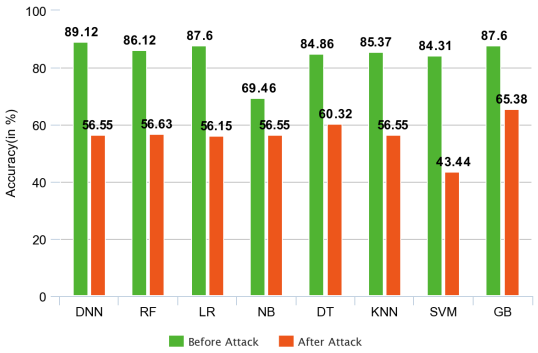

Fig. 3. Performance evaluation of our GAN-based adversarial attack framework (*Previous attack approaches are not shown as a baseline of comparison as they are not applicable in our settings as they, unlike our approach, do not ensure the preservation of networking functional behavior*)

To highlight the effectiveness of proposed GAN-based adversarial attack, we have chosen to perform a series of experiments where we employed DNN, logistic regression (LR), support vector machine (SVM), k-nearest neighbor (KNN), naïve Bayes (NB), random forest (RF), decision trees (DT), and gradient boosting (GB) techniques as black-box IDS. Since the proposed GAN framework only generates adversarial examples for one class at a time, we have used the GAN framework for producing adversarial examples for *Probe* class. We have used accuracy, precision, recall, and F1 score as evaluation parameters for the evasion attack.

Figure 3 provides a comparison between the accuracy of black-box ML/DL-based IDS before the adversarial attack and after it, highlighting that the proposed GAN-based adversarial attack compromises the integrity of the ML/DL-based IDS while ensuring the functional behavior of the network traffic. Decay in the performance of black-box ML/DL-based classifier demonstrates that adversarial examples are increasing the number of false positives and forcing the ML/DL classifier to learn wrong decision boundaries.

V. GAN-Based Defense Methodology

In this section, we discuss how an IDS can defend against proposed adversarial ML attack by opting adversarial training using generative ML models.

A. Adversarial Training by Using Generative Model in IDS

Adversarial training [14] is a method for injecting adversarial examples in training data to ensure that ML/DL model learns the possible adversarial perturbations. This new way of training ML/DL model will improve the robustness and generalization of ML/DL models by training on clean and adversarial examples [10]. To the best of our knowledge, adversarial training has not yet been explored in ML/DL-based IDSs for defending against adversarial examples. A shortcoming attached to the method of adversarial training is that it only provides robustness against the adversarial examples it was trained on and the ML/DL-based IDS will still be evaded by unknown adversarial perturbations.

To overcome this shortcoming, we have proposed a GAN-based method (depicted in Figure 4) for adversarial training of ML/DL models in black-box IDSs for defending against adversarial ML attacks.

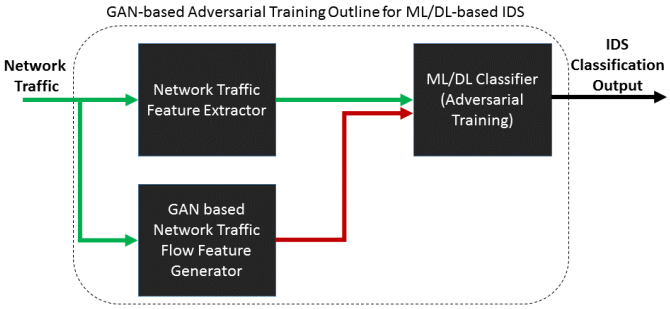

Fig. 4. Outline of GAN-based adversarial training for ML/DL-based IDSs.

Our proposed GAN-based adversarial defense works by including a generative model in the ML/DL-based IDS pipeline and the IDS model is not only trained on the input data but also on the adversarial samples generated by the generative model. Although this procedure resembles adversarial training, our approach is different since using a generative model such as GAN in an IDS will introduce the robustness against both known and unknown adversarial perturbations.

B. Results of GAN-based adversarial training

Results in Table IV highlight that the proposed procedure of defending against adversarial network traffic has improved the robustness of the ML/DL-based IDS. A clear improvement in precision, recall, and F1 score in Table IV indicates that the false positive problem associated with ML/DL-based IDS has also been taken care off by utilizing the proposed GAN-based adversarial defense against adversarial perturbations in network traffic features.

Figure 5 provides a comparison between the accuracy of different ML/DL techniques before an adversarial ML attack, after the attack, after adversarial training, and after GAN-based adversarial training. It is very evident from figure 5 that the proposed GAN-based adversarial training performed better

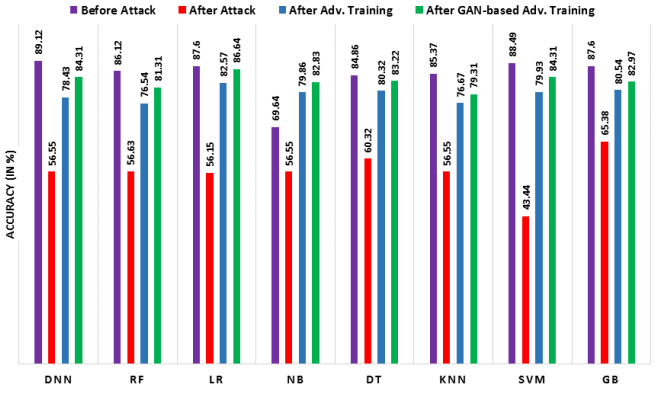

Fig. 5. Performance of GAN-based adversarial ML/DL attack and defense for black-box IDS.

