Int. j. inf. tecnol. (December 2022) 14(7):3365–3375
https://doi.org/10.1007/s41870-022-00910-3

ORIGINAL RESEARCH

N-GAN: a novel anomaly-based network intrusion detection with generative adversarial networks

Auwal Sani Iliyasu[1] · Huifang Deng[1]

Received: 9 November 2021 / Accepted: 17 February 2022 / Published online: 5 March 2022

Abstract Network intrusion detection is one of the popular cyber defense mechanisms, which entails detection of cyber threat at network layer level. Currently, research on network intrusion detection systems (IDS) are mostly based on supervised deep learning (DL) methods, which require large amount of data to generalize well. However, collecting sufficient malicious samples for training supervised DL methods is non-trivial, especially in the modern day constantly evolving landscape of cyber threat. Unsupervised methods mitigate this issue by completely modeling the benign data, thereby establishing a normality threshold, and then flagged any data instance above that threshold as an anomaly. However, these approaches sometimes lead to too many false alarm rates (FARs). We hypothesize that, the problem is due to lack of prior knowledge on the distribution of anomaly (malicious samples), and their focus on only preserving data regularity information. Thus, adding even a few malicious samples during training can significantly improve the quality of learned representations thereby improving their robustness against FARs. Therefore, in this paper we propose N-GAN, a novel network intrusion detection technique based on generative adversarial networks (GAN). Our approach incorporates a few malicious samples during training (weakly supervised), which enable it to learn good representations instead of learning data noises or uninteresting data objects due to lack of such prior knowledge. We evaluate our N-GAN approach on a publicly available intrusion detection dataset, and achieve detection rate that surpasses other reconstruction-based anomaly intrusion detection methods on the same datasets.

Keywords Network security · Intrusion detection · Deep learning · Generative adversarial networks

1 Introduction

Cyber security issues are increasingly becoming a routine struggle, with new threat emerging every day. The proliferations of IoT as well as other communication technologies such as LTE and 5G have led to explosion in data traffic, which greatly increases the diversity and complexity of networks. As systems become increasingly more complex, they are likely to suffer from vulnerabilities, which might be exploited by malicious users. Hence, existing network security appliances such as Intrusion detection systems (IDS) need to be upgraded to keep up with evolving cyber-security challenges.

Traditionally, networks are defended against intrusion using tools such as Snort, which employ signature or rule-based techniques to thwart attacks. In these techniques, experts identify common attack vectors, and then specify rules to recognize such attack vectors in network traffics [1].

Classical machine learning (ML) methods provide an upgrade over the traditional rule-based techniques. These methods exploit various features of network traffic, which enable them to detect attack signature automatically, without explicit rule specification [2].

The massive network traffic growth coupled with complexity of modern-day attacks provides the fertile ground

✉ Auwal Sani Iliyasu
engrausan@gmail.com

Huifang Deng
hdengpp@qq.com

1 School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

for applying deep learning techniques. Deep learning (DL) as an end-to-end learning model offers advantages such as ability to learn and extract useful features automatically, thereby eliminating the need of a domain expert for feature selection [3]. Thus, recent researches on IDS are mostly inclined to this area [4].

However, the constantly changing landscape of cyber-threat makes it very difficult to collect malicious samples sufficient enough to train DL classifiers to generalize well. In addition, there is tendency of a model to become obsolete as new attacks emerge, since, the knowledge of detectable attacks is limited to those type already known and designated as targets during training. Thus, supervised DL methods are unable to detect novel or mutant of an existing attack [5].

In view of that, researchers begin to explore unsupervised anomaly-based IDS. The unsupervised anomaly-based IDS avoid the costly operation of acquiring large-scale malicious samples by leveraging the readily available large-scale normal data. The basic strategy used by the unsupervised anomaly-based IDS is to model the normal data distribution, thereby establishing a normality threshold, and then designate any data point that deviates significantly from this threshold as anomaly. Thus, on the one hand, unsupervised anomaly-based network intrusion detection methods, have the advantage of being able to detect novel or mutant of an existing attack, and thus, are able to keep pace with dynamically changing landscape of cyber-attacks. On the other hand, these techniques suffer from high false alarm rates (FARs), as they focus on only preserving data regularity information [6].

To overcome this limitation, this paper proposes N-GAN, a weakly supervised method for anomaly-based network intrusion detection using generative adversarial networks (GAN) [7]. Our approach incorporates a few anomalous instances during training. The basic idea is to combine both normal and few anomalous instances in the reconstruction learning objectives, thereby explicitly enforcing poor reconstruction for anomalous instance and vice-versa. This enables the model to learn good representations tailored specifically to the intended anomaly instead of learning uninteresting data noises.

In summary, our N-GAN model composes of an encoder and a GAN module. The encoder encodes a data point into a latent space while the generator in the GAN acts as a decoder and reconstructs it back to the features space. A high reconstruction error above certain threshold indicates an anomaly.

The intuition of our design is based on the fact that GANs have strong generative ability. Since the core of any reconstruction-based anomaly detection method is to model the distribution of a normal data, which can be complex and high dimensional, GANs serve as a suitable option for such task. Thus, our N-GAN approach generalizes both GAN and autoencoder (AE) based approaches, by leveraging the modeling power of GANs and reconstruction consistency of the AEs.

