An unsupervised generative adversarial network based-host intrusion detection system for internet of things devices

Idriss Idrissi, Mostafa Azizi, Omar Moussaoui

MATSI Research Laboratory, Ecole Supérieure de Technologie, Mohammed First University, Oujda, Morocco

Article Info

ABSTRACT

Machine learning (ML) and deep learning (DL) have achieved amazing progress in diverse disciplines. One of the most efficient approaches is unsupervised learning (UL), a sort of algorithms for analyzing and clustering unlabeled data; it allows identifying hidden patterns or performing data clustering over provided data without the need for human involvement. There is no prior knowledge of actual abnormalities when using UL methods in anomaly detection (AD); hence, a DL-intrusion detection system (IDS)-based on AD depends intensely on their assumption about the distribution of anomalies. In this paper, we propose a novel unsupervised AD Host-IDS for internet of things (IoT) based on adversarial training architecture using the generative adversarial network (GAN). Our proposed IDS, called "EdgeIDS", targets mostly IoT devices because of their limited functionality; IoT devices send and receive only specific data, not like traditional devices, such as servers or computers that exchange a wide range of data. We benchmarked our proposed "EdgeIDS" on the message queuing telemetry transport (MQTTset) dataset with five attack types, and our obtained results are promising, up to 0.99 in the ROC-AUC metric, and to just 0.035 in the ROC-EER metric. Our proposed technique could be a solution for detecting cyber abnormalities in the IoT.

Corresponding Author:

Idriss Idrissi
MATSI Research, Laboratory, Ecole Supérieure de Technologie, Mohammed First University
BP 473 Campus Universitaire Al Qods, Oujda 60000, Morocco
Email: idrissi@ump.ac.ma

1. INTRODUCTION

In fields as diverse as image identification, self-driving cars, and playing sophisticated games, machine learning and its subset deep learning has made extraordinary progress during the previous decade [1]. These achievements were mostly obtained by using one of two learning paradigms: supervised learning or reinforcement learning to construct deep neural networks. Both approaches necessitate the creation of training signals by humans and their transmission to computers [2]. These are the "targets" (such as the proper label for an image) in supervised learning, and the "rewards" for successful behavior in reinforcement learning (such as getting a high score in a game). As a result, human trainers set the learning boundaries. Unsupervised learning is a paradigm for developing autonomous intelligence in which agents (such as computer programs) are rewarded for learning about the material they perceive without a specific purpose in mind [2].

Anomaly detection systems are designed either manually by specialists defining data thresholds or automatically by learning from existing data using machine learning or deep learning techniques. Building an anomaly detection system manually is time-consuming [3]. This cannot be an acceptable solution for an environment where data evolves over time, such as in the internet of things (IoT) cybersecurity environment.

Companies have been pushed to build remote workforces and operate on cloud-based platforms as a result of COVID-19. The introduction of 5G has made connected devices more linked than ever. In brief, the cybersecurity business has never been more vital. The year 2020 surpassed all records in terms of data breaches and the sheer volume of cyberattacks on organizations, governments, and individuals. The most recent hack (June 2021) knocked off electricity to over 800,000 Puerto Ricans, and a massive fire took out power over the island. This cyberattack was a distributed denial of service (DDoS) attack, which causes online services to become unavailable by flooding them with connection requests. Moreover, two million visitors per second were logged during the attack, locking out many users.

When implementing unsupervised learning methods in anomaly detection looking forward to building an IDS, there is no prior knowledge of actual anomalies. These methods rely significantly on their assumptions about anomaly distribution. Collecting labeled normal data and some labeled anomalous data, on the other hand, is frequently not difficult. In practice, it is frequently recommended to make use of as much freely available labeled data as feasible. As a result, learning expressive representations of normality/abnormality from such labeled data is critical for accurate anomaly identification [4].

Aspects of adversarial auto-encoders (AAE) and generative adversarial network (GAN) are used in some of the techniques for anomaly detection; the best-known ones are AnoGAN [5], efficient GAN based anomaly detection (EGBAD) [3], GANomaly [6], and Skip-GANomaly [7]. AnoGAN firstly proposed this concept but at first, there were some performance issues with this approach hereafter bidirectional GAN (BiGAN) based approach was proposed. Also, efficient GAN based anomaly detections (EGBADs) performed better than AnoGAN. There is also GANomaly a highly inspired by AnoGAN, BiGAN, and EGBAD. This architecture trains a generator on normal data so that they can learn their manifold and the autoencoder is likewise trained at the same time to learn the encoding of the data in their hidden representation proficiently. This architecture includes a generator and a discriminator like the typical GAN architecture. Correspondingly, Skip-GANomaly, a new unsupervised anomaly detection architecture within an adversarial training structure based on GANomaly inspects the role of skip connections within the generator and feature extraction from the discriminator for the manipulation of hidden features. Skip-GANomaly outperforms previous state-of-the-art techniques in terms of numerical findings [8], [9].

In this paper, we propose a new unsupervised anomaly detection host-intrusion detection system for IoT, called "EdgeIDS" built on an adversarial training over a skip-connected encoder-decoder (convolutional neural) network architecture using the generative adversarial network; in other words, this is a specific IDS for IoT based on the Skip-GANomaly technique. This proposal is destined for IoT devices, because every device has limited functionality, for example, a camera sends a video stream and receives a limited type of data (commands); anything, it receives out of the ordinary, could be considered as an anomaly. It is similar for other devices such as smart doors, e-health sensors, or weather sensors that send and receive only specific data (telemetries and commands), and sometimes firmware updates; these ordinary exchanged data are recognized as normal traffic. Not like classical machines like servers or computers that send and receive a variety of data [10]. The rest of the paper is organized as follows. The second section presents a background of the different methods used in our proposal. The third section discusses related works. In the fourth section, we highlight the key points of our proposed method. Before concluding, we show our obtained results in the fifth section.

