Denoising Adversarial Autoencoder for Obfuscated Traffic Detection and Recovery

Ola Salman, Imad H. Elhajj, Ayman Kayssi, Ali Chehab

Denoising Adversarial Autoencoder for Obfuscated Traffic Detection and Recovery

Ola Salman[1], Imad H. Elhajj[1], Ayman Kayssi[1], and Ali Chehab[1]

Department of Electrical and Computer Engineering
American University of Beirut, Beirut 1107 2020, Lebanon
oms15, ie05, chehab, ayman@aub.edu.lb

Abstract. Traffic classification is key for managing both QoS and security in the Internet of Things (IoT). However, new traffic obfuscation techniques have been developed to thwart classification. Traffic mutation is one such obfuscation technique, that consists of modifying the flow's statistical characteristics to mislead the traffic classifier. In fact, this same technique can also be used to hide normal traffic characteristics for the sake of privacy. However, the concern is its use by attackers to bypass intrusion detection systems by modifying the attack traffic characteristics. In this paper, we propose an unsupervised Deep Learning (DL)-based model to detect mutated traffic. This model is based on generative DL architectures, namely Autoencoders (AE) and Generative Adversarial Network (GAN). This model consists of a denoising AE to de-anonymize the mutated traffic and a discriminator to detect it. The implementation results show that the traffic can be denoised when different mutation techniques are applied with a reconstruction error less than 10^{-1} . In addition, the detection rate of fake traffic reaches 83.7%.

Keywords: Machine Learning, Network Security, Traffic Classification, Obfuscation, Deep Learning, IoT, Autoencoder, Generative Adversarial Network.

1 Introduction

In the IoT era, billions of things are being connected to the Internet. This results in an unprecedented growth in the amount of generated traffic. This traffic presents different QoS and security challenges. Traffic classification emerges as a key enabling tool to meeting some of these challenges. For example, IoT devices' vulnerabilities have been exploited to perform critical network attacks (e.g. Mirai) [34]. Tracking these devices and detecting abnormal traffic are key to prevent harmful network attacks.

In this context, Machine Learning (ML) based methods were proposed for traffic classification and intrusion detection. Many supervised and unsupervised methods have been employed accordingly. While supervised methods are devoted to classifying the (attack) traffic based on known class labels, the unsupervised ones allow the detection of unknown traffic. The traditional ML methods rely on

well-structured and hand-designed features. These features are extracted using statistical traffic measures (e.g. maximum, minimum, standard deviation, etc. of the packet sizes, and packets interarrival times).

However, mutation techniques alter the traffic statistical features, making it very challenging to know the original traffic type. Moreover, the mutation techniques might change the packets characteristics while maintaining the flow statistical features unchanged. In this case, the detection of abnormal traffic can be evaded.

Recently, Deep Learning (DL) has acquired a lot of attention due to its representation learning capabilities. Using a new data representation in this paper, we propose an unsupervised DL model to detect abnormal traffic and de-anonymize the mutated one. Generative DL architectures, namely Autoencoders (AE) and Generative Adversarial Networks (GAN), have been applied mainly in the computer vision domain to detect abnormality and to denoise images. AE is a DL architecture for extracting data representation, and GAN has the capability to generate fake data samples to enhance the discrimination between real and fake data. In this paper, we combine AE and GAN to detect abnormal traffic and de-anonymize the mutated one. The proposed architecture consists of an encoder, a decoder, and a discriminator. The encoder-decoder pair form a denoising AE responsible for learning the original data representation and to denoise the mutated one. In parallel, the discriminator is trained to differentiate between the mutated traffic (abnormal) and the denoised traffic (normal). The training of the proposed model relies on data collected from real IoT devices and IoT attacks. The testing results show the robustness of the proposed method to detect mutated traffic and to recover the original one. Note that the proposed model is not limited to IoT traffic and can be applied to any type of network traffic.

The rest of the paper is organized as follows: in section 2, we present an overview of related concepts. In section 3, we present our proposed model. Section 4 details the implementation and presents the evaluation results. In section 5, we discuss the results and present our future work. Finally, we conclude in section 6.

2 Background and Related Work

In this section, we present the related work including a review of the generative DL architectures, their application in detecting abnormality, traffic classification and intrusion detection methods, along with the obfuscation techniques that can affect their accuracy.

2.1 Unsupervised Deep Learning

In this subsection, we explain the DL architectures that we based our work on. These architectures can learn the input data representation and are therefore called generative DL models.

Generative Adversarial Network: Introduced by Goodfellow et al. in [24], GAN consists of two parts: the generator (G) and the discriminator (D). From a game theoretic perspective, GAN can be interpreted as a zero-sum or min-max game between the generator and the discriminator. The generator tries to learn the input data representation to generate data samples very similar to the real ones. The discriminator tries to maximize the probability of distinguishing between fake and real input. The GAN objective function can be presented as follows: $V(G,D) = E_{x\sim p_{data}(x)}(log(D(x)) + E_{z\sim p_z(z)}(log(1 - D(G(z)))))$ where x is the input data, $p_{data}(x)$ is the data distribution, $D(x)$ is the discriminator output, $p_z(z)$ is the fake data distribution, z is a sample from $p_z(z)$, and $G(z)$ is the generator output. The generator aims at minimizing the probability of fake data detection by the discriminator, which means that the G objective is to find $\min_G(E_{z\sim p(z)}(log(1 - D(G(z)))))$. The discriminator aims at maximizing the probability of detecting real data as real and fake data as fake, which means that the D objective is to find $\max_D(E_{x\sim p_{data}(x)}(log(D(x))) + E_{z\sim p(z)}(log(1 - D(G(z)))))$. Thus, the GAN objective is to find $\min_G \max_D(V(G,D))$. Primarily, GAN is applied for synthetic data generation. GAN has been applied also for image anomaly detection [50, 27, 19]. In fact, the adversarial learning permits the discriminator to detect abnormal input data. Furthermore, the generative learning permits the generator to learn the real data representation, which makes GAN suitable for image denoising [40, 47].

