JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(2), JUNE, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

An Adversarial Examples against Deep Learning-Based Network Intrusion Detection System: A Review

Ahmad Abubakar Yunusa[1], Fatima Umar Zambuk[2], Badamasi Imam Ya'u[3], Hamza Hussaini[4], Isma'il Haruna[5].
[1]*Computer Science Department,*
Nigerian Army University Biu, Borno, Nigeria.
[2, 3, 4] *Department of Mathematical Sciences,*
Abubakar Tafawa Balewa University, Bauchi, Nigeria.
[5] *Federal College of Education Gombe, Nigeria.*

ABSTRACT
Security holds significant importance in our daily lives, as any compromise in confidentiality can pose a serious threat from criminals. Cybercriminals constantly seek ways to exploit vulnerabilities and gather data for their own gain, particularly exploiting the known vulnerability of deep learning algorithms to adversarial examples. To protect computers and networks from potential attacks by hackers, who may attempt to steal or manipulate sensitive information stored in databases, cyber security specialists and designers are required to develop various intrusion detection systems. In this research paper, we conduct a survey on existing studies that explore the efficacy of adversarial examples in the domain of network intrusion detection systems, with the goal of devising defensive measures. Our review reveals that multiple attack methods have been investigated, demonstrating their effectiveness in detecting intrusions, and highlighting the vulnerability of deep neural networks to such attack strategies.

ARTICLE INFO
Article History
Received: January, 2023
Received in revised form: February, 2023
Accepted: May, 2023
Published online: June, 2023

KEYWORDS
Machine Learning; Deep Neural Network; Network Intrusion Detection System; Adversarial Examples

INTRODUCTION

A network is a congregation of Computers, Servers, Mainframes, and network devices like switches, routers, network terminal access points, Peripherals, or Other devices to easily share data. Since the internet is an interrelationship of a few networks, it has been observed that it plays a significant role in our day-to-day activities through frequent use of the internet (Dua and Du, 2016). This gave cyber criminals the courage to devise a means of extracting data and implementing vulnerabilities by any means possible for their selfish consumption, Consequently, To safeguard networks and computers from hackers who might attack the systems and take or ruin medical, financial, or other sensitive information from databases, cyber security experts and designers must create a variety of intrusion detection systems (Dua and Du, 2016). With this, several ways of protecting these data were developed by security experts to save victims from those vulnerabilities, as there are a lot of machine learning algorithms such as Hidden Markov Models, Fuzzy Logic, (Jha et al., 2001) and Neural Networks (Kaur & Gill, 2013; Wu and Banzhaf, 2010) have made great achievements in an intrusion detection system. The majority of conventional machine learning techniques with shallow architectures either contain only one hidden layer (for example, ANN) or none at all (for example, Maximum Entropy Model) (Gao et al., 2015).

Intrusion Detection System (IDS) is an important research development in the area of information security that can identify an intrusion, it could be an intrusion that is happening today now or one that has already happened. Typically, intrusion detection relates to a classification issue, like a binary or multiclass classification algorithm., that determines if network traffic conditions are anomalous, normal, or a four-category classification, identifying whether it is normal or any one of Probe, Denial of Service (DOS), User to Root (U2R), Remote to Local (R2L), attack types. In short, improving a classifier's intrusion detection accuracy is the

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

major motivation for intrusion detection. (Yin et al., 2017).

A network or host system can use the IDS to quickly detect intrusions, attacks, or security policy breaches. It is essential for protecting against unauthorized access and malicious activity. Based on operational logic, there are two basic categories of IDS: Both signature-based which compares a database with identified vulnerabilities, and anomaly-based IDS that examines communications Using an activity pattern. (Wang, 2018).

Many intrusion detection systems make use of Machine Learning (ML) Techniques in deploying security measures to computerized information to provide good discretion on the information. Network Intrusion Detection System is the name of an IDS system that examines a packet of networks to find intrusions (NIDS). A NIDS is obtained by mirroring the aforementioned network devices. Deep learning is acknowledged as a suitable approach in NIDS, however recent research has shown that these models are susceptible to intentionally generated inputs known as adversarial examples (Szegedy et al., 2014).

Adversarial examples are created by introducing minor but purposefully chosen perturbations to the original data that caused the records to be inappropriately identified by the deep learning models. These Adversarial examples include the C&W attack (Carlini and Wagner, 2017), DeepFool (Moosavi-Dezfooli et al., 2016), Jacobian-based saliency map attack (JSMA) (Papernot et al., 2016) and Fast gradient sign method (FGSM) (Goodfellow et al., 2015).

Yang et al., (2019) developed a deep neural network (DNN) model for NIDS, an attempt was made to use adversarial samples to cause the model to misclassify. Three alternative attack strategies were examined in a black-box model to create adversary examples. These attack strategies e.g. C&W, ZOO, and GAN were used in launching adversarial attacks.

In this paper, we investigate and present some review on papers that reveals several attacks which had been examined and shown to be efficient at intrusions detection.

The remaining parts of this paper is structured as follows: Section II, a succinct introduction is given on machine learning algorithms and deep learning algorithms implored in the field of adversarial examples against Network Intrusion Detection, section III describes NIDS with adversarial examples as attack techniques, section IV explains some related work, and finally Section VII gives a detailed conclusion and the prospective research direction.

Machine Learning Classifiers

1. *Support Vector Machine (SVM)*

The Support Vector Machines (SVM) (Jin et al., 2021) come in handy in determining the separation boundaries in such situations. The supervised learning method known as SVM is applied to resolve binary classification and regression issues. Therefore, SVMs are linear binary classifiers, i.e., they devise a hyperplane border, in order to divide data points among two distinct groups. To divide these data into two classes, the classifier constructs an N-1-dimensional hyperplane for n-dimensional feature vectors (such as N=2, hyperplane is a line) (Aggarwal, 2018). This classifier line is expressed by the following Equation:

$$y = w.f(x) + b \quad (1)$$

Where $f(x)$ is the feature vector.
w is the weight allocated to the feature vector and
b is the bias.