2. BACKGROUND
2.1. Unsupervised learning
It is a sort of algorithm that analyzes and clusters unlabeled datasets. It learns hidden patterns or makes data clustering without the necessity for human involvement, by discovering resemblances and differences in the given data. It is a widely used exploratory data analysis solution, like for cross-selling strategies, client segmentation, image, or voice recognition [11]. The unsupervised scenario is the most complex, as it is becoming increasingly difficult for modelers to cope with ever-increasing amounts of dark data. The dataset in the unsupervised case is not guaranteed to be clustered in two data groups "normal" and "abnormal", as in a binary classification (as in our EdgeIDS).

2.2. Generative adversarial network (GAN)
Introduced in 2014 by Goodfellow et al. [12], GANs are techniques of generative modeling using deep learning approaches, especially convolutional neural networks (CNN). They are quoted as the most interesting contribution in the last 10 years in ML/DL [13]. They are used mostly in images, video, and voice generation. GANs are an unsupervised learning technique that uses two neural networks (generator and discriminator) opposing one on the other. The generator attempts to generate realistic data, with the intention of tricking the discriminator into thinking that the generated data is genuine [14]. On the other hand, the discriminator [14] is a sort of a classifier that attempts to develop the ability to recognize real data from the

fake ones created by the generator (see Figure 1). It can use any network architecture suitable for classified data types [15].
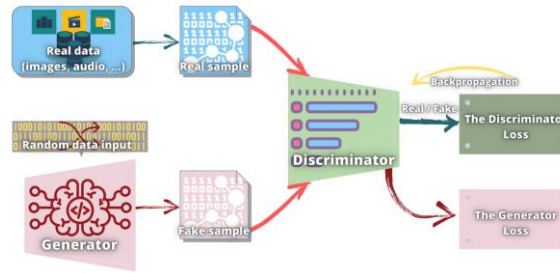

Figure 1. Generative adversarial network architecture

2.3. Skip connections

Extra connections between nodes in different levels of a neural network (NN) that skip one or more nonlinear processing layers are known as skip connections. The use of skip (or residual) connections has significantly enhanced the training of very deep neural networks [7]. Many convolutional architectures now provide skip connections as a standard module. We can give a different path for the gradient by utilizing a skip connection (with backpropagation). These extra pathways are often useful for model convergence. In DL architectures with skip connections, skip some layers in the NN and feed the output of one layer as the input to the following levels (see Figure 2), as the name suggests (instead of only the next one) [16].
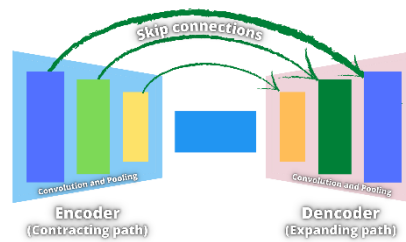

Figure 2. Skip connections architecture

2.4. Adversarial auto-encoder (AAE)

AAE is a probabilistic auto-encoder that merges the auto-encoder architecture with the adversarial loss concept introduced by GAN but is different from GAN. The output of its generator is the produced image, and the input for its discriminator is both the real and fake images, where the AAE generator generates a latent code, it tries to fool the discriminator into thinking the latent code is sampled from the given dataset. Alternatively, the discriminator will predict whether a given hidden code is generated by the autoencoder (fake) or some random vector sampled from the normal dataset (real) (see Figure 3 and Figure 4).
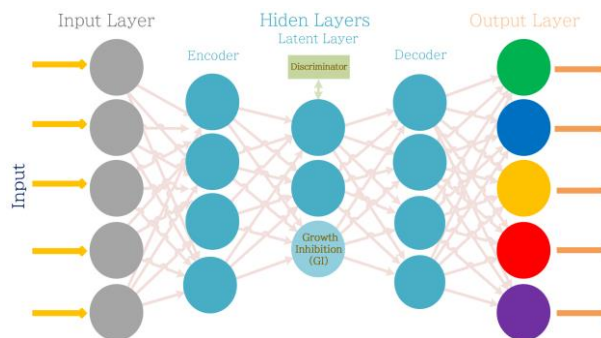

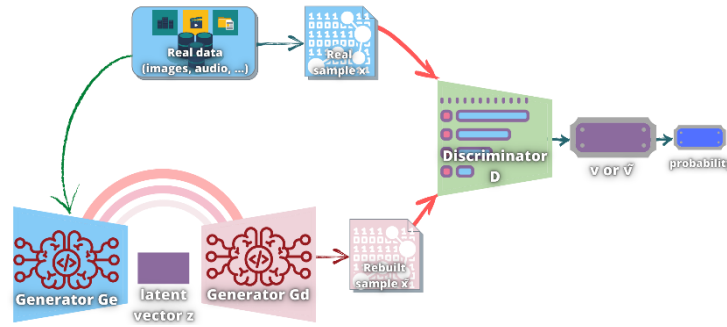Figure 3. Adversarial autoencoder layers

Figure 4. Skip-GANomaly architecture

One of the AAE applications is the anomaly detection. While the lack of the anomaly data is a major problem in the anomaly detection, the unsupervised technique to detect the anomaly is an appropriate solution. Where an auto-encoder can be trained to rebuild an anomaly data (mostly images) to a normal one. Afterward, the anomaly can be detected by calculating the difference between the rebuilt data without the anomaly and the original anomaly data [17]. Using AAE, the autoencoder's performance can be increased with an adversarial loss.