Autoencoders: Being a generative model, AE is a type of DL networks that is specialized in extracting the input data representation. The AE consists of two parts: an encoder and a decoder. The encoder extracts a compressed data code by estimating a function f, in such a way $z = f(x)$, where x is the input data, and z is the extracted representation or latent variable. The decoder aims at reconstructing the input data by relying on the extracted representation. In other terms, the decoder tries to infer the inverse function g, in such a way that $y = g(f(x)) = g(z)$. The AE objective function can be represented at the minimization of the difference between $g(f(x))$ and x. In other terms, the AE aims at minimizing the reconstruction error.
A type of AEs is the probabilistic autoencoder, which aims to infer the distribution of x, $p_\theta(x)$ by means of another distribution $q_\phi(z/x)$ (probability of the latent variable z knowing the input x). A well-known type of probabilistic AEs is the Variational Autoencoder (VAE), which imposes a prior restriction on $p(z)$ to be a normal distribution. In this case, the problem reduces to maximizing the Evidence Lower Bound (ELBO) or maximizing the KullbackLeibler (KL) divergence between $q_\phi(z)$ and $p_{\theta(z/x)}$, represented by $KL(q_\phi(z/x)||p(z))$. Having the ability to extract the real data distribution, VAEs have been applied for anomaly detection. Indeed, when the reconstruction error is large, anomaly is detected in the input data. Borrowing the adversarial concept from GAN, Adversarial AEs (AAE) were introduced by Makhzani et al. in [30]. Similar to the GAN, AAE includes a discriminator that tries to differentiate between the data sampled from the latent variable prior $p(z)$ and the real data. In this case, the discriminator

aims to minimize $L_{dis} = -1/N \sum_{i=0}^{N-1} log(d_x(z_i)) + \sum_{j=N}^{2N} log(1 - d_x(z_j))$, and the generator tries to minimize $L_{prior} = 1/N \sum_{i=0}^{N-1} (log(1 - d_x(z_i))$, where N is the number of samples, and $d_x(z_i)$ is the discriminator output of the latent space variable. In this case, if the total loss function is optimized, $q_\phi(z/x)$ will be very similar to $p(z)$, or in other terms, $KL(q_\phi(z/x)||p(z))$ will be minimized, and thus the log likelihood of the original data distribution will be maximized. AAEs were applied also for anomaly detection in images [23, 46, 44, 5]. Furthermore, a recent work has considered to add the denoising function to the AAEs for image denoising [17]. In this case, the corrupted data $\tilde{x}$ is considered as input and two methods were proposed for model representation. The first consists of matching $\tilde{q}_\phi(z/x)$ to $p(z)$, and the second consists of matching $q_\phi(z/\tilde{x})$ to $p(z)$. However, in our work, we use a sparse AE that aims to minimize the Mean Square Error (MSE) between the reconstructed data and original data. In addition, applying the adversarial concept, we choose to train a discriminator to detect abnormal traffic when the reconstruction error is high. Thus, unlike previous work, the generator part of GAN is omitted [13].

2.2 ML based Traffic Classification and Intrusion Detection

Traffic classification is an essential network function, which is necessary for traffic engineering, QoS management, and security management. Different methods have been proposed for traffic classification using different sets of features [18, 32, 22, 9, 33, 8]. More recently, DL has been applied for traffic classification using new features and new data representations [36, 45, 14, 39, 21].
On the other hand, intrusion detection is an essential network security component. Several approaches have been adopted to detect and prevent network attacks. Traditional Intrusion Detection Systems (IDSes) are rule-based, where the attack signature is known by identifying some patterns in the packets fields. However, this method fails to detect unknown attacks (e.g. zero-day attacks) and requires the inspection of the packet header. In addition, traffic anonymization can be used to thwart detection by traditional IDS. ML methods have been proposed to detect network attacks and traffic abnormality by means of statistical features [6]. However, the traditional methods require the extraction of the features by computing statistical measures that might be data dependent. In addition, the detection of abnormal traffic is not a straightforward task, given that the abnormal traffic might present similar statistical behavior to the normal one [4].
Recently, DL has been applied in the network domain for intrusion detection [3, 43]. More specifically, IoT, which presents aggravated security challenges, has acquired a special attention from the intrusion detection perspective [10, 16]. Convolutional Neural Network (CNN) and Recurrent neural Network (RNN) have been applied for payload-based attack detection in [29]. While supervised learning was mainly considered for identifying specific attacks [48], the detection of unknown and zero-day attacks call for unsupervised-learning-based methods. In fact, the proposed unsupervised [31] or semi-supervised [41] DL methods for

intrusion detection use statistical features. To the best of our knowledge, no previous work considered the recovering (denoising) of the mutated traffic. In addition, this is the first work to consider the question of detection of mutated traffic, in addition to the (unknown) attack traffic.

2.3 Traffic Obfuscation Techniques

In the aim of protecting user privacy, a new research direction is considering traffic obfuscation to thwart classification. In this context, many obfuscation techniques have been proposed [15, 25]. These methods can be classified in seven categories: steganography, tunneling, anonymization, mutation, morphing, and physical layer obfuscation. While steganography and physical layer obfuscation techniques require specific protocols to recover the original data, the remaining techniques can be applied without imposing changes to the current network protocols. Anonymization and tunneling hide some packet-related information by encryption and establishment of virtual connections (port numbers and internet addresses). However, statistical ML methods still have power to classify the anonymized or tunneled traffic. Mutation and morphing are two techniques that consider modifying the statistical traffic characteristics considering the modification of the packet size and the packet Inter-Arrival Times (IAT) [11]. This has great impact on the accuracy of ML-based classification and intrusion detection. Even though the obfuscation techniques were intended to protect the user privacy, attackers might use them to perform their attacks without being detected. In this context, padding and traffic shaping are proposed to modify the packet size and IAT respectively [7, 35].
Recently, the GAN DL architecture has been employed to adapt malware traffic to the normal one [42, 51, 28, 37]. On the contrary, in this paper, we will consider a discriminative denoising DL model to detect abnormality by relying on the trained discriminator and to de-anonymize mutated traffic by means of a denoising AE.
Our contributions in this paper can thus be summarized by the following points:

- The discriminative part of GAN with a denoising AE are combined to allow ***mutated traffic detection*** and recovery.
- A ***new data representation*** is used to permit the de-anonymization of mutated traffic by applying DL-based techniques.