The paper is organized as follows: Sect. 2 presents the relevant works in the literature. Section 3 provides a detail explanation of our N-GAN approach. Results and discussion are presented in Sect. 4, while Sect. 5 concludes the paper.

2 Related works

Network intrusion detection based on ML techniques have been well studied in the literature. Thus, popular classical ML approaches such as K-nearest neighbor (KNN) [8], support vector machines (SVM) [9], decision tree (DT) [10] and random forest (RF) [11] have all been employed as network-based IDS. For example, Kutrannont et al. [12] proposed a KNN based IDS. KNN operates based on the assumption that a sample belongs to the class where most of its top K-neighbors reside. Therefore, the parameter K affects the performance of the model. In their work, Kutrannont et al. proposed the integration of a simplified neighborhood classification using the percentage instead of group ranking. Taking into account the unevenness of the data distribution, the improve rule selects fixed percentage (50%) of the neighboring samples as neighbors, while its efficiency is enhanced via parallel processing using graphical processing unit (GPU). The algorithm performs well on sparse data, achieving an accuracy of 99.30%.

Goeschel et al. [13] employed a combination of SVM, decision tree (DT) and Naïve Bayes classifiers. The SVM is first trained to perform binary classification to separate the data instances into benign and malicious classes. The malicious classes are then categorized into specific class of attacks using a DT classifier. However, since DT can only separate known classes of attacks, they further employ a Naïve Bayes classifier to identify the unknown attacks types. The hybrid method achieves an accuracy of 99.62% and a False alarm rate of 1.57%.

Malik et al. [14] proposed an IDS using random forest (RF) and particle swarm optimization (PSO). They train the IDS in two stages: feature selection and classification. The PSO serves as feature selection algorithm which is used to select appropriate features for classifying the attacks, while the RF is used as the classifier. They evaluate their approach using KDD cup99 dataset, and achieve detection rate of 99.92%, 99.49% and 88.46% on DoS, Probe, and U2R attack classes.

Similarly, classical unsupervised ML methods have also been served as IDS. For example, Peng et al. [15] proposed mini batch k-means with principal component analysis

(PCA) for IDS clustering. The mini batch IDS was used to improve its efficiency on big dataset. They first employed PCA to reduce the dimensionality of the data, and then applied the k-means algorithm on the reduced features. They evaluated their model using the KDD99 dataset.

However, recent researches on intrusion detection systems (IDS) are mostly based on supervised deep learning techniques. These techniques can extract features directly from raw data as well as perform classification in an end-to-end fashion. Therefore, they achieved better performance when compared against classical machine learning methods [16]. Thus, supervised deep learning model such as multi-layer perceptron (MLP), convolutional neural network (CNN), autoencoders (AE), recurrent neural network (RNN) and long short-term memory (LSTM) have all been applied in the context of network intrusion detection.

For example, Min et al. [17] proposed an IDS named TR-IDS, which leverages both statistical features as well as payload features. They employed CNN to extract important features directly from the payload. To accomplish that, they first encoded each byte in the payload in to a word vector using skip-gram word embedding, and then apply the CNN to extract the features. The extracted features are then combined with the statistical features generated from each network flow which include fields from packet header and statistical attributes of the entire flow. The features are then used to train a Random Forest classifier, which achieves an accuracy of 99.13%

Yin et al. [18] employed recurrent neural network (RNN) for intrusion detection task. The RNN model outperformed the classical ML techniques (support vector machines (SVM) and random forest) when evaluated on the NSL-KDD dataset.

Wang et al. [19] employed a combination of CNN and LSTM. Intuitively, the CNN leans low-level spatial features of network traffic, while the LSTM learns the high-level temporal features of the data. The learned features enable the model to improve the false alarm rate of the IDS.

In another work, AlQatf et al. [20] employed sparse autoencoder (AE) for dimensionality reduction. The reduced features were then retrained using SVM classifier. This enables the model to outperformed classical machine learning methods.

As can be seen, the above approaches are all supervised learning approach. Thus, they suffer from all the limitations highlighted in Sect. 1.

On the other hand, unsupervised deep learning techniques for IDS are mostly reconstruction-based, and based on generative models such as GAN and AE as well as its variants. Although, GANs have previously been employed in the general domain of anomaly detection especially in medical imaging such as in [21–23]. However, in the general domain of Network traffic analysis, GANs are mostly utilized to synthesize new data instance. For example, AS Iliyasu et al. [24] employ GAN for encrypted traffic classification. The work utilizes GAN in a semi-supervised fashion, whereby new samples generated by the GAN generator helps to train the model with a few labeled data to mitigate the difficulty associated in collecting and labeling large amount of encrypted data. GANs are also employed in [25, 26] to solve data imbalance problems. The GAN generator is used to synthesize data for the minority classes, and then another model is then employed to perform traffic analysis (anomaly detection or traffic identification) on the balanced dataset. Currently, a few works apply GAN directly to the intrusion detection problem. Li et al. [27] applied GAN to the intrusion detection problem. The intuition is that, given its strong generative ability, a GAN that has been well trained to model the distribution of normal sample should be able to reconstruct such a normal sample from a certain latent representation. Therefore, during detection, a query sample is first converted to a latent space. The latent features are then pass to GANs' generator, which acts as a decoder and reconstruct it back to the feature space. A high reconstruction error above certain threshold indicates an anomaly. However, one drawback of this approach is, it requires mapping of a given query sample to GAN's latent space, which requires costly optimization procedure. To alleviate this problem, GAN methods that simultaneously learn an encoder during training that enables fast mapping to a GANs' latent space are developed [28, 29]. Our N-GAN model also falls in this category, in the sense that we also employ an encoder to enable fast mapping of a given query to a latent space. However, unlike the above methods which are purely unsupervised, our N-GAN model is weakly supervised as it incorporates a few malicious instances during encoder training. This enables our model to explicitly enforce poor reconstruction for malicious instance, and thus, increase its robustness against false alarm rates.