2.5. Skip-GANomaly

It is a new approach for anomaly detection using an adversarial training. It contains a generator G and a discriminator D just like the typical GAN architecture (see Figure 4) [7]. The generator G is a bow-tie network that contains an encoder Ge and a decoder Gd. The encoder Ge is capable of plotting a high-dimensional image P to a lower-dimensional latent vector z such that $Ge : x \rightarrow z$ ($x \in R^{w \times h \times c}, z \in R^d$). Contrariwise, the decoder Gd reallocates the latent vector V to image space such that $Gd : z \rightarrow \tilde{x}$ ($\tilde{x} \in R^{w \times h \times c}, z \in R^d$). The generator G can rebuild the image through the process of "encode–decode" such that $G : x \rightarrow z \rightarrow \tilde{x}$. Concerning the discriminator D, its mission is to differentiate between the real images x and the rebuilt ones x. The discriminator D is also a feature extractor for the input image, which can be either the real original or the rebuilt ones, and so the extracted latent representation is $v = f(x)$ or $\tilde{v} = f(\tilde{x})$ ($v, \tilde{v} \in R^d$).

In both Ge and Gd, five blocks are present. Convolution and BatchNorm layers are included in all the blocks of Ge, as well as the leaky ReLU activation function, which is utilized to downsample the input. Likewise, every block in Gd with transposed convolution and BatchNorm layers, as well as the leaky ReLU activation function, upsamples the input in the same way. To extract features at different scales more effectively, a skip connection similar to the model is introduced between Ge and Gd, ensuring that each down-sampling block is concatenated with its corresponding up-sampling block. For the discriminator, the same structure as with deep convolutional GAN (DCGAN) [18] is taken into account.

The adversarial loss L_{adv} leads the Generator to generate realistic data (usually images) that deceive the discriminator D (1). The contextual loss L_{con} directs the generator to develop data that are contextually sound rather than ones that will trick the discriminator D. To this end, the input and output images are pixel-by-pixel compared (2). The latent loss L_{lat} directs the encoders within the generator and discriminator to build strong latent representations of the input and generated images (3). Hence, the total training objective is a weighted sum of the losses mentioned above (4).

$$L_{adv} = \sum_{x \sim p_x}[\log D(x)] + \sum_{x \sim p_x}[\log(1 - D(\hat{x})] \tag{1}$$

$$L_{con} = \sum_{x \sim p_x} |x - \hat{x}|_1 \tag{2}$$

$$L_{lat} = \sum_{x \sim p_x} |f(x) - f(\hat{x})|_2 \tag{3}$$

$$L = \omega_{adv} L_{adv} + \omega_{con} L_{con} + \omega_{lat} L_{lat} \tag{4}$$

Where ω_{adv}, ω_{con}, and ω_{lat} are the weighting parameters that modify the prominence of individual losses in relation to the overall objective function.

2.6. Intrusion detection system (IDS)

They are dedicated devices or just computer programs that are able to track traffic across an entire network. There exists a multitude of IDS systems. According to the different taxonomies of IDS systems, we distinguish the network intrusion detection system (NIDS) [19], an IDS capable of analyzing incoming

network traffic, and the host intrusion detection system (HIDS), an IDS capable of monitoring sensitive operating system files, and Hybrid IDS solution that combines the two solutions to ease the weakness of the other two categories [20].

There are also different detection methodologies within these systems (see Figure 5). Thus, IDSs are most often classified into three categories: one group the detection systems is done by signatures; an IDS that search for specific patterns with a database of known attack signatures, the other, the detection system by anomalies; an IDS that search for possible threats through a system analysis and classification of anomalies, mostly using machine or deep learning techniques, and the third and last one is a hybrid detection solution that also combines the two solutions to ease the weakness of the other two categories [21].
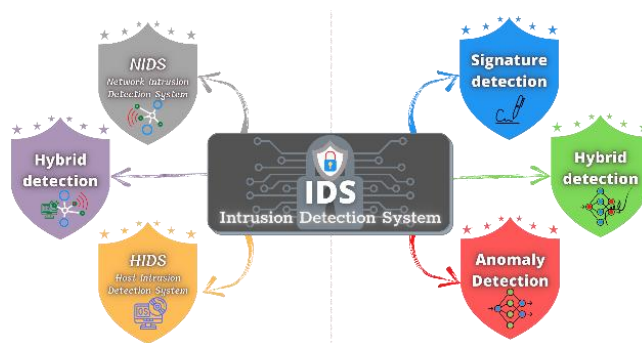

Figure 5. IDS classes

3. RELATED WORKS

Belenko *et al*. [22] examined the GAN's applicability for intrusion detection and found it to be more promising than a traditional ANN for dealing with real-world problems. They came to the conclusion that GAN-based networks may be used to look for security anomalies and cyber hazards, as well as to generate more anomalies to improve the quality of the flagged data samples. Ferdowsi and Saad [23] proposed a distributed GAN-based IDS solution in their study that can detect IoT intrusion with minimal reliance on a central unit. Every IoT devices (IoTD) may examine its own data as well as neighboring IoTDs in their method to identify internal and external threats. Their proposed distributed IDS that do not really require an IoTD to share datasets, making it realistic to use in IoT where we should protect user data privacy. They have demonstrated analytically that their distributed IDS outperforms a solo IDS that only has access to a single IDS' dataset. In comparison to a standalone IDS, simulation findings revealed that the proposed distributed GAN-based IDS has up to 20% greater accuracy, 25% better precision, and 60% lower false-positive rate.