3 Proposed Abnormal Traffic Detection

In this section, we detail our proposed DL model, the attack model, and the traffic representation method.

3.1 Attack Model

Our attack model consists of an attacker trying to modify the packet size (padding) and IAT (shaping), in such a way to hide any information that serves for attack detection or traffic classification. The mutations are therefore of two types,

padding and shaping [28, 29], as shown in Fig. 1. In the following, we summarized the packet padding techniques listed in [7, 35, 12], with s being the packet original size and m(s) being the packet size after mutation.

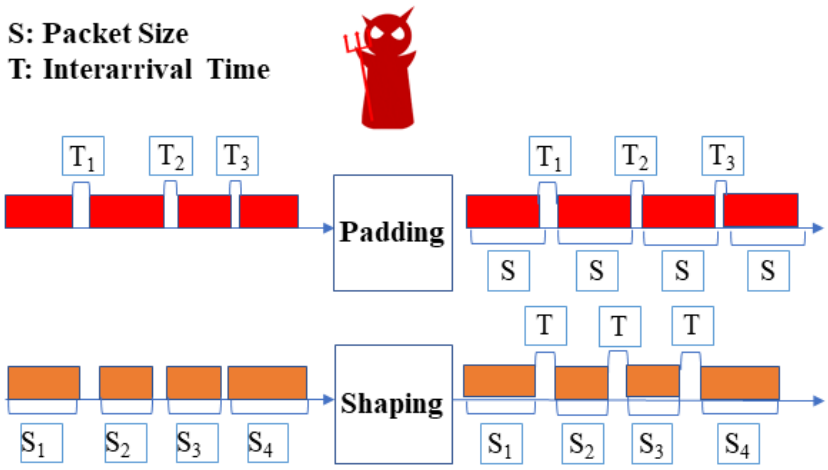

Fig. 1. Attack model

1. **Padding to the Maximum Transmission Unit (MTU):** this technique consists of padding all the flow packets to the same size, which is the MTU. In this case, the mutation can be expressed by $m(s) = MTU$.
2. **Linear padding:** this technique consists of linearly padding the flow packets sizes. In this case, the mutation equation can be expressed as follows: $m(s) = \lceil s/c \rceil * c$, where c is a parameter to choose, and $\lceil s/c \rceil$ is the ceiling of s/c.
3. **Exponential padding:** this technique consists of padding the packet size in an exponential manner. The mutation can be expressed by the following equation: $m(s) = min(2^l og_2(s), MTU)$.
4. **Elephants and mice padding:** this technique consists of padding the packet to a certain size c (mice), if the original packet size if less than c. If not, the packet size is padded to the MTU (elephant).
5. **Random padding:** this technique consists of padding the packet randomly to a size chosen randomly from the interval $([s, MTU])$. In this case, the mutation function can be expressed as following: $m(s) = RAND([s, MTU])$.
6. **Probabilistic padding:** this technique assumes that the packet sizes follow a normal distribution. In this case, the mutated size is chosen randomly based on a normal distribution, where the mean (μ) and standard deviation (σ) are computed considering the original traffic packet sizes. In this case, $m(s) = GAUSS(\mu, \sigma)$ [12].

   For the IAT shaping techniques, we list in the following the ones included

in [49] in addition to a new technique proposed in [12]:

7. **Constant IAT:** this technique consists of sending the packets at a fixed IAT.
8. **Variable IAT:** this technique consists of sending the packets at a variable interval of time randomly chosen from the interval $[I_1, I_2]$.
9. **Probabilistic IAT shaping:** this technique assumes that the IAT follows a normal distribution. Thus, the packets are transmitted at an interval chosen randomly based on a normal distribution, where the mean and standard deviation are computed based on the original traffic packets IAT.

The last method, described in [20], combines shaping and padding.

10. **Fixed size and fixed IAT:** this method consists of padding the size of all the flow packets to the MTU and sending all the packets at a fixed time interval.

This model considers two types of adversaries, including:

- **Malicious adversary:** this type of attackers aims at mutating the attack traffic to evade intrusion detection.
- **Benign adversary:** this type of attackers aims at hiding the traffic characteristics to protect the user privacy.