3 Proposed N-GAN approach

This section explains our proposed N-GAN model. The model is based on the Wasserstein GAN, which is another version of GAN. Therefore, it is imperative to first briefly introduce the background of GAN to enable proper understanding of our approach.

3.1 GAN

Generative adversarial networks (GAN), is a recently developed new class of artificial neural network by

Godfellow et al. The model is composed of two neural networks, termed as generator and discriminator, which are trained in adversarial manner. In its basic formulation, the generator G takes a random noise z from a latent space $p(z)$, and then generates a new data instance x_{fake}.While the discriminator D receives its input x from both real data distribution $p(x)$ and the G. The discriminator then reviews each data instances, and decides whether the data is real (i.e. comes from actual training dataset) or comes from G. Thus, during training, G is optimized to fool D while D is trained to discern real data from the fake data (one generated by G).

Therefore In the context of game theory,G and D plays the following two-player minimax game:

$$\min_{\mathrm{G}} \max_{\mathrm{D}} E_{x\sim p(x)}[\log D(x)] + E_{x\sim p(z)}[\log(1 - D(G(z)))] \tag{1}$$

where $D(x) \in [0, 1]$ indicates how real the discriminator input is. On reaching the Nash equilibrium, D is unable to distinguish samples generated by G from the actual data instances. Thus, at that point, G is producing samples that closely follow $p(x)$.

However, GANs are known to be unstable, and difficult to train, which makes reaching the Nash equilibrium very difficult. Therefore, several research papers directed their efforts in improving the stability of training[30–33].Wasserstein GAN [33] improves GAN training stability by employing Wasserstein distance to measure the distance between real and fake data distributions, which assures more training stability.

$$\min_{G} \max_{D} \mathrm{E}_{x\sim p(x)}[D(x)] - \mathrm{E}_{z\sim p(z)}[D(G(z))] \tag{2}$$

As can be seen from the Eq. (2), D now serves as a critic, which gives high score to real samples and low ones to fake data.

3.2 N-GAN

As mentioned earlier, our N-GAN model comprises of a GAN and an Encoder modules. All the modules are defined as deep neural networks. We train our N-GAN model in two different phases. The first phase involves training the GAN module of the N-GAN using large-scale benign network traffic data, while the trained GAN is used to train the encoder module in the second phase using both benign network traffic data and a few malicious samples.

3.2.1 GAN training

This entails a regular GAN training as depicted in Fig. 1, and explained in Sect. 3.1, whereby generator and discriminator are trained in an adversarial manner using

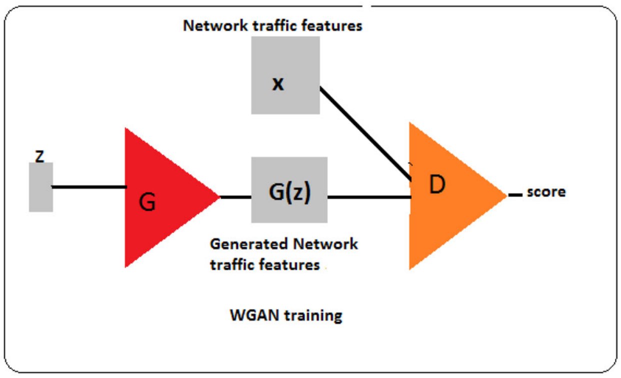


Fig. 1 GAN training setup

normal network traffic data. The generator G generates fake network traffic features from random noise z sampled from latent space $p(z)$, and then passes the generated network traffic features $G(z)$ to the Discriminator D, which tries to distinguish it from the real network traffic features. The cycle continues up to the point where G outputs samples that closely follow the distribution of real network traffic features, and therefore,D will then be unable to distinguish real network traffic features from the fake ones (those generated by the generator).Therefore, at the end of the training G learns the inherent structural variability of normal network traffic, which enables it to map latent features to normal network traffic feature space.

3.2.2 Encoder training

In this phase, the trained GAN module is used to train an encoder module E. The settings are similar to standard Autoencoder, with the GAN generator acting as a decoder. During the training, only the encoder weights are fine-tuned, where as the generators'weight remain fixed. The encoder learns to map network traffic features into latent features, however, instead of employing the residual loss as in conventional reconstruction-based approach, we employ a contrastive loss function[34]. The encoder is thus trained with a contrastive loss function, which takes into account both normal and a few malicious samples. The loss function constrains the N-GAN model such that benign samples are well reconstructed while the malicious samples are forced to have bad reconstruction (Fig. 2).