Salman *et al*. [24] used two well-known DL architectures (AE and GAN) to recreate the original traffic and detect aberrant traffic, which was inspired by a promising DL application, called picture denoising. Their tests show that the proposed approach is effective and robust in detecting abnormal traffic in all of its variations. Yuan *et al*. [25] proposed a technique for converting network traffic data to images and detecting anomalous traffic using CNN. The trained classifier would be put on smart home edge nodes. In addition, their proposed method (AC-GAN) is utilized to generate synthesized network traffic samples that will be utilized to train the classifier to balance the amount of data between the minor and major classes in the intrusion detection dataset. They used the UNSW-NB15 dataset to evaluate our scheme's performance. The results demonstrate that this technique could improve network traffic classification precision, particularly for minor threat types. When compared to other methods, the utilized classifier performed well in the classification of normal and abnormality.

Shahriar *et al*. [26] proposed a GAN-assisted IDS that outperforms a standalone IDS for an unbalanced dataset or any developing domain of cyber-physical systems with limited data for model training. They tested the model on the NSL KDD'99 benchmark dataset. Even after being trained with a tiny initial dataset, experimental research demonstrates that the proposed G-IDS framework predicts with greater accuracy than independent IDS.

4. PROPOSED METHOD

"EdgeIDS" is our proposed anomaly detection host IDS, which is created by learning just normal traffic from a dataset in an unsupervised learning, then validating this model on both normal and abnormal data (an attack class from the dataset). In an IoT environment, this solution will be deployed in every edge

device within the edge node (see Figure 6). With such a deployment, the proposed "EdgeIDS" will be constrained to analyze the network's inbound data in real-time, assuming that this device is legitimate and trustworthy; otherwise, a second network IDS "DL-NIDS" must be deployed on the fog node to reinforce security on a device with more computation power, preferably connected to a lightweight hardware accelerator for deep NN. Building the unsupervised anomaly detection host-IDS for IoT called "EdgeIDS" on an adversarial training over a skip-connected encoder-decoder (convolutional neural) network architecture using the GAN approach consists of a three-phases process, while repeatedly working each time with a different class of attacks to get more results of analytics (Figure 7).
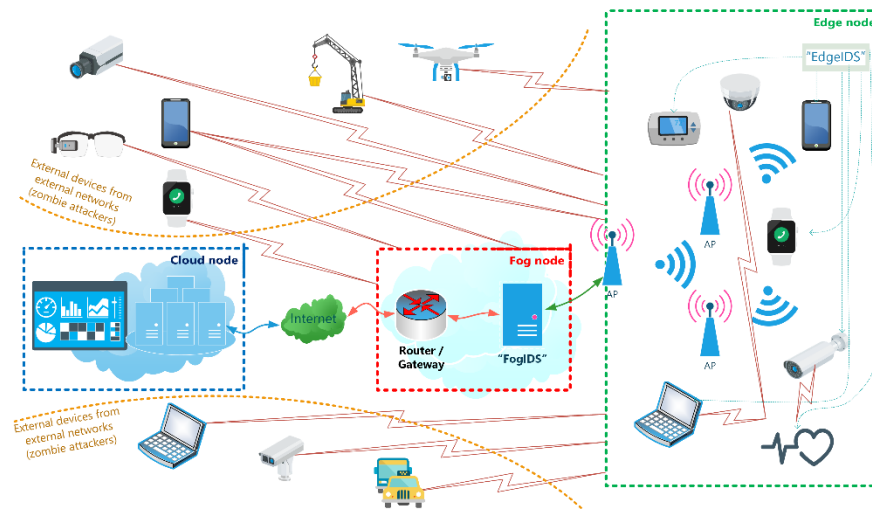

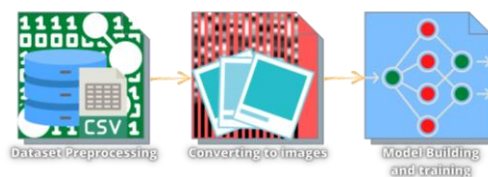Figure 6. Architecture of proposed approach

Figure 7. EdgeIDS building steps

i)   First Phase: Dataset Preprocessing, cleaning data and identifying features are all part of the preprocessing dataset process. Working features into a model with comparable distributions but drastically different means, or on radically different scales, might result in inaccurate predictions. To reduce considerable disparities in mean and variance, a common solution to these difficulties is to first "normalize" characteristics. Because the term "normalization" has come to mean various things in statistics, it can be confusing. Normalization procedures, on the other hand, all have one thing in common: they align distinct datasets for easy comparison. Because the terminology "normalization" has come to mean several things in statistics, it can be confusing. Normalization procedures, on the other hand, all have one thing in common: they align distinct datasets for easy comparison. One well-known normalization approach is re-scaling, which stretches and squeezes the values in the datasets to fit on a scale from 0 to 1. This normalization method will remove the units that were previously applied to the datasets, and it is useful when comparing datasets with different factors or units, such as miles to meters, or in our case, while trying to squeeze data between 0 and 255. In other words, this normalization is useful for our case when transforming numerical data into image pixels, we did this transformation using the "sklearn.preprocessing" package, "NumPy" library, in order to achieve the best deep learning model performance.

ii) Second Phase: Converting to images, after normalizing the values, we convert each comma-separated values (CSV) line to an image with 33 pixels. The number of pixels is the number of features of the used dataset, and can be varied on other datasets, in the conversion process, we used the Python imaging library (PIL), and the used format of the image is portable network graphics (PNG); an extensible image format with lossless compression that is open source. The main reason why we converted the dataset into

images is to preserve the integrity of the used algorithm in their proposed original form in a proofing concept way (see Figure 8).