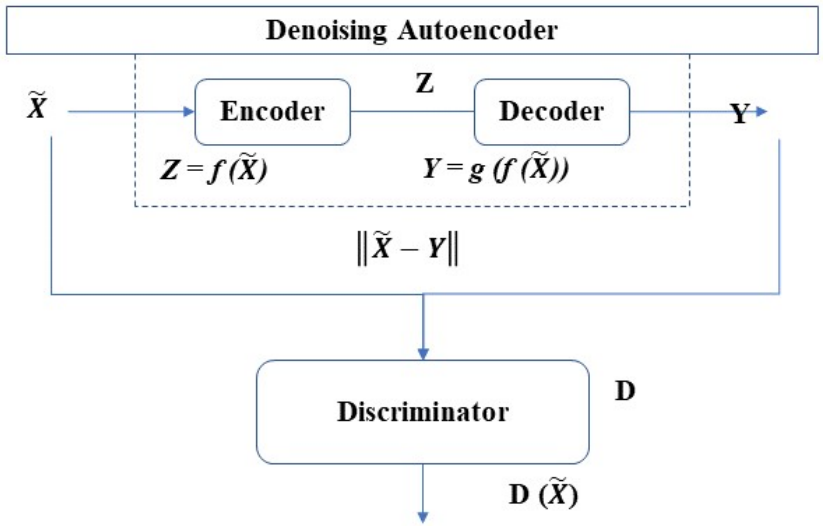

Fig. 2. Proposed model

3.2 Deep Learning Model

Our proposed model, illustrated in Fig. 2, consists of two parts: a denoising AE and a discriminator. The denoising AE consists of an encoder and a decoder. The encoder aims at estimating the function f, that maps the input space of X to the latent space of the latent variable Z, where Z is a compressed representation of X. The denoising AE is fed with a dataset containing the mutated version $\tilde{x}$ of the initial data X. The AE aims at minimizing the reconstruction error $L(X, \tilde{X})$. In our case, the loss function is the MSE function, $L(X, \tilde{X}) = (X - g(f(\tilde{X})))^2$, where X is the original data and $g(f(\tilde{X}))$ is the reconstructed data by considering the mutated data as input. The discriminator aims at differentiating between normal and abnormal traffic. In fact, the discriminator will be trained on classifying the mutated data (i.e. the mutated input data) as abnormal and the reconstructed data (i.e. the decoder output) as normal. The output function of the discriminator is a sigmoid function $p(y = y^{(j)}/x) = 1/(1 + e^{-y^{(j)}})$, where $y^{(j)}$ is the output class that can take the two values: 0 for $j = 0$ and 1 for $j = 1$, and the loss function is a cross entropy function $L_D = 1/m \sum_{j=0}^{1} y^{(j)} log(y^{(j)}) + (1 - y^{(j)}) log(1 - y^{(j)}))$.

3.3 Proposed Model Workflow

After training the model, the proposed scheme workflow, illustrated in Fig. 3, consists of: 1) passing the traffic to the discriminator; 2) if the traffic is normal, it is passed to the classifier of normal traffic; 3) If not, it is passed to the denoiser. 4) After denoising, it is passed again to the discriminator. 5) If a normal traffic is detected, it is passed to the classifier; 6) if not, the traffic is detected as abnormal, so it might be obfuscated or attack traffic. The same workflow is repeated for the abnormal traffic to know the attack type or detect an unknown attack traffic.
In fact, two cases are considered. First, when training the model on normal traffic, the aim is to detect attack traffic and the mutated one as abnormal. In this case, the denoising aims at recovering the normal traffic for classification. However, when training the model with attack traffic, at the testing phase the aim is to detect unknown attacks and to denoise the mutated attack traffic to know the exact attack type. Note that the classifier will be trained in a supervised mode to classify the traffic based on the IoT device type in the normal traffic case and based on the attack type in the attack traffic case.

3.4 Data Representation

In our case, the data consists of collected network traffic. This traffic is filtered by flows, where a flow is defined as being the set of packets having the same: source IP address, source port number, destination IP address, destination port number, and protocol (TCP or UDP). For each packet, we extract three features: size (s), IAT (t), and direction (d). For each flow, we consider the first $4 * 4$ packets in either direction. In fact, as shown in Fig. 3, the extracted features can be visualized in $4 * 4$ RGB images, where the i^{th} pixel RGB coordinates are

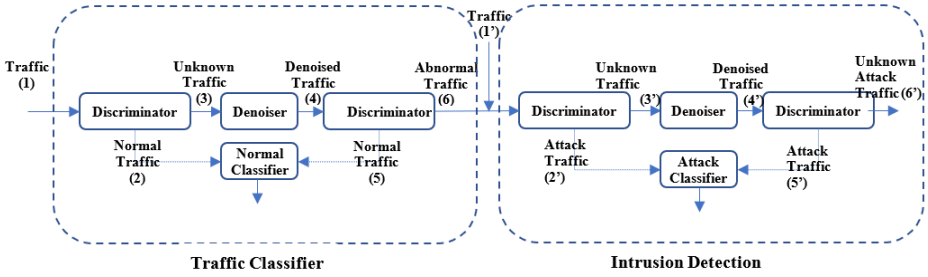

Fig. 3. Proposed scheme workflow

represented by the i^{th} packet features $[s, t, d]$. Thus, every feature represents a color channel. In our case, R is given for the size , G is given for the interarrival time, and B is given for the direction. This data representation is explained in detail in [38].

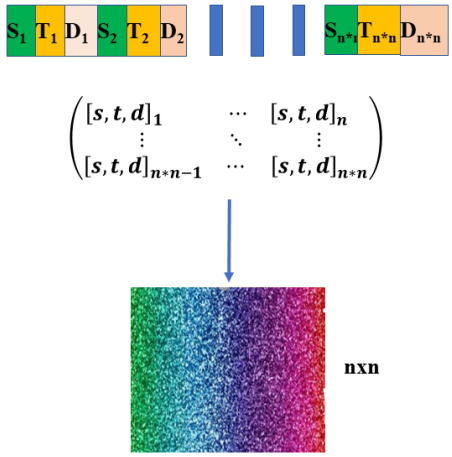

Fig. 4. Data Representation

4 Implementation

In this section, we detail the data collection and preprocessing, the model implementation, and the evaluation results.