3.2.3 Contrastive loss function

The contrastive loss function uses data from two classes. In our case, the two classes X^+ and X^- are drawn from benign and malicious distributions. We also assume there exist n number of samples X^+ and m number of samples X^-, such that $n \gg m$. The objective is to discover a manifold which

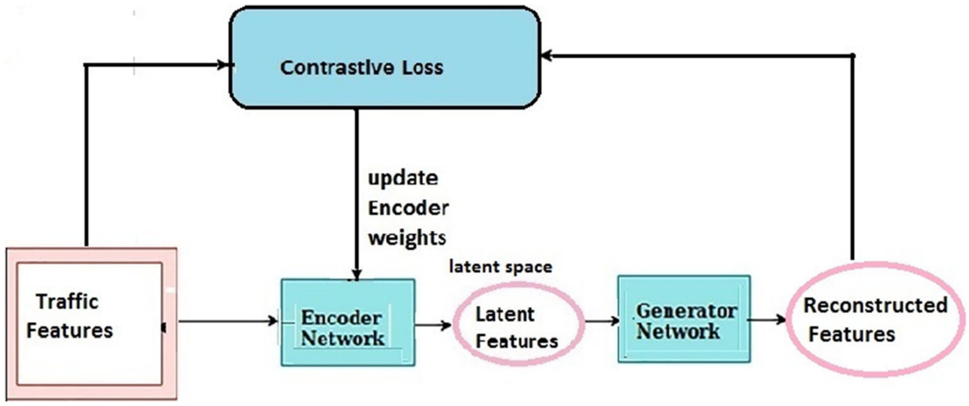


Fig. 2 Encoder training setup

is good at reconstructing the data of the benign class while ensuring that those of the malicious class are pushed away from the manifold.

Let $l(x)$ denote the label of an example x, with $l(x) \in \{-1, 1\}$ and $d(x)$ the distance of that example to the manifold with $d(x) = \|x - \bar{x}\|$. Then the loss function is described as:

$$L(X^{+} \cup X^{-}) = \sum_{x \in X^{+} \cup X^{-}} \max(0, l(x) \cdot d(x) - 1) \quad (3)$$

3.2.4 Anomaly score derivation

Once the model training is complete, the anomaly detection procedure is reconstruction-based. Network traffic features are first passed through the encoder component, which transforms it into a latent features. These latent features are then passed to the generator module, which acts as a decoder. The generator then maps the latent features back to the network traffic feature space. The intuition is, these sequences of transformations are similar to an identity transform, and thus, the degree of deviations from the transformations defines an anomaly score. Therefore, in the case of normal network traffic, these mappings yield a minimal deviation. However, a larger deviation is expected with regards to other network traffics such as attack traffics, since, the model is trained to explicitly generate large reconstruction error for network traffics that do not lie in the manifold of normal network traffic. Thus, the anomaly score formula is derived from Eq. (4):

$$A = x - G(E(x)) \quad (4)$$

Algorithm 1 The Wgan Training Process

```
Input: Normal Dataset $x^{(1)},\ldots,x^{(m)}$
Output: Discriminator $f_D$ ,Generator $f_G$ ;
$\theta_D,\theta_G \leftarrow$initialize network parameters;
1:Repeat
2:for $i$=1:$N$ do
3:Draw a batch of $N$ samples $x^{(1)},\ldots,x^{(n)}$
from the dataset
4:Draw a batch of $N$ samples $z^{(1)},\ldots,z^{(n)}$
from Noise samples $\sim p(z)$
5 : $L_D = \frac{1}{N}\sum_{i=1}^{N} f_D\left(x^{(i)}\right) - f_D\left(f_G\left(z^{i}\right)\right)$
6:    $L_G = \frac{1}{N}\sum_{i=1}^{N} f_D\left(f_G\left(z^{(i)}\right)\right)$
7:    end for
//update parameters with gradients
8 : $\theta_G \leftarrow \theta_G - \nabla_{\theta_G} L_G$
9 : $\theta_D \leftarrow \theta_D - \nabla_{\theta_D} L_D$
10: untilconvergence of parameters $\theta_D, \theta_G$
```

Algorithm 2 Encoder Training

Input: Training dataset X containing, n normal data samples and m few malicious samples i.e. $n \gg m$, $l(x)$ the label of the dataset with $l(x) \in \{-1,1\}$, trained generator f_G

Output: Encoder f_E and a Generator f_G

```
$\theta_E \leftarrow$ initialize encoder parameters
1: Repeat
1 :for $i$=to $k$ do
  2:    Draw a batch of $k$ samples $x^{(1)},....,x^{(k)}$ from the dataset $X$
3:      $z^i = f_E\left(x^i\right)$
4:      $x^{'i} = f_G\left(z^i\right)$
3:  $L_E = \frac{1}{k}\sum_{i=1}^{k} \max\left(0, l(x).\left(||x^i - x^{'i}|| - 1\right)\right)$
4:    end for
// update parameters with gradients
5:  $\theta_E \leftarrow \theta_E - \nabla_{\theta_E} L_E$
6:  until convergence of parameters $\theta_E$
```