Figure 8. Converted image

iii) Third Phase: Model Building and training, the model is initially fitted to a training dataset; a subset of the dataset using parameters to improve model performance; these parameters have adjusted during the training process to provide better results.

Hyper-parameter tuning is essential to ensure that the model performs effectively for each dataset and architecture. Hyper-parameters affect how quickly or efficiently the objective function can be reached. The major parameters tuned for Skip-GANomaly are the loss weights (ω_{adv}, ω_{con}, and ω_{lat}) and the size of the latent vector z. The size of z determines the quantity of information kept in z and, as a result, the encoder loss. And unquestionably the number of training epochs.

In our research, we worked with the message queuing telemetry transport (MQTTset) [27], [28], a dataset related to the MQTT IoT communication protocol. The dataset was collected in an IoT environment that contains one MQTT broker and eight sensors in a smart home where these sensors collect several information such as temperature, humidity, CO-Gas, smoke, light, motion, fan, and a door at different time intervals since the behavior of each sensor is different with the others. MQTTset was built using IoT-Flock [29], an open-source IoT traffic generation tool that emulates IoT devices and networks based on the MQTT and constrained application protocol (CoAPs). As well, IoT-Flock is capable of generating IoT normal and attack traffics over a real-time network using a single physical IoT machines. This dataset contains six labels: a normal traffic labeled as "Legitimate", and five attacks "SlowITe", "Bruteforce", "Malformed data", "Flooding", and "DoS attack". As mentioned before, we repeatedly trained and tested our proposed approach each time with a different class of attacks to get more results analytics, on the following classes distributions in Figure 9, We trained the model exclusively on legitimate traffic (normal data) in each trial, but we tested it on a set of legitimate traffic (normal) and another set from an attack class in the validation process, with the purpose to determine whether the model could distinguish legitimate traffic from an attack thereby detecting abnormalities.

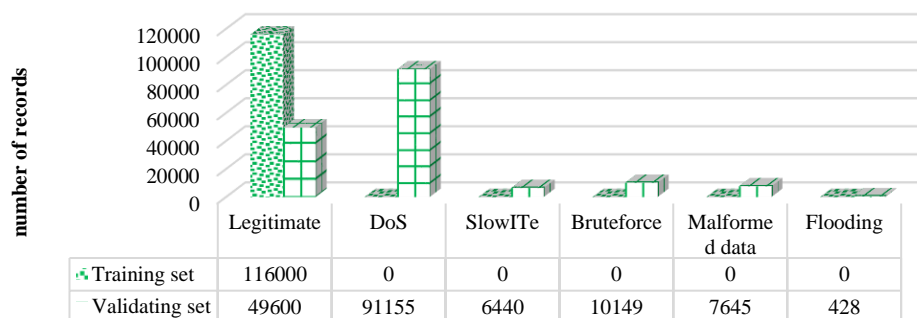|	Legitimate	DoS	SlowITe	Bruteforce	Malformed data	Flooding
Training set	116000	0	0	0	0	0
Validating set	49600	91155	6440	10149	7645	428

Figure 9. Distribution of dataset classes

5. RESULTS AND DISCUSSION
5.1. Hardware characteristics
Our findings were achieved using PyTorch (1.3.1) on a high-performance computing (HPC) infrastructure with the following hardware specifications:

a) CPU: two Intel Xeon Gold 6148 (2.4 GHz/20 cores)
b) RAM: 192 GB
c) GPU: two NVIDIA Tesla P100 (12 GB) with Cuda v10.1

        PyTorch is an open-source Python software library for machine learning developed by Facebook, that was derived from the Torch library (which is used with the Lua language). It is an enhanced tensor library for deep learning using GPUs and CPUs. The calculations are optimized and executed by the processor (CPU) or, when possible, by a graphics processor (GPU) supporting compute unified device architecture (CUDA). It offers two high-level features; a tensor computation (similar to NumPy) through a robust GPU acceleration, and a Deep neural network built on a tape-based autograd system. Our code is largely based on the "CycleGAN and pix2pix in PyTorch" and "Skip-GANomaly".

5.2. Evaluation Metrics

        We used the receiver operating characteristic (ROC) curve to evaluate our models. it is a graph that displays a classification model's performance across all categorization levels. The rate of true positives is plotted as a function of the rate of false positives on this curve:

−   True positive rate (TPR): is the recall's equivalence. As a function, it is defined by (5):

$$TPR = \frac{TP}{TP+FN} \qquad (5)$$

−   False positive rate (FPR): it is defined by (6):

$$FPR = \frac{FP}{FP+FN} \qquad (6)$$

with:
a)   True positive (TP): is the number of positive class records correctly classified.
b)   True negative (TN): is the number of negative class records correctly classified.
c)   False positive (FP): is the number of negative class records wrongly classified.
d)   False negative (FN): is the number of positive class records wrongly classified.