4.1 Data Collection and Preprocessing

For collecting normal IoT traffic, we used a set of IoT devices, including Wi-Fi-enabled devices and hub-connected devices. The Wi-Fi devices are: D-Link HD

180-Degree Wi-Fi Camera DCS-8200LH, D-Link Wi-Fi Smart Plug DSP-W215, D-Link Wi-Fi Siren DCH-S220, and D-Link Wi-Fi Water Sensor DCH-S160, while the hub-connected devices are components of the Samsung SmartThings Home Monitoring Kit, including a motion sensor, a multi-purpose sensor, and a smart plug. These devices were installed in a home environment and were left to function normally. The Wi-Fi enabled devices are routed through a laptop to the Internet. A bridge is created at this laptop to forward the incoming traffic to the Ethernet interface connected to the home gateway. For the hub-connected devices, the hub is connected to the laptop by Ethernet and this laptop is configured to forward the incoming traffic to the wireless interface connected to the home gateway. In both cases, the traffic is collected at the laptop. One day of traffic for each device was considered for training and one day of traffic was used for testing.
For IoT attack traffic training data, we consider the dataset collected in [26], this dataset consists of multiple PCAP files for each type of attack. We choose one file for each type of attack for training and one file for testing.
To preprocess the data and extract the flows, the dpkt Python library was used [1]. In total, for training, we have 10320 flows of normal traffic and 3308 flows of attack traffic. For testing, we have 258 flows of normal traffic, 350 flows of attack traffic, and 2000 flows of (unknown) attack traffic. The normal traffic is categorized into five classes based on the device model while the attack traffic is categorized into three classes: data theft, denial of service and scanning.

4.2 Model Implementation and Training

The proposed model was implemented in Python using the tensorflow library [2]. The encoder, decoder and the discriminator consist of a 2-layer fully connected neural network with 1000 neurons each. The output layer of the decoder and the discriminator is a sigmoid layer, with the difference that the decoder output is of the same dimension of the input; however, the discriminator output is of dimension one. The model is trained with 100 epochs and a batch size of 100, the learning rate is 10^{-3}, and the momentum decay is 0.9 (beta1). The Rectified Linear Unit (ReLU) was used as the activation function for all hidden layers, and the weights are optimized using the Adam optimizer. For generating mutated data, we implement the mutation techniques listed in section III-B ((1) $\rightarrow$(10)). The training data is randomly mutated, where each data sample is uniformly mutated to one of the 10 mutation techniques.
For testing the effectiveness of the denoising process on the classification accuracy, we implement a CNN classifier with three convolutional layers, two max pooling layers, and two fully connected layers with one dropout layer with 50% dropout probability. Similarly, ReLU is applied for activation in the hidden layers, and the Adam algorithm is used for optimization with a learning rate of 10^{-3}. In addition, we apply cross validation for the classifier training with 4 folds. The performance metric used to evaluate the classifier is the accuracy, which is the ratio of the correctly classified samples to the total number of samples.

4.3 Evaluation and Results

For each of the experiments (1) →(10), the corresponding mutation technique is applied to the testing data. First, the traffic is passed to the discriminator. Then, it is passed to the denoising AE. Furthermore, the mutated and denoised traffic are passed to a classifier, that classifies the flow based on the device model for normal traffic and based on the attack type for attack traffic.
Samples of the mutated traffic and resulted images after denoising are included in Table 5. It is visually clear from the included images that the denoiser succeeds in learning the original data representation disregarding the mutation level. The difference between the denoised images and the original ones shows that the model does not overfit the data, however it learns the representation and the noise pattern.

For the normal traffic, the denoiser succeeds in recovering flows very similar to the original ones for most of the mutation techniques, except for the mutations (8) and (9). Similarly, for the attack traffic, the denoiser recovers the main flow representation, except for the mutation (2). In Table 1, the MSE between the original data and the mutated one (mutation degradation), the reconstruction loss, and the abnormality detection rate are reported for the normal traffic and the attack traffic.

Mutation	**Autoencoder Loss**		**Mutation Loss**		**Discrimination Rate**	
Technique	**Normal**	**Attack**	**Normal**	**Attack**	**Normal**	**Attack**
(1)	0.042	0.0357	0.2713	0.1698	100%	100%
(2)	0.03218	0.1569	0.0171	0.082	100%	100%
(3)	0.0149	0.1569	0.0009	0.0065	88.99%	83.13%
(4)	0.0543	0.0463	0.2704	0.1696	50%	49.94%
(5)	0.067	0.0519	0.0171	0.082	100%	100%
(6)	0.0621	0.0703	0.0129	0.0411	49%	49.88%
(7)	0.0378	0.1569	0.2974	0.3245	100%	100%
(8)	0.0595	0.0257	0.09878	0.1016	99.02%	100%
(9)	0.1728	0.0303	0.2825	0.0012	50%	41.59%
(10)	0.0369	0.0519	0.5687	0.4943	100%	100%

Table 1. Testing evaluation results

The results present a high detection rate for the different mutation techniques, except for the mutation techniques (4), (6), and (9). This can be explained by the fact that (6) and (9) are normal distribution based mutations. In these cases, the main traffic characteristics will remain unchanged and this will harden the differentiation between the original and mutated traffic. (4) is a mice-elephant mutation of the packet sizes. However, in our case (i.e. IoT traffic), most of the packets are of small size and therefore the mutation (4) will have limited affect on the traffic characteristics. Moreover, it can be noticed that the MSE between the recovered data and the original one (reconstruction loss)