2. *Decision Tree (DT)*

An artificial intelligence model called a decision tree is a machine learning model constructed on a sing series of options based on variable values to go in one direction or another. A distinct method to machine learning is represented theoretically and mathematically by decision trees. Using some training data, Decision Tree seeks to construct a model by analyzing the decision rules for the anticipated class or value of the objective variable (Aggarwal, 2018). For a limited number of decision items, it is simple and intuitive to manually design a decision tree. For a lot of information, the bagging technique and the rules drawn from the data can be used to create the decision tree.

3. *K-Nearest Neighbors (KNN)*

The K-Nearest Neighbors is a statistical technique which can be used for solving regression and classification problems. The idea of finding the K-nearest neighbors for a specific

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

data point is relatively straightforward. It is assumed that related products are located close to one another. Here, "close" refers to a distance between two places that may be measured as a straightforward Euclidean distance. (Aggarwal, 2018). You can determine that the test data belongs to a class by locating the largest group of things that are similar to the test data. Once a new data point is received for which the classification needs to be predicted, check the database for the K points that are closest to the new data point. The new data point is also given the class or group that its close neighbors belong to. (Aggarwal, 2018).

Deep Learning Classifiers

Deep learning is a branch of machine learning's discipline, which focused with developing algorithms that are inspired by the structure and operation of the brain called Artificial Neural Networks. (Brownlee, 2019). It belongs to a larger group of artificial neural network-based representation learning machine learning techniques. Learning can be unsupervised, semi-supervised, and supervised learning. Deep Neural Networks (DNN), Deep Belief Networks (DBN), Convolutional neural networks (CNN), Deep Reinforcement Learning (DRL), and Recurrent Neural Networks (RNN) are examples of deep-learning designs that have been used in a variety of fields including Network security, Natural language processing, biometrics, Computer vision, Speech recognition, machine translation, drug design, material inspection, board game programs, and medical image analysis. These designs yielded outcomes that were on par with, and in some cases even above, the performance of human experts. (Aggarwal, 2018).

Few hand-engineered features and specialist knowledge are needed for deep learning. The complexity of data can be retrieved from the raw input features with a higher and more abstract level representation due to the development of big data and hardware acceleration. Recent studies have demonstrated that deep learning-based IDS have impressive learning capabilities or exceed their conventional ML-based competitors (Wang, 2018).

1. Artificial neural networks (ANN)

Artificial neural networks (ANNs), sometimes called neural networks or simply ANNs, are structures in computers that are modeled after the organic neural networks seen in animal brains. Artificial neurons, which are a set of interconnected units or nodes that loosely resemble the neurons in a biological brain, are the foundation of an ANN. (Aggarwal, 2018). Multiple computational layers are common in neural networks given that the computations performed there are hidden from the user, the additional intermediate layer(s) between input and output are referred to as hidden layers (Nguyen et al., 2019).

Consider each node to be a separate linear regression model, with weights, input data, an output data and a bias. The equations are expressed as follows:

$$\sum w_i x_i + \text{bias} = w_1x_1 + w_2x_2 + w_3x_3 + \text{bias} \quad (2)$$

$$\text{Output} = f(x) = 1 \text{ if } \sum w_1x_1 + b >= 0; 0 \text{ if } \sum w_1x_1 + b < 0 \quad (3)$$

Where w is the assigned weight.
x is the input.

Following the identification of the input layer, weights are assigned. With heavier weights having a greater impact on the outcome than lighter ones do on other inputs, these weights help determine the importance of each variable.

Subsequently, before combining, each input is multiplied by the corresponding weight. Once the output has passed through it, an activation function will then decide what the output will be. This "fires" (or activates) the node, sending data to the network's next layer, if the output rises beyond a predetermined threshold. Consequently, the input of one node becomes the output of the following node. (Jin et al., 2021). This neural network is a feedforward network since data is passed from one layer to the next layer throughout this procedure. A neural network is referred to as shallow when it has one or two hidden layers and deep when its hidden layers count above two.

2. Convolutional neural network (CNN)

CNNs, or convolutional neural networks, are constructed using the convolution technique. Instead of the usual two-dimensional array, CNN has a three-dimensional configuration of neurons (Aggarwal, 2018). Convolutional layer is the name of the top

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

layer where each neuron in the convolutional layer only processes a small piece of the visual field. The input features are extracted in batches, much like a filter. The network can calculate these actions repeatedly to complete the entire image processing since it understands the images in their component components. Processing entails converting an image from RGB or HSI scale to greyscale. Additionally, adjustments to the pixel value can be made to help detect edges and categorize the images.

The propagation is unidirectional when a CNN has one or more convolutional layers followed by pooling. When it is bidirectional, a fully connected neural network receives the output of the convolution layer and uses it to classify images. The image can be extracted in part using filters. In MLP the weighted sum of the inputs is supplied to the activation function. Convolution uses RELU activation function While MLP employs a nonlinear activation function followed by softmax. Convolution neural networks provide highly successful outcomes in the recognition of images and videos, semantic parsing, and paraphrase detection (Jin et al., 2021).

3. Recurrent neural network (RNN)

Recurrent neural networks (RNNs) models are neural networks which are most effective at handling sequential data. With time series data or just a sequence of data with repeated qualities, a series or sequence of data can be seen in several issue domains (Jin et al., 2021). By having the capacity to recall previous data and analyze the current vector in light of inputs that are either before or after the current vector in the sequence, RNN models are able to address the challenge of sequence modeling.