Algorithm 3 The Anomaly detection Process

Input: Test dataset $x^1,...,x^m$, trained generator f_G ,trained encoder f_E, threshold ∂

output: Anomalous data

```
1: for $i$=1:$m$ do
2:    $A_x = x^{(i)} - f_G\left(f_E\left(x^{(i)}\right)\right)$
// determine if test sample is anomalous
3:    if $A_x > \partial$ then
4:        $x^{(i)}$ is an anomaly
5:    else
6: $x^{(i)}$ is normal
7:    end
```

4 Evaluation

4.1 Datasets

We evaluate the performance of our approach against the baseline models using the CIC-IDS2017 network intrusion detection dataset. This dataset reflects recent attacks, and to some extent satisfy the criteria for reliable intrusion detection datasets as proposed by [35], which are: anonymity, attack diversity, complete capture, complete interaction, complete network configuration, available protocol, complete traffic feature set, meta data, heterogeneity, and labeling. The CIC-IDS2017 was developed at the Canadian Institute of Cybersecurity of the University of New Brunswick (UNB) in 2017. The dataset contains about 2,830,743 rows of features, as well as raw PCAP files, which were captured at different days in a week (from Monday to Friday). The dataset has 80 features, and each row is labeled as benign or one of fourteen type of attacks as depicted in Table 1.

We use the extracted features of the datasets. We preprocess the datasets to remove unimportant features and incomplete records with respect to the required fields. For instance, we remove all rows that have the feature “Flow Packets/s” equal to “Infinity” or “NaN”. We then remove the features that have the same value for all rows, which are: Bwd PSH Flags,Bwd URG Flags,FwdAvg Bytes/Bulk, FwdAvg Packets/Bulk, FwdAvg Bulk/Rate,BwdAvg Bytes/Bulk, BwdAvg Packets/Bulk, BwdAvg Bulk/Rate and FwdAvg Bytes/Bulk.

Table 1 composition of CIC-IDS2017 dataset.

Table 1 Flow composition of the CIC_IDS2017 dataset

Attack	Flow count
Benign	2,358,036
DoS Hulk	231,073
Port Scan	158,930
DDoS	41,835
DoS Golden eye	10,293
FTP Patator	7938
SSH Patator	5897
DoS Slow Loris	5796
DoS Slow Http Test	5499
Botnet	1966
Web attack: Brute force	1507
Web attack: XSS	652
Infiltration	36
Web attack SQL injection	21
Heart bleed	11

After preprocessing, we were able to extract about 2 million records. Benign traffic constitutes about 1.5 million records, while attacks makeup the remaining number. We used 50% of the benign traffic (anomaly-free set) to train our GAN module.

4.2 Implementation

We employ Tensorflow Keras 2.4 as deep learning framework to implement our N-GAN model. We implement the GAN module (generator and discriminator), and encoder module as standard Multi-layer Perceptron (MLP) neural networks.

4.3 Evaluation metrics

Anomaly detection is considered as two-class classification problem, with anomalous samples as positives and normal samples as negatives. To assess the performance of our model, we consider two different metric categories. Those that are defined over a single threshold, and those that aggregate their decision over several possible thresholds.

For the single threshold metric, Accuracy Precision and Recall are the common metrics normally used. Accuracy, which indicates the percentage of correctly classified data items, is a poor metric for anomaly detection task, where the dataset is largely skewed in favor of normal samples. Similarly, high recall which implies Detection rate (DR) indicates fewer chances of a model missing alarming anomalies, while high precision highlights the ability of a model to classify normal data. We consider recall to be more significant in our case than precision, for example, if a model predict a normal sample as an anomaly, it is easier to rectify the judgment of the model through expert knowledge, since the anomalous samples are of small quantity. However, if model miss an anomaly, it is difficult to find anomalous data in such a huge datasets. Thus, we consider Detection Rate (DR) score as our single threshold metrics, and thus, we tweak our threshold accordingly.

Similarly, we employ receiver operator characteristic (ROC) for metrics that use several thresholds. ROC is commonly used in the literature for anomaly detection task as it gives a good picture of an algorithm performance. The area under the curve (AUC) of an ROC curve can be describe as the probability that a model assign higher score to an attack instance than to a randomly chosen non-attack instance. An AUC of 1 indicates perfect separation of classes (attack from benign) while an AUC of 0.5 means the model performs no better than a chance.

4.4 Experiments and results

To evaluate the effectiveness of our N-GAN model, we conduct three different experiments. For the baseline models, we employed conventional GAN and Auto encoder (AE).

4.4.1 *Experiment 1: general anomaly detection*

The objective of this experiment is to discover whether a few samples of attacks included during training will help our model to detect attacks that it has never seen before (novel attacks).

To perform this experiment, we first train our GAN module with benign samples (anomaly-free set) in the first phase. In the second phase, i.e. encoder training, we then select different classes of attacks in both the training and the test set. For instance, the CIC-IDS2017 can be divided in to 14 classes of attacks as depicted in Table 1. We use the following attacks: DoS Hulk, Heart bleed, Port Scan, FTP-Patator, DoS slow Loris, Brute force web attack, and Infiltration to construct our training set. We select 500 samples of each attack categories making a total of 3500 attacks samples. We then combine it with 500,000 benign samples to compose our dataset for encoder training. After training, we then evaluate our model using the following attacks classes: DoS Golden eye, DoS Slow Http Test, SSH-Patator, XSS web attack, SQL injection web attack, and Botnet which are not present in the training set.