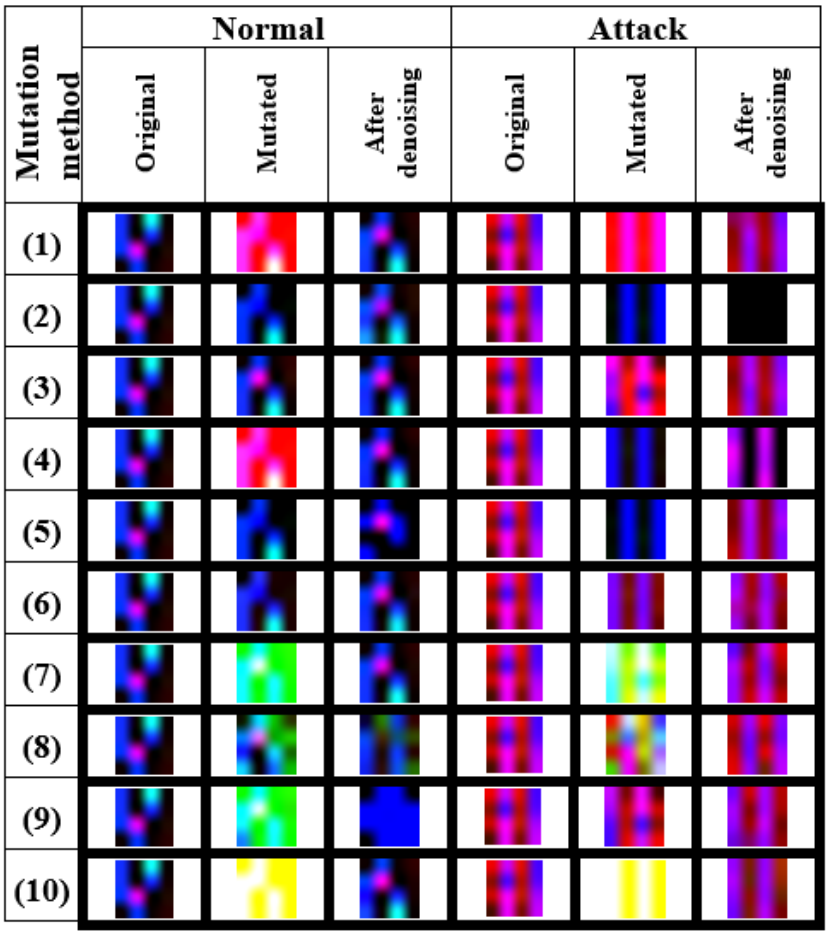

Fig. 5. Visualized images representing network traffic

is lower than the MSE between the mutated data and the original one (mutation degradation) in most of the cases. This means that the denoising process decreases the degradation effect by reconstructing a version of the data that is closer to the original one.

Table 2 presents the accuracy of the classification based on the traffic label: device model for normal traffic and attack type for attack traffic. The mutation techniques (1), (4), (7), and (10) affect the accuracy and misleads the classification noticeably in the normal traffic case; however, after denoising, the accuracy increases. However, for the techniques (5), (6), (8), and (9), the denoiser fails to reconstruct the original traffic. This is due to the fact that the randomness will create a denoised traffic of random type. Moreover, for the mutation technique

(2), the mutation is linear and this does not affect the CNN classifier accuracy, being immune to the linear degradation of the image. However, for statistical based machine learning methods, the mutation techniques (2), (5), and (6) affect noticeably the classification accuracy. To see the effect of the mutation on the statistical based classification, we include the results of the mutated and denoised traffic statistical based classification in Table2. The statistical classification uses a subset of the Moore features (see Table3) and Random Forest (RF) as classifier. It can be noticed that overall our representation method with CNN classifier outperforms the RF method before and after denoising in the normal traffic case. However, in the attack traffic case, our method gives better results after denoising.

Mutation Technique	**Before Denoising**				**After Denoising**			
	Normal		**Attack**		**Normal**		**Attack**	
	CNN	**RF**	**CNN**	**RF**	**CNN**	**RF**	**CNN**	**RF**
(1)	30.14%	54.47%	51.55%	**80.84%**	**65.19**%	38.78%	53.1%	37.05%
(2)	**93.13**%	25%	**36.27%**	36.24%	79.16%	68.68%	33.41%	33.41%
(3)	**99.01**%	87.19%	76.25%	**90.33**%	87.99%	78.79%	33.41%	33.41%
(4)	29.9%	55.2%	51.55%	**80.1%**	**67.4**%	40.68%	43.79%	42.57%
(5)	**93.13**%	25.06%	36.99%	36.21%	20.09%	20.09%	**63.12%**	38.21%
(6)	**95.09**%	25.06%	56.55%	**62.94%**	56.12%	33.88%	36.99%	42.69%
(7)	75.49%	**85.6%**	33.17%	19.63%	77.69%	55.39%	**33.41%**	**33.41%**
(8)	78.18%	**81.92%**	33.15%	15.96%	62.99%	41.85%	**62.05%**	38.48%
(9)	80.14%	**82.35%**	34.24%	39.64%	20.09%	20.09%	**62.76%**	30.31%
(10)	19.6%	28.18%	33.17%	25.65%	**78.43**%	49.14%	**60.6%**	43.22%

Table 2. Classification Results

5 Discussion and Future Work

The results show that unsupervised DL architectures are very powerful in learning the data representation. AE is a well-known DL generative model, that is highly effective in extracting compressed representation from image-type data. Consequently, after representing network traffic as images, we applied AE to extract the representation patterns from IoT traffic. Moreover, the capability of AE to denoise images is used to overcome the mutation technique challenges. The results show the effectiveness of the proposed method to recover the original traffic representation for different levels of mutation, some of which are rather severe (e.g. fixed packet size and IAT). In fact, the considered mutation techniques cause any statistical classifier to fail. However, one limitation of this work is that we assume that we know ahead of time the mutation technique used by the attacker, so we can decide to choose to apply the classifier before or after denoising. While we considered mutation techniques in this paper, we will study as future work additional obfuscation techniques. In addition to the abnormal