A typical RNN architecture consists of multiple neural networks, with processed data from the prior series being sent into each network along with sequential inputs. In order to utilize the contextual information that is included in the sequence data, RNN is based on a straightforward neural network. Each input in the sequence, automatically, generates a memory state that is derived from the current and previous inputs (Aggarwal, 2018). The result of the next sequence is computed using this memory and the subsequent input vector. RNN evaluates the network's state for each input in a sequence and transmits that state to the computation of its resulting output. The output of sequence t is represented in Equations. 4 and 5 below:

$$a^{<t>} = g(W_a a^{<t-1>} + W_x X^{<t>} + b_a) \quad (4)$$

$$Y'^{<t>} = g(W_y a^{<t>} + b_y) \quad (5)$$

Where a is computed activation input from sequence t
X is input at sequence t
Y' is output computed at sequence t
W_x is the weight vector for an input vector
W_y is the weight vector for the output vector
W_a is the weight vector for the activation input
g is the activation function: tanh for the activation input and Sigmoid for the output

4. Long short-term memory (LSTM)

LSTM networks are a type of RNN that uses special units in addition to standard units (Shende, 2020). A 'memory cell' found in LSTM units is capable of storing data in memory for a longer period of time. Information entry into the memory, output, and forgetting are all governed by a set of gates (Aggarwal, 2018). There are three distinct categories of gates: the input gate, the output gate, and the forget gate. How much information from the previous sample will be stored in memory is determined by the input gate; the output gate controls how much data is sent to the following layer; while forget gates manage the rate at which memory storage is torn. They can discover longer-term dependencies according to this architecture (Laghrissi et al., 2021).

5. Deep beliefs network (DBN)

DBNs are a representation of a generative model. We specify the input activations in a feedforward network, and they influence the responses of the feature neurons eventually in the network (Aggarwal, 2018). Similar techniques can be employed with generative models like DBNs, but it is also possible to "run the network backward" to produce values for the input activations by specifying the values of some of the feature neurons (Gao et al., 2015). To put it more specifically, a DBN trained on images of handwritten numbers can (perhaps, and with some caution) also be used to create images that resemble handwritten numbers. In other words,

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

the DBN would be writing in some sense. A generative model is similar to the human brain in this regard because it can both read and write numbers. Unsupervised and semi-supervised learning can be implemented with DBN. For example, even if the training images are labelled, DBNs can learn useful features for comprehending other images when trained using image data (Gao et al., 2015). Additionally, the ability to perform unsupervised learning is quite intriguing from a fundamental scientific standpoint as well as potentially having practical uses.

6. *Multi-layer perceptron (MLP)*

An entrance point into complicated neural networks that process incoming data across numerous artificial neural layers. A fully connected neural network is one in which every node is linked to every neuron in the layer above. There are several hidden layers present in both the input and output layers, for a total of at least three layers (Jin et al., 2021). It propagates in both directions, either forward propagation or backward propagation.

The inputs are multiplied by weights and provided to the activation function where they transform in backpropagation to decrease the loss. Weights are just machine-clear values in neural networks (Aggarwal, 2018). Depending on the variation between training inputs and presented outputs, they self-adjust. Softmax is used as the output layer activation function with nonlinear activation functions.

METHODOLOGY

This section contains the methodology of this research work summarized as follows:

Performing a Keyword-based Search

Some most recognized academic research databases which include Web of Science, Research gate, IEEE Xplore, Sciencedirect and Google scholar were used in the keyword-based Searching of papers related to the topic. These focused keywords include: Intrusion Detection, Big Data-based Intrusion Detection System, Intrusion Detection System using big data and deep learning, and Adversarial examples for Intrusion Detection System Using deep learning. The word "artificial" was not included in the search term because of the fact that in the articles NN have been discussed by both "neural network(s)" and "artificial neural network(s)" phrase, therefore accidental excluding any important articles was avoided. In particular, taking into account that the word "Artificial Neural Network" is often employed to refer to a specific architecture, like a multi-layer perceptron.

Searching Relevant Cited Papers

Intrusion Detection, Adversarial Intrusion Detection System and Deep Learning papers were cited. Papers in the references of this research paper were also investigated and cited.

Reviewing the Related Papers

Some of the Cited papers collected were reviewed in due cause by focusing and identifying the Author(s), identifying the Machine Learning or Deep learning Model(s) proposed by the Author(s), stating the purpose for of the proposed model(s), identifying the source of data collection, summarized the result and compared the performance of the model(s) with other mode(s) (if any) and finally stated the limitation of the proposed model.

Introducing the Prediction Models

This study emphasizes on machine learning models especially deep learning models applied in network intrusion detection. The application and development of these models were discussed in section II.

A. Network Intrusion Detection System (NIDS)

Intrusion Detection Systems (IDS) are computer applications that are designed to detect abnormal or harmful initiatives for breaching a computer network (Lakshminarayana et al., 2019). They are a crucial strategy in any attempt to address cyber security issues. An intrusion detection system (IDS) enables one to discover whether a network is being attacked (Alsaeedi and Khan, 2019). Once an alert is received that a network is attacked, an action to quickly prevent that attack from taking over the system is taken. Machine Learning and Deep Learning methodologies, which include supervised, unsupervised -and semi-supervised are employed in order to facilitate these intrusion detection systems (Heng and Weise, 2019). Gao et al., (2015) proposed the use Deep Belief Networks model to develop

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

an intrusion detection system, (khorram, 2021) built an IDS model based on the most effective Machine learning algorithms i.e. KNN, SVM, and RF algorithms.

A NIDS is a system that identifies harmful behavior in a network (Chandran and Kumar, 2018). Typical network harmful actions include probing, denial of service (DoS), remote to local (R2L), user to root (U2R), etc are common malicious network actions (Yang et al., 2019).

B. Adversarial Examples

Almost every element of our everyday life is now impacted by an information technology infrastructure. However, user data and Critical services are now more vulnerable to cyber-attacks due to this level of flawless connection (Goodfellow et al., 2015). Adversarial examples are tailored data intended to confuse a neural network and cause it to incorrectly categorize a certain input (Szegedy et al., 2014). These infamous inputs are undetectable to the human eye, but they prevent the network from correctly identifying the image's contents. Researchers and security professionals alike must develop unique knowledge, in-depth understanding, and the appropriate abilities to address these issues as threats become more complex and prevalent (Heng & Weise, 2019). Hence, Threat modeling, Attack simulation, counter major design noise detection (for evasion-based attacks), and Information laundering were the approaches to protecting machine learning introduced by researchers. Application scenarios of these Adversarial attack strategies are either black-box attacks or white-box attacks. Attackers using white-box techniques have access to the classifier's training data, model parameters, and other crucial details. While the attacker in a black-box scenario has little to no knowledge about the classifier and is therefore severely constrained (Martins et al., 2020).