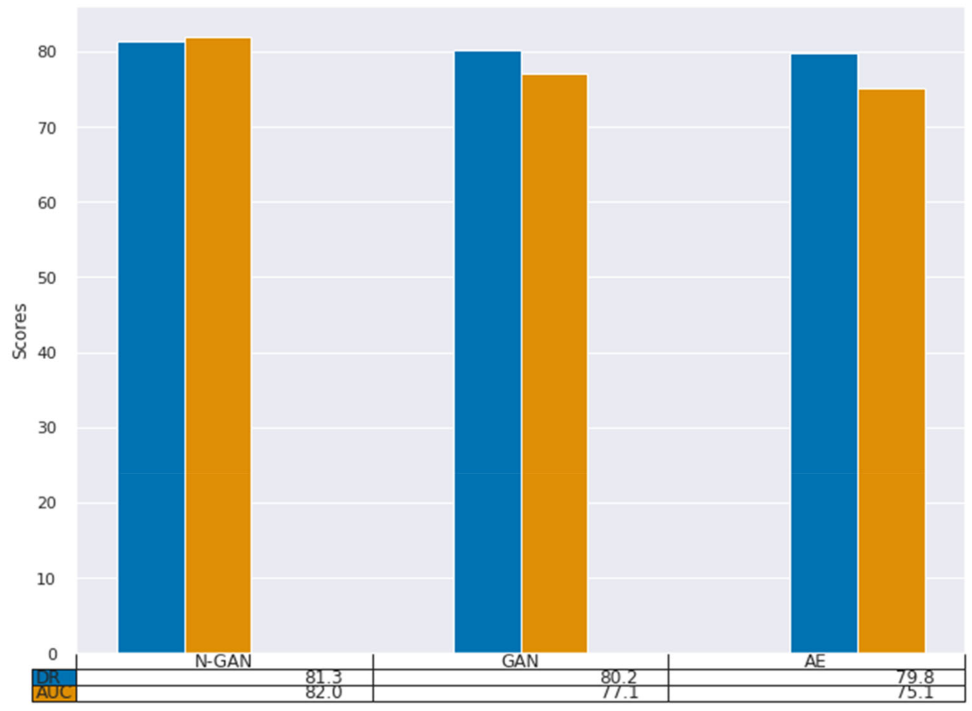
For the baseline models, we trained them with benign samples, and then evaluated them using the same test dataset as in our N-GAN model.

4.4.2 *Experiment 2: detection of specific class of an attack*

In this experiment we try to find out whether adding a few malicious samples of a known attack has effect in aiding detection of that attack. Thus, we train our model with a few samples (500 samples) of the attack in question. We conduct the experiment by selecting attacks which have up to 500 samples in the datasets.

4.4.3 *Experiment 3: detecting mutant of an existing attacks (attacks variants)*

In this Experiment, the objective is to analyze how good our N-GAN model is in detecting attack variants (Mutant of an existing attack). Therefore, we train our model with few samples of certain attacks classes, and then evaluate the model against subclasses of those attacks. To perform the experiment, we extracted categories of attacks which we believe to comprise variants of one another. For instance, the DoS compose of the following attack variants: DoS Golden eye, DoS Hulk, and DoS slowhttptest and DoS slowloris. Thus, we train and evaluate our model using different subclasses of the attacks. We train our model with few samples of DoS Hulk, and DoS Golden eye, and then evaluate with DoS slow Http Test and DoS Slowris.



	N-GAN	GAN	AE
🟦 DR	81.3	80.2	79.8
🟧 AUC	82.0	77.1	75.1

Fig. 3 Result of experiment 1: general anomaly detection

Table 2 Result of experiment 2: detection of specific class of an attack

Model Attack	N-GAN		GAN		AE		N-GAN—max (GAN, AE) Max (GAN,AE) (Improvement (%))	
	DR	AUC	DR	AUC	DR	AUC	DR	AUC
DoS Hulk	0.991	0.970	0.913	0.867	0.791	0.727	8.543	10.300
DoSslow httptest	0.921	0.918	0.795	0.717	0.825	0.865	11.636	5.300
DoS Slowloris	0.907	0.915	0.835	0.731	0.693	0.648	8.622	17.600
FTP-Patator	0.453	0.568	0.421	0.432	0.666	0.417	20.433	13.600
SSH-Patator	0.411	0.433	0.351	0.250	0.359	0.423	14.485	1.000
DDoS	0.965	0.938	0.850	0.794	0.735	0.839	13.529	9.900
Botnet	0.870	0.830	0.771	0.723	0.691	0.698	12.840	10.700
DoS Golden eye	0.984	0.967	0.876	0.794	0.834	0.877	12.323	9.000
Port scan	0.901	0.875	0.787	0.728	0.693	0.681	14.485	14.700

Figure 3 present the result of our experiment1.As can be observed from the results, our N-GAN model improves the DR and AUC by 1.4% and 6.4% respectively. The DR is slightly better than that of the baseline models. This is owing to the fact that; our model always tries to discover a singular representation for identification based on the few malicious samples observed during training. However, it is difficult to learn such singular representation due to the diverse nature of the attacks. For instance, DoS attempts to shut down a system to stop traffic flow altogether, whereas attacks such as SSH-Patator attempts to infiltrate the system undetected. Since attacks differ in purpose and implementations, thus, our N-GAN model performance is similar to the baseline models whose modus operandi is only to characterize the benign traffic.