        At varied categorization thresholds, a ROC curve plots TPR vs. FPR. As the classification threshold is lowered, more items are classified as positive, resulting in an increase in both false positives and true positives. A typical ROC curve is depicted in the diagram below. We remind here that accuracy is calculated on anticipated classes, whereas ROC AUC is calculated on projected scores [30]. It implies that we will need to figure out what is the best threshold for a given problem. Furthermore, accuracy takes into account the percentage of correctly assigned positive and negative classes. If the problem is substantially imbalanced, as it is in this example, we can predict that all observations belong to the majority class and receive a very high accuracy score, that is why we are evaluating our models on the ROC AUC metric.

5.3. Evaluating the results

        As shown in Table 1 and Figures 10-15, the proposed technique performed better in 'Bruteforce' of the 'Malformed data' attack classes; around 0.990 in AUC. Furthermore, while its AUC was not the highest in the other categories such as 'DoS', 'SlowITe', and 'Flooding' (more than 0.965), it was nearly the same as or higher than the other state-of-the-art IDS methods. The main possible cause for this degradation from results of the other best performing classes is due to the number of samples in the testing set, and due to their big resemblance to legitimate traffic.

Table 1. ROC-AUC and ROC-EER obtained values for each class

Attack	AUC	EER
DoS	0.965	0.085
Bruteforce	0.990	0.035
SlowITe	0.966	0.081
Malformed data	0.989	0.053
Flooding	0.965	0.101
Abnormal	0.975	0.078

        Our model's performance is in height when the ROC curve approaches the upper left corner, as shown in Figures 10-15. We may deduce that both FN and FP are tending to be 0 based on their formula when the coordinate of the top left corner point is (0,1), which suggests that TPR=1 and FPR=0. As a result, this situation is the case for almost all the testing samples to be identified correctly. For every class of attack, we used to train our "EdgeIDS" model to predict its probability of abnormality. By comparing with their

binary ground-truth labels, we plotted the ROC curve as shown in Figures 10-15. We obtained the highest ROC-AUC value of 0.990 for the 'Bruteforce' class and 0.989 in the AUC value for the 'Malformed data', and just 0.035 and 0.053 respectively for the ROC-EER. Which stands for "receiving operating characteristic equal error rate," and it is the accuracy at the ROC operating point in which the false positive and false negative rates are equal. For the other classes, the ROC-EER was nearly low around 0.08; combined with the ROC-AUC, the models for these classes averagely have a good performance in detecting the abnormalities. Similarly, for the 'Abnormal' class; a class that combines all the attacks obtains a ROC-AUC value of 0.975 and a ROC-EER value of 0.078.
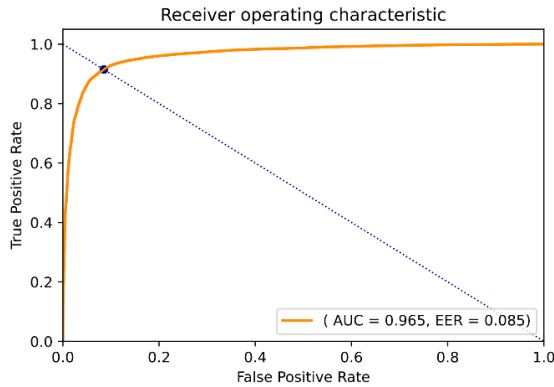
Figure 10. ROC metrics for the "DoS" attack class
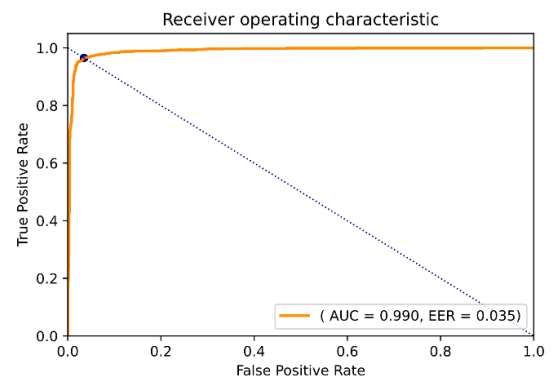
Figure 11. ROC metrics for the "Bruteforce" attack class
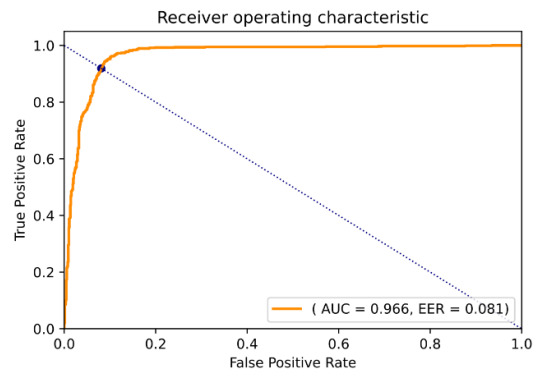
Figure 12. ROC metrics for the "SlowITe" attack class
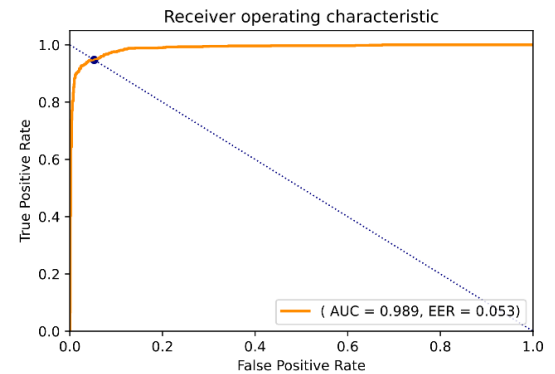
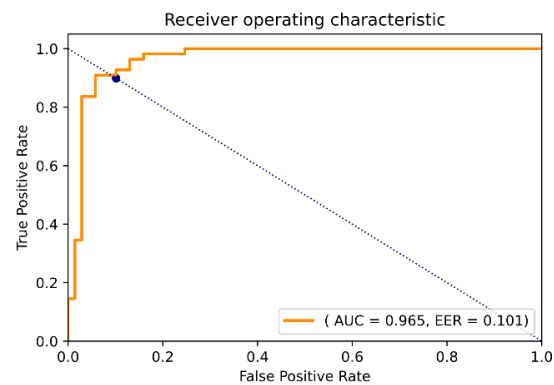Figure 13. ROC metrics for the "Malformed data" attack class