There are several types of such attacks strategies, which include: the Limited-memory Broyden Fletcher Goldfarb Shanno (L-BFGS), the Fast Gradient Sign Method (FGSM), the Jacobian-based Saliency Map Attack (JSMA), Carlini and Wagner (C&W), Deepfool, Zeroth Order Optimization (ZOO), the Generative adversarial networks (GAN), etc.

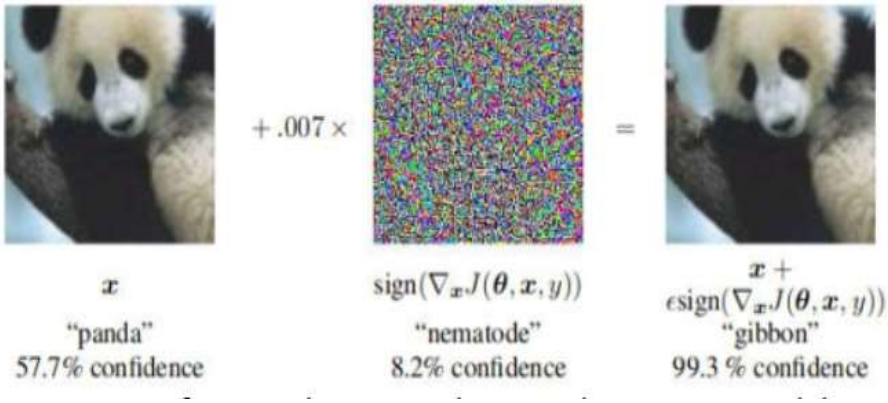

Figure 1: Demonstration of an Adversarial sample generated by applying FGSM to GoogleNet. Left: Clean Image of Panda, Middle: Perturbation, Right: Adversarial Image classified as Gibbon. (Ren et al., 2020)

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(2), JUNE, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

1) Limited-memory Broyden Fletcher Goldfarb Shanno (L-BFGS)

(Szegedy et al., 2014) proposed a generic targeted issue employing L-BFGS optimization, and one of the earliest gradient approaches for generating adversarial samples in the field of imaging.

Given an original image, searches for a distinct image that is identical to the input image but has been incorrectly classified by the classifier using L_2 distance. In order to reduce the amount of perturbations r introduced to the original image under L_2 distance, the task is described as an optimization problem via the addition of noise to the initial image expressed in the equation below:

$$min\ ||r||^2\ subject\ to: f(x + r) = l \quad (6)$$

Where x represents the original input image.
r represents the perturbation.
f represents the classifier's loss function.
l represents the faulty prediction/label.

Despite the fact that this method can generate adversarial example, it is not very practical because it employs a computationally expensive algorithm to identify the ideal response (Martins et al., 2020).

2) Fast Gradient Sign Method (FGSM)

Although the L-BGSM attack uses an inefficient and time-consuming linear search approach to determine the optimal value, Goodfellow et al., (2015) suggested a simple and fast gradient-based strategy (FGSM) technique for generating adversarial examples with the target of decreasing the optimum level of perturbation (L_{inf} distance metric) applied on any image pixel to create misclassification.

At each pixel, they just did a one-step gradient update in the direction of the sign of gradient. Their perturbation can be represented as follows:

$$\eta = \epsilon * sign(\nabla x\ J(x, l)) \quad (7)$$

Where η is the perturbed sample.
ϵ is a hyper-parameter determining the amount of perturbation added to each feature.
J is the cost function.
x is the original input.
l is the target label.

This technique is one of the fastest in terms of computing time by allowing adversarial examples to be generated quickly.

3) Jacobian-based Saliency Map Attack (JSMA)

Mcdaniel, et al., (2016) propose the JSMA, a successful target attack capable of fooling DNNs with small L_0 perturbations. Before applying the softmax layer, the approach computes the Jacobian matrix of the logit outputs $l(x)$.

An input sample's saliency maps, which show the saliency values of each feature, are computed using this method. These numbers show how much each feature's modification affects process of classification.

This features are then chosen in decreasing order of saliency value, with each feature being affected by the value Θ. When a misclassification takes place or a certain number of updated features is reached, the process ends (Martins et al., 2020).

This method uses feature selection to reduce the number of features modified (L_0 distance metric) while increasing misclassification.

4) Carlini and Wagner (C&W)

C&W offer a series of adversarial techniques based on optimization that can produce L_0, L_2, L_∞, as well as the norm-measured adversarial examples, namely CW_0, CW_2, and CW_∞. It develops the optimization target in the same way as L-BFGS (Ren et al., 2020).

Unlike the L-BFGS, which uses the identical loss function, the authors suggested using various loss functions, such as hinge loss rather than cross-entropy lossIn order to remove the box limitations and make the minimization issues easier to solve, they also presented a new variant w. (Martins et al., 2020). This strategy shows that State-of-the-art defenses, that is adversarial training and defensive distillation can be outperformed more effectively than the L-BFGS attack strategy.

5) Deepfool

DeepFool is a new attack strategy that finds the smallest L_2 adversarial effects in both an affine binary and generic binary differentiable classifiers which was proposed by (Moosavi-Dezfooli et al., 2016). In order to construct the attack, one must first determine an analytically linear decision boundary that divides samples into

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

different classes, and then one must add a perturbation perpendicular to the decision boundary.