However, in experiment 2, there is significant difference in performance between our model and the baseline models as shown in the result (Table 2). We achieve up to 20.433% and 17.6% improvement in DR and AUC respectively. This is because our N-GAN model leverages the few malicious samples present during training to learn representations specific to the attack in question, hence the reason for significant gap in performance between our model and the baselines.

Similarly, Fig. 4 present the result of experiment 3. As can be seen from the results, the trend is similar as in experiment 2. This is because, attacks mutants share

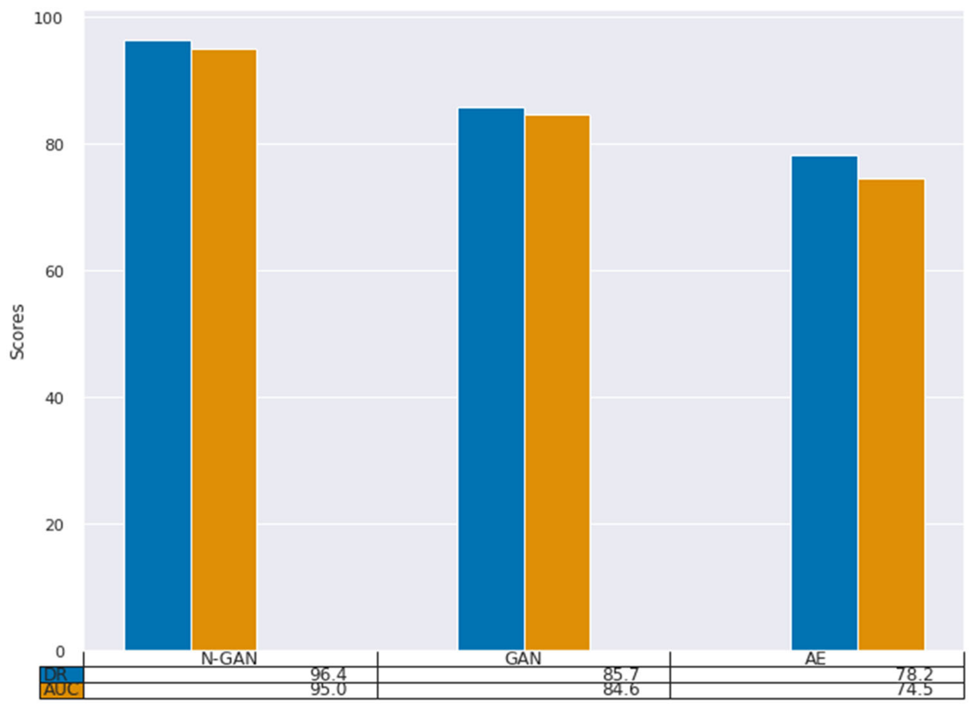


Fig. 4 Result of experiment 3: DoS attacks variants detection

similar attributes, thus, the presence of few samples of subclasses of attacks during training helps the model to discover the manifold of broader class of a such attacks. Thus, our model significantly outperforms the baseline models.

5 Conclusions

Unsupervised anomaly-based intrusion detection techniques offer promising alternative to the difficulty in acquiring large-scale malicious samples for training supervised learning models. However, these techniques suffers from high false alarm rate, as they are only trained on benign traffic which makes them learn suboptimal representations for detecting anomaly, since they lack prior knowledge of distribution of anomalies, and focus on only preserving data regularity information.

This paper proposes a novel anomaly-based intrusion detection techniques using GAN to address the limitations. Our method incorporates a few malicious samples during training.

To demonstrate the efficacy of our approach, we evaluated our model using a publicly available intrusion detection dataset. Our proposed method outperforms the baseline models on three different scenarios. In the first scenario, i.e. general anomaly detection, our model performs slightly better than the baseline models. This is due to the diverse nature of the attacks, and it is difficult to learn singular representations for identification based on the few malicious samples observe during training. However, our model performance improved significantly in the second and third scenario, when detecting specific class of an attack and mutant of an existing attack. The fact that, our model leverages few samples of attack during training helps the model to discover the manifold of good representations for detecting such attack. Therefore, it is safe to say that, our model can be used in situation like zero-day attacks. Since, even in such scenario, few samples of attacks can be obtained which will be sufficient enough to train our model to detect similar occurrence of the same attack or its variants in the future.