Feature	Description
total_fwd_pkt	Total packets in the forward direction
total_fwd_bytes	Total bytes in the forward direction
total_bck_pkt	Total packets in the backward direction
total_bck_bytes	Total bytes in the backward direction
min_pckt_size_fwd	The min packet size in the forward direction
mean_pckt_size_fwd	The mean packet size in the forward direction
max_pckt_size_fwd	The max packet size in the forward direction
std_pckt_size_fwd	The standard deviation packet size in the forward direction
min_pckt_size_bck	The min packet size in the backward direction
mean_pckt_size_bck	The mean size of packets in the backward direction
max_pckt_size_bck	The max packet size in the backward direction
std_pckt_size_bck	The standard deviation packet size in the backward direction
total_size	The total flow size
min_pckt_size	The min packet size in either direction
mean_pckt_size	The mean size of packets in either direction
max_pckt_size	The max packet size in either direction
std_pckt_size	The standard deviation packet size in either direction
min_iat_fwd	The minimum interarrival time in the forward direction
mean_iat_fwd	The mean interarrival time in the forward direction
max_iat_fwd	The maximum interarrival time in the forward direction
std_iat_fwd	The standard deviation interarrival time in the forward direction
min_iat_bck	The minimum interarrival time in the backward direction
mean_iat_bck	The mean interarrival time in the backward direction
max_iat_bck	The maximum interarrival time in the backward direction
std_iat_bck	The standard deviation interarrival time in the backward direction
total_time	The duration of the flow
min_iat	The minimum interarrival time in either direction
mean_iat	The mean interarrival time in either direction
max_iat	The maximum interarrival time in either direction
std_iat	The standard deviation interarrival time in either direction
avg_pckt_fwd	The average number of packets in the forward direction
avg_bytes_fwd	The average number of bytes in the forward direction
avg_pckt_bck	The average number of packets in the backward direction
avg_bytes_bck	The average number of bytes in the backward direction
avg_iat_fwd	The proportion of flow time in the forward direction to the total flow time
avg_iat_bck	The proportion of flow time in the backward direction to the total flow time

Table 3. Feature Set for statistical classification

traffic detection using the discriminative part of the GAN architecture, in the morphing case for example, we need to consider the generative part also. The generator will be trained to generate morphed traffic similar to the original one. In this case, the model convergence will not be a straightforward task.

6 Conclusion

Applying ML for network management and network security has gained a lot of interest in the last decade. However, new traffic obfuscation techniques have been developed to thwart classification and avoid detection in the sake of user privacy. These techniques have been employed by attackers to mount their attacks without being detected. While the obfuscation techniques might be detected by behavioral or statistical ML techniques, the recovery of the initial traffic is unfeasible. In this paper, inspired from a promising DL application, which is image denoising, we transform the traffic to images then combine two well-established DL architectures (AE and GAN) to reconstruct the original traffic and detect abnormal one. The test results show the effectiveness and robustness of the proposed model to detect abnormal traffic in all its variants.

Acknowledgments

Research funded by the AUB University Research Board, the Lebanese National Council for Scientific Research, and TELUS Corp., Canada.

References

1. dpkt, howpublished = https://dpkt.readthedocs.io/en/latest/, note = Accessed: 2019
2. tensorflow, howpublished = https://www.tensorflow.org/, note = Accessed: 2019
3. Alom, M.Z., Taha, T.M.: Network intrusion detection for cyber security using unsupervised deep learning approaches. In: 2017 IEEE National Aerospace and Electronics Conference (NAECON). pp. 63–69. IEEE (2017)
4. Baddar, S.A.H., Merlo, A., Migliardi, M.: Behavioral-anomaly detection in forensics analysis. IEEE Security & Privacy **17**(1), 55–62 (2019)
5. Beggel, L., Pfeiffer, M., Bischl, B.: Robust anomaly detection in images using adversarial autoencoders. arXiv preprint arXiv:1901.06355 (2019)
6. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials **18**(2), 1153–1176 (2015)
7. Buyun Qu, Zhibin Zhang, Le Guo, Xingquan Zhu, Li Guo, Dan Meng: An empirical study of morphing on network traffic classification. In: 7th International Conference on Communications and Networking in China. pp. 227–232 (Aug 2012). https://doi.org/10.1109/ChinaCom.2012.6417481
8. Callado, A.C., Kamienski, C.A., Szabó, G., Gero, B.P., Kelner, J., Fernandes, S.F., Sadok, D.F.H.: A survey on internet traffic identification. IEEE Communications Surveys and Tutorials **11**(3), 37–52 (2009)
9. Cao, Z., Xiong, G., Zhao, Y., Li, Z., Guo, L.: A survey on encrypted traffic classification. In: International Conference on Applications and Techniques in Information Security. pp. 73–81. Springer (2014)
10. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P.: Network intrusion detection for iot security based on learning techniques. IEEE Communications Surveys & Tutorials (2019)

11. Chaddad, L., Chehab, A., Elhajj, I.H., Kayssi, A.: App traffic mutation: Toward defending against mobile statistical traffic analysis. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 27–32 (April 2018). https://doi.org/10.1109/INFCOMW.2018.8406899
12. Chaddad, L., Chehab, A., Elhajj, I.H., Kayssi, A.: Mobile traffic anonymization through probabilistic distribution. In: 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). pp. 242–248 (Feb 2019). https://doi.org/10.1109/ICIN.2019.8685871
13. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019)
14. Chen, Z., He, K., Li, J., Geng, Y.: Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 1271–1276. IEEE (2017)
15. Cheng, T., Lin, Y., Lai, Y., Lin, P.: Evasion techniques: Sneaking through your intrusion detection/prevention systems. IEEE Communications Surveys Tutorials **14**(4), 1011–1020 (Fourth 2012). https://doi.org/10.1109/SURV.2011.092311.00082
16. da Costa, K.A., Papa, J.P., Lisboa, C.O., Munoz, R., de Albuquerque, V.H.C.: Internet of things: A survey on machine learning-based intrusion detection approaches. Computer Networks **151**, 147–157 (2019)
17. Creswell, A., Bharath, A.A.: Denoising adversarial autoencoders. IEEE transactions on neural networks and learning systems **30**(4), 968–984 (2018)
18. Dainotti, A., Pescape, A., Claffy, K.C.: Issues and future directions in traffic classification. IEEE network **26**(1), 35–40 (2012)
19. Deecke, L., Vandermeulen, R., Ruff, L., Mandt, S., Kloft, M.: Image anomaly detection with generative adversarial networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 3–17. Springer (2018)
20. Dyer, K.P., Coull, S.E., Ristenpart, T., Shrimpton, T.: Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In: Proceedings of the 2012 IEEE Symposium on Security and Privacy. pp. 332–346. SP '12, IEEE Computer Society, Washington, DC, USA (2012). https://doi.org/10.1109/SP.2012.28, https://doi.org/10.1109/SP.2012.28
21. Fadlullah, Z.M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., Mizutani, K.: State-of-the-art deep learning: Evolving machine intelligence toward tomorrows intelligent network traffic control systems. IEEE Communications Surveys & Tutorials **19**(4), 2432–2455 (2017)
22. Finsterbusch, M., Richter, C., Rocha, E., Muller, J.A., Hanssgen, K.: A survey of payload-based traffic classification approaches. IEEE Communications Surveys & Tutorials **16**(2), 1135–1156 (2013)
23. Ger, S., Klabjan, D.: Autoencoders and generative adversarial networks for anomaly detection for sequences. arXiv preprint arXiv:1901.02514 (2019)
24. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
25. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity **2**(1), 20 (Jul 2019). https://doi.org/10.1186/s42400-019-0038-7, https://doi.org/10.1186/s42400-019-0038-7
26. Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B.: Towards the development of realistic botnet dataset in the internet of things for network