6) Zeroth Order Optimization (ZOO)

Many signal processing and machine learning applications use zeroth-order Optimization (ZOO). ZOO is a subset of gradient-free optimization, Therefore the approach is contrary from the gradient-based adversarial generating method (Chen et al., 2017). It's similar to gradient-based methods in that it's utilized to solve optimization problems. It does not, however, require the gradient and instead relies solely on function evaluations. It may be used in a black-box attack without the need to transfer models. The three major processes of ZO optimization which are performed iteratively are:

1. Gradient estimation
2. Descent direction computation, and
3. Solution update.

Access to the victim deep learning models is not required by the ZOO because of the gradient estimation of Hessian and gradient. However, querying and estimating the gradients need time-consuming calculations. Experiments revealed that ZOO's performance was comparable to C&W's Attack.

7) Generative adversarial networks (GAN) attack technique

In June 2014, Ian Goodfellow and his colleagues developed a family of machine learning architectures called Generative Adversarial Networks (GAN). This technique gains the ability to produce new data with similar statistics as the data used for training. For instance, a GAN built on images can produce fresh images with numerous realistic features that, to human viewers, appear to be at least apparently authentic (Goodfellow et al., 2014). Generator G and Discriminator D are the two networks that make up the GAN. The generator G receives in random noise vector Z and parameter vector $theta_g$ to produce realistic data with a distribution that resembles that of real data to trick the discriminator into treating them as real data. The discriminator, on the other hand, seeks to accurately identify whether input data originate from the generator or genuine data. The generated $G(Z;\theta_g)$ networks are then utilized in order distribute false data using data distribution F_g (Kuppa et al., 2021).

Dataset Processing Stage

There were only a few datasets which are applied in intrusion detection scenarios, among which were used in the reviewed papers, where NSL-KDD was used in six research, CICIDS2017 in two studies, and CTU-13 in three studies.

Table 1: Summary of Datasets used in intrusion detections

Dataset	Status in this review	Description
DARPA1998	Not Used	Intrusion Detection Evaluation dataset with two parts: an offline evaluation and a real-time evaluation.
KDD CUP 99	Not Used	Intrusion detection dataset with Dos, Probe, R2U, and U2R attack types.
NSL-KDD	Used	An extension of the KDD CUP1999 dataset with the same attack types.
CISC2010	Not Used	Is an artificially created HTTP with thousands of web requests.
ISCX2012	Not Used	Comprises of identified network traces that also include whole packet payloads in pcap format.
CICIDS2017	Used	Intrusion detection dataset with numerous types of attacks including Dos, XSS, SQL Injection, Heartbleed, Bonet, etc.
CTU-13	Used	The CTU-13 dataset comprises thirteen records (scenarios) of

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

	various botnet samples. A distinct piece of malware that utilized many protocols and carried out various tasks is run in every scenario.

REVIEW OF RELATED WORKS

Some relevant papers were acquired and reviewed in order to: identify the Author(s), reveal the name of the learning model(s) the Author(s) suggested, and state the reasons for the model's proposal, state the purpose of each component that comprises the model, summarize the outcomes of the model's performance evaluation, and state the limitations of the proposed models. The comparative analysis done between any of the models and other models in the research paper was reviewed and the Results acquired were specified.

Rigaki and Elragal, (2017) evaluated the impact of adversarial examples in a scenario involving intrusion detection. The authors used NSL-KDD dataset for testing, generating Targeted attacks with FGSM and JSMA. Random forest, decision tree, linear SVM, multi-layer perceptron (MLP), neural network, and voting ensembles of the earlier three classification models to perform classification. The results showed that linear SVM is the most classifier with a decrease of 27% on JSMA. Random forest was the most reliable classifier, with reduced performance drops across all parameters, including an 18% drop in accuracy and a drop of 6% F1 score and AUC. Also, it was highlighted that while the investigated attacks verified their effectiveness against unidentified classifiers, the success of the cyber-attacks under different conditions has not been shown, and they still required an understanding of the data preprocessing, such as the classification features.

Wang, (2018) used the Multi-Layer Perceptron classifier in this research to assess how well modern attack methods perform using the NSL-KDD dataset for intrusion detection based on a deep learning algorithm. The performance of the MLP neural network was Evaluated, Attacks by CW tend to be less harmful than those by JSMA., FGSM, and Deepfool attacks had an AUC of 0.80, whereas the targeting FGSM had an AUC of 0.44 and was the most successful attack, In terms of usability and pooling at the initial stage, JSMA attacks are considerably more alluring for an adversary, generating an AUC of about 0.5 across all classes, however, because the size and parameters are increased by the LSTM cell, LSTM-CNN takes longer to train. Although the DNN is rather fast, it has low accuracy and specificity scores. Also, the authors examined various attacks on the improved features, discovering that almost all attacks use the same seven features, as well as describing how an attacker might physically edit each one to undertake adversarial attacks in a realistic scenario. The attacker is conscious of the target's deep neural networks, and white-box attacks were taken into consideration.

Lin et al., (2018) Adversarial techniques that can fool and defeat the intrusion detection system were generated using a variant of a WGAN dubbed IDSGAN on an NSL-KDD dataset. Adversarial network traffic using a generator G was created to construct adversarial attack traffic on real harmful traffic and a discriminator D to distinguish between the two. The classifiers tested for the effectiveness of the threats were neural network, MLP, naive Bayes, SVM, decision tree, k-nearest neighbors, logistic regression, and random forest. IDSGAN takes advantage of a generator to turn actual harmful data into adversarial malicious activity and a discriminator to identify examples of legitimate traffic from unwanted traffic. Adversarial attack examples carry out black-box attacks over the detection system. The findings demonstrated a reduction in adversarial example identification rates across all classifiers. The authors concluded that the suggested technique works well at producing adversarial attacks by producing fresh malicious samples. More research can propose to test this strategy on more attack types.