Declarations

Conflict of interest We hereby declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Scarfone KA, Mell PM (2007) Guide to intrusion detection and prevention systems (IDPS). National Institute of Standards and Technology, Gaithersburg. https://doi.org/10.6028/NIST.SP.800-94
2. Tsai C-F, Hsu Y-F, Lin C-Y, Lin W-Y (2009) Intrusion detection by machine learning: a review. Expert Syst Appl 36:11994–12000. https://doi.org/10.1016/j.eswa.2009.05.029
3. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
4. Gamage S, Samarabandu J (2020) Deep learning methods in network intrusion detection: a survey and an objective comparison. J Netw Comput Appl 169:102767. https://doi.org/10.1016/j.jnca.2020.102767
5. Xu C, Shen J, Du X (2020) A method of few-shot network intrusion detection based on meta-learning framework. IEEE Trans Inf Forensics Secur 15:3540–3552. https://doi.org/10.1109/TIFS.2020.2991876
6. Ziai A (2021) Active learning for network intrusion detection. In Data Science, pp 3–14. Springer, Singapore
7. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Advances in neural information processing systems, 27
8. Liao Y, Vemuri VR (2002) Use of K-nearest neighbor classifier for intrusion detection. Comput Secur 21:439–448. https://doi.org/10.1016/S0167-4048(02)00514-X
9. Li W, Liu Z (2011) A method of SVM with normalization in intrusion detection. Procedia Environ Sci 11:256–262. https://doi.org/10.1016/j.proenv.2011.12.040
10. Kumar M, Hanumanthappa M, Kumar TVS (2012) Intrusion detection system using decision tree algorithm. In: 2012 IEEE 14th international conference on communication technology. p 629–634. IEEE, Chengdu, China. https://doi.org/10.1109/ICCT.2012.6511281
11. Farnaaz N, Jabbar MA (2016) Random forest modeling for network intrusion detection system. Procedia Comput Sci 89:213–217. https://doi.org/10.1016/j.procs.2016.06.047
12. Kuttranont P, Boonprakob K, Phaudphut C, Permpol S, Aimtongkhamand P, KoKaew U, Waikham B, So-In C (2017) Parallel KNN and neighborhood classification implementations on GPU for network intrusion detection. J Telecommun Electron Comput Eng 9:5
13. Goeschel K (2016) Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. In Southeast Con 2016, p 1–6. IEEE, Norfolk, VA, USA. https://doi.org/10.1109/SECON.2016.7506774
14. Malik AJ, Shahzad W, Khan FA (2015) Network intrusion detection using hybrid binary PSO and random forests algorithm: network intrusion detection using hybrid binary PSO. Secur Commun Netw 8:2646–2660. https://doi.org/10.1002/sec.508
15. Peng K, Leung VCM, Huang Q (2018) Clustering approach based on mini batch kmeans for intrusion detection system over big data. IEEE Access 6:11897–11906. https://doi.org/10.1109/ACCESS.2018.2810267
16. Liu H, Lang B (2019) Machine learning and deep learning methods for intrusion detection systems: a survey. Appl Sci 9:4396. https://doi.org/10.3390/app9204396
17. Min E, Long J, Liu Q, Cui J, Chen W (2018) TR-IDS: anomaly-based intrusion detection through text-convolutional neural network and random forest. Secur Commun Netw 2018:1–9. https://doi.org/10.1155/2018/4943509
18. Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418
19. Wang W, Sheng Y, Wang J, Zeng X, Ye X, Huang Y, Zhu M (2018) HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access 6:1792–1806

20. Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabahi K (2018) Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access 6:52843–52856. https://doi.org/10.1109/ACCESS.2018.2869577
21. Schlegl T, Seeböck P, Waldstein SM, Langs G, Schmidt-Erfurth U (2019) f-AnoGAN: fast unsupervised anomaly detection with generative adversarial networks. Med Image Anal 54:30–44. https://doi.org/10.1016/j.media.2019.01.010
22. Schlegl T, Seeböck P, Waldstein SM, Schmidt-Erfurth U, Langs G (2017) Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. ArXiv170305921 Cs
23. Akcay S, Atapour-Abarghouei A, Breckon TP (2018) GANomaly: semi-supervised anomaly detection via adversarial training. ArXiv180506725 Cs
24. Iliyasu AS, Deng H (2020) Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks. IEEE Access 8:118–126. https://doi.org/10.1109/ACCESS.2019.2962106
25. Lee J, Park K (2019) AE-CGAN model based high performance network intrusion detection system. Appl Sci 9:4221. https://doi.org/10.3390/app9204221
26. Jiang W, Hong Y, Zhou B, He X, Cheng C (2019) A GAN-based anomaly detection approach for imbalanced industrial time series. IEEE Access 7:143608–143619. https://doi.org/10.1109/ACCESS.2019.2944689
27. Li D, Chen D, Shi L, Jin B, Goh J, Ng S-K (2019) MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks. ArXiv190104997 Cs Stat
28. Chen H, Jiang L (2019) Efficient GAN-based method for cyber-intrusion detection. ArXiv190402426 Cs Stat
29. Zenati H, Foo CS, Lecouat B, Manek G, Chandrasekhar VR (2019) Efficient GAN-based anomaly detection. ArXiv180206222 Cs Stat
30. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. ArXiv151106434 Cs
31. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training GANs. ArXiv160603498 Cs
32. Mirza M, Osindero S (2014) Conditional generative adversarial nets. ArXiv14111784 Cs Stat
33. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein GAN. ArXiv170107875 Cs Stat
34. Jurie SR, Frédéric (2014) Discriminative autoencoders for small targets detection. In: 22nd international conference pattern recognition, IEEE
35. Lashkari AH, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp, p 108–116
36. https://www.unb.ca/cic/datasets. Accessed 24 July 2021