forensic analytics: Bot-iot dataset. Future Generation Computer Systems **100**, 779 – 796 (2019). https://doi.org/https://doi.org/10.1016/j.future.2019.05.041, http://www.sciencedirect.com/science/article/pii/S0167739X18327687
27. Li, D., Chen, D., Goh, J., Ng, S.k.: Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:1809.04758 (2018)
28. Lin, Z., Shi, Y., Xue, Z.: IDSGAN: generative adversarial networks for attack generation against intrusion detection. CoRR **abs/1809.02077** (2018), http://arxiv.org/abs/1809.02077
29. Liu, H., Lang, B., Liu, M., Yan, H.: Cnn and rnn based payload classification methods for attack detection. Knowledge-Based Systems **163**, 332–341 (2019)
30. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
31. Munir, M., Siddiqui, S.A., Dengel, A., Ahmed, S.: Deepant: A deep learning approach for unsupervised anomaly detection in time series. IEEE Access **7**, 1991–2005 (2018)
32. Nguyen, T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE communications surveys & tutorials **10**(4), 56–76 (2008)
33. Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., Aguilar, J.: Towards the deployment of machine learning solutions in network traffic classification: a systematic survey. IEEE Communications Surveys & Tutorials **21**(2), 1988–2014 (2018)
34. Perrone, G., Vecchio, M., Pecori, R., Giaffreda, R.: The day after mirai: A survey on mqtt security solutions after the largest cyber-attack carried out through an army of iot devices. In: IoTBDS. pp. 246–253 (2017)
35. Qu, B., Zhang, Z., Zhu, X., Meng, D.: An empirical study of morphing on behavior-based network traffic classification. Security and Communication Networks **8**(1), 68–79 (2015). https://doi.org/10.1002/sec.755, https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.755
36. Rezaei, S., Liu, X.: Deep learning for encrypted traffic classification: An overview. IEEE communications magazine **57**(5), 76–81 (2019)
37. Rigaki, M., Garcia, S.: Bringing a gan to a knife-fight: Adapting malware communication to avoid detection. In: 2018 IEEE Security and Privacy Workshops (SPW). pp. 70–75 (May 2018). https://doi.org/10.1109/SPW.2018.00019
38. Salman, O., Elhajj, I.H., Chehab, A., Kayssi, A.: A multi-level internet traffic classifier using deep learning. In: 2018 9th International Conference on the Network of the Future (NOF). pp. 68–75 (Nov 2018). https://doi.org/10.1109/NOF.2018.8598055
39. Smit, D., Millar, K., Page, C., Cheng, A., Chew, H.G., Lim, C.C.: Looking deeper: Using deep learning to identify internet communications traffic. In: 2017 Australasian Conference of Undergraduate Research (ACUR) (2017)
40. Tripathi, S., Lipton, Z.C., Nguyen, T.Q.: Correction by projection: Denoising images with generative adversarial networks. arXiv preprint arXiv:1803.04477 (2018)
41. Umer, M.F., Sher, M., Bi, Y.: A two-stage flow-based intrusion detection model for next-generation networks. PloS one **13**(1), e0180945 (2018)
42. Verma, G., Ciftcioglu, E., Sheatsley, R., Chan, K., Scott, L.: Network traffic obfuscation: An adversarial machine learning approach. In: MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM). pp. 1–6 (Oct 2018). https://doi.org/10.1109/MILCOM.2018.8599680
43. Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 41525–41550 (2019)

44. Vu, H.S., Ueta, D., Hashimoto, K., Maeno, K., Pranata, S., Shen, S.M.: Anomaly detection with adversarial dual autoencoders. arXiv preprint arXiv:1902.06924 (2019)
45. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 43–48. IEEE (2017)
46. Wang, X., Du, Y., Lin, S., Cui, P., Yang, Y.: Self-adversarial variational autoencoder with gaussian anomaly prior distribution for anomaly detection. arXiv preprint arXiv:1903.00904 (2019)
47. Warde-Farley, D., Bengio, Y.: Improving generative adversarial networks with denoising feature matching (2016)
48. Xiao, Y., Xing, C., Zhang, T., Zhao, Z.: An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access **7**, 42210–42219 (2019)
49. Xinwen Fu, Graham, B., Bettati, R., Wei Zhao: On effectiveness of link padding for statistical traffic analysis attacks. In: 23rd International Conference on Distributed Computing Systems, 2003. Proceedings. pp. 340–347 (May 2003). https://doi.org/10.1109/ICDCS.2003.1203483
50. Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient gan-based anomaly detection. arXiv preprint arXiv:1802.06222 (2018)
51. Zhang, H., Yu, X., Ren, P., Luo, C., Min, G.: Deep adversarial learning in intrusion detection: A data augmentation enhanced framework. CoRR **abs/1901.07949** (2019), http://arxiv.org/abs/1901.07949