Yan et al., (2019) Introduced DoS-WGAN, a typical structure that takes utilization of Wasserstein generative adversarial networks (WGAN) and gradient penalty technology in order to disguise hazardous denial of service (DoS) attacks as the regular network traffic to get against network traffic classifiers, which the NSL-KDD uses. Four categories have been developed by the authors to group the features: i) Basic features which refer to the connection's most fundamental characteristics ii) the traffic features across a two-

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

second timeframe are the resulting output of all interactions made in the past two seconds makeup iii) Content features, which correspond to the connection's content and are unimportant for DoS attacks; iv) Statistics for the 100 prior connections are shown in the traffic features more than 100 connections window. To train WGAN, adversarial DoS samples were created and approximated to the distribution of ordinary traffic inputs, between the generator and the discriminator, to prevent the alteration of the traffic features and content features of the 100 connection units, a converter was introduced. The true positive rate of a CNN classifier was reduced, demonstrating the impact of the attack can be increased while keeping its integrity by not altering the denial of service attack's functioning features. Conclusively, DoS-WGAN is more stable and more efficient than other methods for creating and producing samples in opposition to a CNN-based IDS.

Yang et al., (2019) investigate how adversarial instances affect the effectiveness of the deep neural network (DNN) developed to identify anomalous activities within black-box adversarial methods. The DNN classifier was trained to detect NIDS with three attack techniques on the NSL-KDD dataset, these attack strategies were C&W, GAN, and Zoo. In the beginning, to attack a different trained model using C&W, another model was built and utilized to create adversarial samples. In the second instance, without first training a replacement, attack the target classifier, by querying the DNN classifier, ZOO was utilized to calculate the gradient and generate adversarial attacks. In generating adversarial samples, WGAN was trained. In the third scenario. Attack based on ZOO outperforms the Substitute model the most effective but least realistic strategy is GAN because it took too many queries to get adversarial samples that would be seen in a real setting.

Martins et al., (2019) evaluated the effect of JSMA FGSM, C&W, and Deepfool adversarial attacks on the CICIDS2017 and NSL-KDD datasets. Existing malicious samples were modified to analyze the footprint of multiple classifiers which include: random forest, decision tree, naive Bayes, SVM, DAE, and neural network. The DAE was utilized to classify images by initially training the autoencoder, preserving the layers, and then training more layers related to the autoencoder's output, which carried out classification. The performance of the classifiers was shown to be deteriorated by all classifiers employing an average AUC, the NSL-KDD dataset and the CICIDS dataset saw decreases of 13% and 40%, respectively. The DAE was found to be the most resilient classifier. Adversarial training was then applied to all classifiers as a defensive measure, it was shown that all classifiers performed better, with the random forest on average outperforming the others on the two datasets.

Apruzzese et al., (2019) carried out attacks on the network intrusion detectors' integrity. On the CTU-13 dataset, MLP neural networks and KNN models were used. Having divided the detectors into numerous instances, each dedicated to a different botnet family, Random forest was discovered to be the best-performing detector, whereas KNN was the worst. A custom adversarial attack was then implemented targeting three attributes: *exchanged_bytes* (amount of data shared), *duration* (time spent connecting), and *total_packets* (amount of packets exchanged. This approach operates by adding random values to each feature over a predetermined interval, resulting in new adversarial samples. Random forest is found to be the best-performing classifier, whereas MLP neural network is the worst. Adversarial training and feature removal defenses were then tested where the recall was increased for both Random forest and KNN While applying feature removal, the recall improved for both MLP and random forest when using adversarial training.

Apruzzese and Colajanni, (2018) by changing network flows alone utilizing the CTU13 dataset, the authors investigated the efficacy of adversarial examples performed in a situation when an attacker attempts to covertly spread botnet software throughout a large network while remaining undiscovered. A random forest detector trained on active botnets utilizing network flow parameters like connection protocol, IP addresses, ports, and packets sent was used on the dataset. The dataset contains 7 different varieties of malware. 7 classifiers (1 for each attack type) used training sets with 95% normal and 5% harmful samples for training. With two exceptions, the detection rate was generally greater than 0.95; one of these should be highlighted because of how little sample data was provided for the Sugou attack,

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

which was one of the outliers. The number of packets, period of the flow, bytes exchanged, and up to four other parameters of the original malicious flows are all randomly altered. The detection rates for each type of attack were drastically reduced, according to the number of variables altered and the amount of the modification, decreasing detection performance.

Huang et al., (2020) evaluated the efficiency of adversarial samples by generating DDoS adversarial samples using two methods: Genetic Attack (GA) and Probability Weighted Packet Saliency Attack (PWPSA) on the CICIDS2017 dataset. Both techniques modify the initial data samples by adding or removing missing packets. In GA, the authors use the genetic algorithm to create a group of modified samples and then find the best variation among them. In PWPSA, the authors iteratively change the original sample, using position saliency and at each stage, the insertion or replacement order is determined by the packet score. The Experiments demonstrate that both strategies have a high success rate in bypassing DDoS detectors therefore both GA and PWPSA methods are powerful and effective.

Table 2: Summary of Related Works by Proposed Methods and Data

S/N	Reference	Title	Classifiers used	Datasets	Attack Algorithms	Results	Limitations
1	Rigaki and Elragal, (2017)	Adversarial deep learning against intrusion detection classifiers	Decision Tree, Random forest, SVM, Voting ensemble.	NSL-KDD	JSMA	The most enduring is a random forest (dropped with 10% accuracy) SVM is the most affected classifier (dropped with 27% accuracy)	The effectiveness of attacks under different conditions and classification features weren't put to the test.
2	Wang, (2018)	Deep Learning-Based Intrusion Detection with Adversaries	MLP	NSL-KDD	JSMA, FGSM, Deepfool, C&W	CW attacks appear to be less damaging than JSMA, FGSM, and Deepfool attacks, JSMA attacks are more intriguing to an adversary in terms of use and application.	White-box attacks were considered as the target deep neural networks known by the adversary. So, black box attacks are not tested
3	Lin et al., (2018)	IDSGAN: Generative Adversarial Networks for Attack Generation	Neural Network	NSL-KDD	WGAN	All classifiers' detection rates decrease.	The performance of the classifiers was not improved by