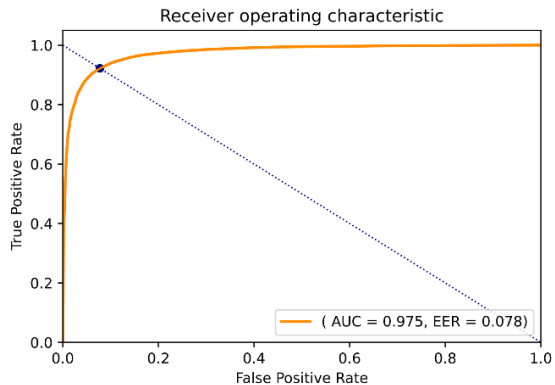Figure 14. ROC metrics for the "Flooding" attack class

Figure 15. ROC metrics for the "Abnormal" class

We also kept track of the model training duration, which took around 6 minutes every iteration (give or take); we got the best results for each class in varied iteration numbers ranging from five to twenty. The iteration timing was the same for each class because we trained our model on the same train set (legitimate traffic), but the timing was varied while testing on the anomalous data owing to imbalanced data between the classes. We notice that our generated "EdgeIDS" has a strong capacity for identifying abnormalities based on these outcomes (attacks).

6.   CONCLUSION

The necessity for IoT security has been highlighted by a series of high-profile cases in which a common IoT device was used to access and attack a larger network. It is crucial for assuring the security of networks that have IoT devices attached to them. IoT security encompasses a wide range of tactics, strategies, protocols, and actions aimed at mitigating modern enterprises' growing IoT risks. By exploiting the generative adversarial network (Skip-GANomaly), this paper proposes a specific IDS, called "EdgeIDS", for IoT devices. This proposed "EdgeIDS" outperforms quantitatively state-of-the-art methods. The experimental results in this study shed light on the suggested method's capacity to detect anomalies traffic thus attacks in an IoT environment. Because most IoT devices have limited capabilities, the proposed "EdgeIDS" will be compelled to analyze only the network's inbound data in real-time. In addition, another network IDS "DL-NIDS" must be deployed on the fog node to strengthen security on a device with higher processing power, ideally coupled to a lightweight hardware accelerator.

ACKNOWLEDGEMENTS

REFERENCES

[1]   M. Boukabous and M. Azizi, "A comparative study of deep learning based language representation learning models," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 1032-1040, 2020, doi: 10.11591/ijeecs.v22.i2.pp1032-1040.
[2]   A. Graves and K. Clancy, Unsupervised learning: the curious pupil, 2019, Accessed: Jun. 10, 2021. [Online]. Available: https://deepmind.com/blog/article/unsupervised-learning
[3]   H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," *arXiv preprint arXiv:1802.06222*, 2018.
[4]   G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep Learning for Anomaly Detection: A Review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1-38, 2021, doi: 10.1145/3439950.
[5]   T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*, vol. 10265, 2017, pp. 146–147, doi: 10.1007/978-3-319-59050-9_12.
[6]   S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised Anomaly Detection via Adversarial Training," *Lecture Notes in Computer Science*, vol. 11363, pp. 622–637, 2019, doi: 10.1007/978-3-030-20893-6_39.
[7]   S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection," *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1-8, doi: 10.1109/IJCNN.2019.8851808.
[8]   A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial Autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.
[9]   Y. Zhu, D. Chen, L. Yang, G. Yuan, R. Wei, and Y. Hu, "Defect detection of Aluminum Conductor Composite Core (ACCC) wires based on semi-supervised anomaly detection," *Energy Reports*, vol. 7, pp. 183–189, 2021, doi: 10.1016/j.egyr.2021.01.095.
[10]   I. Idrissi, M. Azizi, and O. Moussaoui, "A Lightweight Optimized Deep Learning-based Host-Intrusion Detection System Deployed on the Edge for IoT," *Int. J. Comput. Digit. Syst.*, 2021.
[11]   IBM Cloud Education, What is Unsupervised Learning?, IBM, 2020. Accessed: Jun. 04, 2021. [Online]. Available: https://www.ibm.com/cloud/learn/unsupervised-learning
[12]   I. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Jun. 2020, doi: 10.1145/3422622.
[13]   A. Kherraki and R. El Ouazzani, "Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 1, pp. 110–120, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp110-120.
[14]   K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. -Y. Wang, "Generative adversarial networks: introduction and outlook," in *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588-598, 2017, doi: 10.1109/JAS.2017.7510583.
[15]   M. Berrahal and M. Azizi, "Augmented Binary Multi-Labeled CNN for Practical Facial Attribute Classification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 973–979, 2021, doi: 10.11591/ijeecs.v23.i2.pp973-979.
[16]   L. -F. Dong, Y. -Z. Gan, X. -L. Mao, Y. -B. Yang, and C. Shen, "Learning Deep Representations Using Convolutional Auto-Encoders with Symmetric Skip Connections," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 3006-3010, doi: 10.1109/ICASSP.2018.8462085.
[17]   K. Hara and K. Shiomoto, "Intrusion Detection System using Semi-Supervised Learning with Adversarial Auto-encoder," *NOMS 2020-2020 IEEE/IFIP Net. Operat. Management Symposium*, 2020, pp. 1-8, doi: 10.1109/NOMS47738.2020.9110343.