TABLE IV
PERFORMANCE EVALUATION OF PROPOSED GAN-BASED ADVERSARIAL ATTACK AND DEFENSE FRAMEWORK ON DNN, RF, LR, NB, DT, KNN, SVM, AND GB SCHEMES. (EVALUATION PARAMETERS ARE ALL IN %).

ML/DL Scheme	DNN				RF				LR				NB			
Performance	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training
Accuracy	**89.12**	56.55	78.43	**84.31**	**86.12**	56.63	76.54	**81.31**	**87.60**	56.15	82.57	**86.64**	**69.64**	56.55	79.86	**82.83**
Precision	**89.28**	31.98	78.56	**82.83**	**88.68**	63.80	76.64	**85.37**	**89.48**	31.05	81.19	**86.63**	**70.56**	31.98	78.87	**82.83**
Recall	**89.12**	56.55	78.43	**84.31**	**86.12**	56.63	76.54	**81.31**	**87.60**	56.15	82.57	**86.64**	**69.46**	56.55	79.86	**82.84**
F1-Score	**89.03**	40.86	78.49	**83.56**	**85.59**	41.12	76.58	**83.29**	**87.72**	40.20	81.87	**86.63**	**67.70**	40.86	79.36	**82.83**
ML/DL Scheme	DT				KNN				SVM				GB			
Performance	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training	Before Attack	After Attack	After Adv. Training	After GAN-based Adv. Training
Accuracy	**84.86**	60.32	80.32	**83.22**	**85.37**	56.55	76.67	**79.31**	**88.49**	43.44	79.93	**84.31**	**87.60**	65.38	80.54	**82.97**
Precision	**86.88**	71.88	79.75	**86.56**	**87.84**	56.55	76.56	**80.54**	**88.93**	18.87	78.89	**87.33**	**88.91**	77.12	81.33	**86.62**
Recall	**84.86**	60.32	80.33	**83.22**	**85.37**	56.55	76.67	**79.31**	**88.49**	43.44	79.93	**84.31**	**87.60**	65.38	80.54	**82.97**
F1-Score	**84.34**	49.50	80.03	**82.37**	**84.81**	40.86	76.61	**78.64**	**88.34**	26.31	79.40	**83.60**	**87.30**	58.26	80.93	**82.06**

than the simple adversarial training procedure. The improvement in robustness using GAN-based adversarial training can be further improved by carefully selecting the hyperparameters of GAN in the IDS. We have also noticed a unique result where NB has shown a drastic improvement against adversarial perturbations, once we have performed GAN-based adversarial training. NB decouples class conditional feature densities. A clear improvement in the accuracy of the black-box IDS performance after including GAN in its training pipeline also strengthen our decision of using GAN as a defense against adversarial perturbations.

VI. CONCLUSIONS

In this paper, we have proposed a GAN-based adversarial attack on black-box ML/DL-based IDSs. The proposed adversarial attack has successfully evaded the IDS while ensuring the preservation of functional behavior of the network traffic features. We have reported the results for only one class but the proposed attack is applicable to other network traffic classes. We have also proposed and validated a GAN-based defense against adversarial perturbations to ensure robustness against adversarial ML attacks. Our results highlight that a GAN-based defense has improved the robustness of IDSs against adversarial perturbations. In the future, we will concentrate on improving our GAN-based attack and defense framework so that it can also be applied to other networking related tasks. Another important future work is to design a discrete domain GAN purely for networking applications.

REFERENCES

[1] H. S. Anderson, A. Kharkar, B. Filar, D. Evans, and P. Roth, "Learning to evade static PE machine learning malware models via reinforcement learning," *arXiv preprint arXiv:1801.08917*, 2018.

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[4] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.

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

[6] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

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

[8] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on*. IEEE, 2015, pp. 92–96.

[9] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, and F. Roli, "Adversarial malware binaries: Evading deep learning for malware detection in executables," *arXiv preprint arXiv:1803.04173*, 2018.

[10] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie *et al.*, "Adversarial attacks and defences competition," *arXiv preprint arXiv:1804.00097*, 2018.

[11] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM transactions on Information and system security (TiSSEC)*, vol. 3, no. 4, pp. 227–261, 2000.

[12] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: generative adversarial networks for attack generation against intrusion detection," *arXiv preprint arXiv:1809.02077*, 2018.

[13] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of network and computer applications*, vol. 36, no. 1, pp. 42–57, 2013.

[14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks." *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1312.html#SzegedyZSBEGF13

[15] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Systems with Applications*, vol. 36, no. 10, pp. 11 994–12 000, 2009.

[16] M. Usama, J. Qadir, and A. Al-Fuqaha, "Adversarial attacks on cognitive self-organizing networks: The challenge and the way forward," in *Proceedings of The 43rd Annual IEEE Conference on Local Computer Networks (LCN 2018)*. IEEE, 2018.

[17] W. Xu, Y. Qi, and D. Evans, "Automatically evading classifiers," in *Proceedings of the 2016 Network and Distributed Systems Symposium*, 2016.