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

S/N	Reference	Title	Classifiers used	Datasets	Attack Algorithms	Results	Limitations
		against Intrusion Detection					a variety of changeable features.
4	Yan et al., (2019)	Automatically synthesizing DoS attack traces using generative adversarial networks	CNN	NSL-KDD	WGAN	True Positive Rate Reduced	Only the content and traffic features of the 100 connections windows were altered by the authors.
5	Yang et al., (2019)	Adversarial Examples Against the Deep Learning Base-Network Intrusion Detection System	Naive Bayes Random forest SVM DNN	NSL-KDD	C&W ZOO GAN	Under the black-box attack, the target DNN model's accuracy, precision, recall, and All Fscores have been greatly decreased. ZOO-based attacks outperform GAN and substitute model attacks in terms of effectiveness.	Attackers cannot access the IDS's classifier (black-box attacks).
6	Martins et al., (2019)	Investigate the behavior of different state-of-the-art machine learning algorithms in an adversarial scenario	Random forest Decision tree Naive Bayes Neural Network RBF SVM Denoising Autoencoder	NSL-KDD, CICIDS2017	FGSM JSMA Deepfool C&W	Successful adversarial attacks cause all classifiers' performance to decline by 13% to 40%. The approach with the most tolerance to attack is Denoising Autoencoder.	Perturbations on lower-level features such as packet level that would have enabled generation of adversarial DoS attacks on real scenarios were not addressed.

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

S/N	Reference	Title	Classifiers used	Datasets	Attack Algorithms	Results	Limitations
						A random forest is the classifier that is the most resilient.	
7	Apruzzese et al., (2019)	Analyzing the Footprint of Classifiers in Adversarial Denial of Service Contexts	Random forest MLP KNN	CTU-13	Random noise added to selected features	KNN is the least effective classifier, whereas Random Forest is the best. Random forest is the best classifier under attack and MLP NN is the worst.	Platforms for cyber security can incorporate more powerful machine learning-based techniques.
8	Apruzzese and Colajanni, (2018)	Evading Botnet Detectors Based on Flaws and Random Forest with Adversarial Samples	Random forest	CTU-13	Random noise added to selected features	Detection rates reduced proportionally to the number of features attacked and the magnitude of perturbations	Future techniques aimed at improving the performance of network intrusion detection systems based on machine learning can further be implemented.
9	Huang et al., (2020)	Adversarial Attack against LSTM-based DDoS Intrusion Detection System	LSTM	CICIDS2017	GA PWPSA	Both GA and PWPSA methods are powerful and effective.	more reliable LSTM-based DDoS detector needs to be created.

CONCLUSION

In this paper, we reviewed several studies that used adversarial machine learning concepts for intrusion detection. We started by demonstrating many essential principles that can aid in learning the basic principles of machine learning, as well as adversarial attack methods.

We then looked at how these strategies could be used in intrusion detection scenarios and came to the following conclusions:

1. There have been studies on a number of adversarial attack strategies, with some being more useful than others depending on the circumstance. T Most attacks originate by employing genetic algorithms to improve

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

adversarial samples or by taking use of a neural network's gradient to find vulnerabilities in classifiers.

2. On normal data, all classifiers produced similar results, however, when subjected to adversarial attacks, decision trees, linear SVMs and naive Bayes, have been most negatively impacted, whereas random forests, RBF SVMs, and neural networks with attacker-specific architectures were the most enduring.

Recommendations for future research are concluded as follows: attacking time-based intrusion detection systems in real-time, such as classifiers based on recurrent neural networks, because all attacks and detections were carried out on separate samples.

Therefore, the creation of those adversarial attacks is required in the applications of deep learning models such as long short-term classifier for intrusion detection. Given that LSTM networks are effective at managing time series issues, it is necessary to train them for network intrusion detection in order to detect state-of-the-art attack techniques. Some techniques that have been proposed to deceive feed-forward neural networks, such as FGSM, JSMA, and C&W, can also be applied to attack LSTM models, an approach that has seen some recent attempts.

REFERENCES

Aggarwal, C. C. (2018). Neural Networks and Deep Learning. In *Neural Networks and Deep Learning*. https://doi.org/10.1007/978-3-319-94463-0

Alsaeedi, A., & Khan, M. (2019). Performance Analysis of Network Intrusion Detection System using Machine Learning. *International Journal of Advanced Computer Science and Applications*, *10*, 671–678. https://doi.org/10.14569/IJACSA.2019.0101286

Apruzzese, G., & Colajanni, M. (2018). *Evading Botnet Detectors Based on Flows and Random Forest with Adversarial Samples*. https://doi.org/10.1109/NCA.2018.8548327

Apruzzese, G., Colajanni, M., Ferretti, L., & Marchetti, M. (2019). *Addressing Adversarial Attacks Against Security Systems Based on Machine Learning*. https://doi.org/10.23919/CYCON.2019.8756865

Bourou, S., El Saer, A., Velivasaki, T., Voulkidis, A., & Zahariadis, T. (2021). A Review of Tabular Data Synthesis Using GANs on an IDS Dataset. *Information*, *12*, 375. https://doi.org/10.3390/info12090375

Carlini, N., & Wagner, D. (2017). *Towards Evaluating the Robustness of Neural Networks*. https://doi.org/10.1109/SP.2017.49

Chandran, S., & Kumar, K. (2018). A survey of intrusion detection techniques. *International Journal of Engineering & Technology*, *7*, 187. https://doi.org/10.14419/ijet.v7i2.4.13036

Dua, S., & Du, X. (2016). Data Mining and Machine Learning in Cybersecurity. *Data Mining and Machine Learning in Cybersecurity*. https://doi.org/10.1201/b10867

Gao, N., Gao, L., Gao, Q., & Wang, H. (2015). An Intrusion Detection Model Based on Deep Belief Networks. *Proceedings - 2014 2nd International Conference on Advanced Cloud and Big Data, CBD 2014*, 247–252. https://doi.org/10.1109/CBD.2014.41

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, *November*.