[18] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv preprint arXiv:1511.06434*, 2015.

[19] I. Idrissi, M. Azizi, and O. Moussaoui, "Accelerating the Update of a DL-Based IDS for IoT Using Deep Transfer Learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1059–1067, 2021, doi: 10.11591/ijeecs.v23.i2.pp1059-1067.

[20] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for iot against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 110–120, 2021, doi: 10.11591/ijai.v10.i1.pp110-120.

[21] I. Idrissi, M. Azizi, and O. Moussaoui, "IoT security with Deep Learning-based Intrusion Detection Systems: A systematic literature review," *2020 4th Int. Conf. Intel. Comp. Data Scie. (ICDS)*, 2020, pp. 1-10, doi: 10.1109/ICDS50568.2020.9268713.

[22] V. Belenko, V. Chernenko, M. Kalinin, and V. Krundyshev, "Evaluation of GAN Applicability for Intrusion Detection in Self-Organizing Networks of Cyber Physical Systems," *2018 International Russian Automation Conference (RusAutoCon)*, 2018, pp. 1-7, doi: 10.1109/RUSAUTOCON.2018.8501783.

[23] A. Ferdowsi and W. Saad, "Generative Adversarial Networks for Distributed Intrusion Detection in the Internet of Things," *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9014102.

[24] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, "Denoising Adversarial Autoencoder for Obfuscated Traffic Detection and Recovery," *Int. Conf. Machine Learning for Networking*, vol. 12081, pp. 99–116, 2020, doi: 10.1007/978-3-030-45778-5_8.

[25] D. Yuan *et al*., "Intrusion Detection for Smart Home Security Based on Data Augmentation with Edge Computing," *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1-6, doi: 10.1109/ICC40277.2020.9148632.

[26] M. H. Shahriar, N. I. Haque, M. A. Rahman, and M. Alonso, "G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System," *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 376-385, doi: 10.1109/COMPSAC48688.2020.0-218.

[27] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a new dataset for machine learning techniques on MQTT," *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–17, 2020, doi: 10.3390/s20226578.

[28] MQTTset, Kaggle. Accessed: Jun. 07, 2021. [Online]. Available: https://www.kaggle.com/cnrieiit/mqttset

[29] S. Ghazanfar, F. Hussain, A. U. Rehman, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT-Flock: An Open-source Framework for IoT Traffic Generation," *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 2020, pp. 1-6, doi: 10.1109/ICETST49965.2020.9080732.

[30] J. Czakon, F1 Score vs ROC AUC vs Accuracy vs PR AUC : Which Evaluation Metric Should You Choose ?, Neptuneblog, 2021. Accessed: Jun. 15, 2021. [Online]. Available: https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc

BIOGRAPHIES OF AUTHORS

Idriss Idrissi 🆔 🅖 SC Ⓟ is a Ph.D. candidate in Computer Engineering at Mohammed First University in Oujda, Morocco, where he is researching internet of things security using Deep Learning. He has an M.Sc. degree in internet of things from Sidi Mohamed Ben Abdellah University in Fez, Morocco (2019), a B.Sc. degree in Computer Engineering from Mohammed First University (2016). Additionally, he holds several certifications in networking, artificial intelligence, cybersecurity, and programming. Also, he was a reviewer for various international conferences and journals. And is currently employed as an administrative at Mohammed First University.

Prof. Dr. Mostafa Azizi 🆔 🅖 SC Ⓟ received a State Engineer degree in Automation and Industrial Computing from the Engineering School EMI of Rabat, Morocco in 1993, then a Master degree in Automation and Industrial Computing from the Faculty of Sciences of Oujda, Morocco in 1995, and a Ph.D. degree in Computer Science from the University of Montreal, Canada in 2001. He earned also tens of online certifications in Programming, Networking, AI, Computer Security. He is currently a Professor at the ESTO, University Mohammed First of Oujda. His research interests include Security and Networking, AI, Software Engineering, IoT, and Embedded Systems. His research findings with his team are published in over 100 peer-reviewed communications and papers. He also served as PC member and reviewer in several international conferences and journals.

Prof. Dr. Omar Moussaoui 🆔 🅖 SC Ⓟ is an Associate Professor at the Higher School of Technology (ESTO) of Mohammed First University, Oujda – Morocco. He has been a member of the Computer Science Department of ESTO since 2013. He is currently director of the MATSI research laboratory. Omar completed his Ph. D. in computer science at the University of Cergy-Pontoise France in 2006. His research interests lie in the fields of IoT, wireless networks and security. He has actively collaborated with researchers in several other computer science disciplines. He participated in several scientific & organizing committees of national and international conferences. He served as reviewer for numerous international journals. He has more than 20 publications in international journals and conferences and he has co-authored 2 book chapters. Omar is an instructor for CISCO Networking Academy on CCNA Routing & Switching and CCNA Security.