Heng, L., & Weise, T. (2019). Intrusion Detection System Using Convolutional Neuronal Networks: A Cognitive Computing Approach for Anomaly Detection based on Deep Learning. *Proceedings of 2019 IEEE 18th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2019*, 34–40. https://doi.org/10.1109/ICCICC46617.2019.9146088

Huang, W., Peng, X., Shi, Z., & Ma, Y. (2020). Adversarial Attack against LSTM-based DDoS Intrusion Detection System. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2020-Novem*, 686–693.

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

https://doi.org/10.1109/ICTAI50040.2020.00110

Jha, S., Tan, K., & Maxion, R. A. (2001). Markov chains, classifiers, and intrusion detection. *Proceedings of the Computer Security Foundations Workshop*, 206–219. https://doi.org/10.1109/CSFW.2001.930147

Jin, Y., Wang, H., & Sun, C. (2021). Introduction to Machine Learning. In *Studies in Computational Intelligence* (Vol. 975). https://doi.org/10.1007/978-3-030-74640-7_4

Kaur, H., & Gill, N. (2013). Host-base Anomaly Detection using Fuzzy Genetic Approach (FGA). *International Journal of Computer Applications*, *74*(20), 5–9. https://doi.org/10.5120/13024-0026

KHORRAM, T. (2021). Network Intrusion Detection using Optimized Machine Learning Algorithms. *European Journal of Science and Technology*, *25*, 463–474. https://doi.org/10.31590/ejosat.849723

Kolen, J. F., & Kremer, S. C. (2010). Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies. In *A Field Guide to Dynamical Recurrent Networks*. https://doi.org/10.1109/9780470544037.ch14

Kuppa, A., Aouad, L., & Le-Khac, N. A. (2021). Towards Improving Privacy of Synthetic DataSets. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12703 LNCS*(May), 106–119. https://doi.org/10.1007/978-3-030-76663-4_6

Laghrissi, F. E., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00448-4

Lakshminarayana, D., Philips, J., & Tabrizi, N. (2019). *A Survey of Intrusion Detection Techniques*. https://doi.org/10.1109/ICMLA.2019.00187

Lin, Z., Shi, Y., & Xue, Z. (2018). *IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection*. 1–16. http://arxiv.org/abs/1809.02077

Martins, N., Cruz, J., Cruz, T., & Henriques Abreu, P. (2020). Adversarial Machine Learning Applied to Intrusion and Malware Scenarios: A Systematic Review. *IEEE Access*, *PP*, 1. https://doi.org/10.1109/ACCESS.2020.2974752

Martins, N., Cruz, J. M., Cruz, T., & Abreu, P. H. (2019). Analyzing the Footprint of Classifiers in Adversarial Denial of Service Contexts. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11805 LNAI*(July 2021), 256–267. https://doi.org/10.1007/978-3-030-30244-3_22

Moosavi-Dezfooli, S. M., Fawzi, A., & Frossard, P. (2016). DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 2574–2582. https://doi.org/10.1109/CVPR.2016.282

Nguyen, G., Dlugolinsky, S., Bobak, M., Tran, V., Lopez Garcia, A., Heredia, I., Malík, P., & Hluchý, L. (2019). Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, *52*, 77–124. https://doi.org/10.1007/s10462-018-09679-z

Panigrahi, C. R. (2018). *Advanced Computing and Intelligent Engineering* (Vol. 1).

Papernot, N., Mcdaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. *Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016*, 372–387. https://doi.org/10.1109/EuroSP.2016.36

Papernot, N., McDaniel, P., Swami, A., & Harang, R. (2016). *Crafting adversarial input sequences for recurrent neural networks*. https://doi.org/10.1109/MILCOM.2016.7795300

Papernot2016.pdf. (n.d.).

Ren, K., Zheng, T., Qin, Z., & Liu, X. (2020). Adversarial Attacks and Defenses in Deep Learning. *Engineering*, *6*. https://doi.org/10.1016/j.eng.2019.12.012

Rigaki, M., & Elragal, A. (2017). Adversarial Deep

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.

Learning Against Intrusion Detection Classifiers. *CEUR Workshop Proceedings*, *2057*(December), 35–48.

Shende, S. (2020). Long Short-Term Memory (LSTM) Deep Learning Method for Intrusion Detection in Network Security. *International Journal of Engineering Research And*, *V9*. https://doi.org/10.17577/IJERTV9IS061016

Sun, M., Tang, F., Yi, J., Wang, F., & Zhou, J. (2018). Identify susceptible locations in medical records via adversarial attacks on deep predictive models. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *April*, 793–801. https://doi.org/10.1145/3219819.3219909

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, *April 2014*.

Wang, W., Tang, B., Wang, R., Wang, L., & Ye, A. (2019). *A survey on Adversarial Attacks and Defenses in Text*.

Wang, Z. (2018). Deep Learning-Based Intrusion Detection with Adversaries. *IEEE Access*, *6*, 38367–38384. https://doi.org/10.1109/ACCESS.2018.2854599

Wu, S. X., & Banzhaf, W. (2010). The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing Journal*, *10*(1), 1–35. https://doi.org/10.1016/j.asoc.2009.06.019

Yan, Q., Wang, M., Huang, W., Luo, X., & Yu, F. R. (2019). Automatically synthesizing DoS attack traces using generative adversarial networks. *International Journal of Machine Learning and Cybernetics*, *10*(12), 3387–3396. https://doi.org/10.1007/s13042-019-00925-6

Yang, K., Liu, J., Zhang, C., & Fang, Y. (2019). Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems. *Proceedings - IEEE Military Communications Conference MILCOM*, *2019-Octob*, 559–564. https://doi.org/10.1109/MILCOM.2018.8599759

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

Zhang, W. E., Sheng, Q., Alhazmi, A., & Li, C. (2020). Adversarial Attacks on Deep-learning Models in Natural Language Processing: A Survey. *ACM Transactions on Intelligent Systems and Technology*, *11*, 1–41. https://doi.org/10.1145/3374217

Corresponding author: Ahmad Abubakar Yunusa
aameerahmad22@gmail.com
Computer Science Department, Army University, Biu, Borno State.