International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

Performance Evaluation and Validation of Intelligent Security Mechanism in Software Defined Network

Shailesh Bendale[1], Brijendra Gupta[2]
[1]SKNCoE, Research center
NBN Sinhgad School of Engineering
Pune, India
bendale.shailesh@gmail.com
[2]SKNCoE, research center
Siddhant College of Engg
Pune, India
gupbrij@rediffmail.com

Abstract— Network attacks are discovered using intrusion detection systems (IDS), one of the most crucial security solutions. Machine learning techniques-based intrusion detection approaches have been rapidly created as a result of the widespread use of standard machine learning algorithms in the security field. Unfortunately, as technology has advanced and there have been faults in the machine learning-based intrusion detection system, the system has consistently failed to fulfill the standards for cyber security. Generative adversarial networks (GANs) have drawn a lot of interest recently and have been utilized widely in anomaly detection due to their enormous capacity for learning difficult high-dimensional real time data distribution. Traditional machine learning algorithms for intrusion detection have a number of drawbacks that deep learning techniques can significantly mitigate. With the help of a real time dataset, this work suggests employing GANs and its variants to detect network intrusions in SDN. The feasibility and comparison results are also presented. For different kinds of datasets, the BiGAN outcomes outperform the GAN.

Keywords- SDN; GAN; BiGAN; Security; InSDN dataset.

I. INTRODUCTION

The growth and acceptance of network and information technology has resulted in a profound and broad integration of the digital world into every facet of daily social interactions. Important and confidential data is being stored online by more and more people, businesses, and governments. Information security is experiencing increasingly major problems as a result of increasingly varied threats as internet usage increases. One of the key unavoidable solutions to utilize in the cybersecurity field is intrusion detection as an active protection system. In order to find illegal activity on systems and computer networks that might be threats to the confidentiality, integrity, or availability of information, intrusion detection systems (IDS) distribute sample data to construct an intrusion detection model (CIA). When such assaults are found, an IDS provides intrusion alarms to counter them. An IDS's major objective is to distinguish between legitimate and malicious network traffic and computer usage, a task that standard stateless firewalls are unable to complete.

As a result of its high efficacy, efficiency, and rapid deployment during the past few decades, machine learning algorithms have been frequently applied to improve intrusion detection systems. Machine learning-based IDSes have gained popularity in modern times. However, numerous flaws in conventional machine learning algorithms have been amplified due to the enormous quantity and complexity of harmful assaults, such as the focus on analyzing low-dimensional data and lack of reaction to data with high dimensions and dependency on manual feature selection.

Machine learning tasks that process high-dimensional data can now be successfully completed using the manual feature selection method that deep learning omits. In recent years, the field of intrusion detection has seen a substantial increase in the application of deep learning algorithms. The accuracy and usability of IDSes have been significantly improved by recurrent neural networks (RNNs), automatic encoder (AE), and convolutional neural networks (CNNs) algorithms. A family of deep learning algorithms known as generative adversarial networks (GAN) is composed of two neural networks that compete with one another in a two-player game framework. Since GAN's paper was first released by Goodfellow et al. [1] in 2014.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

II. GENERATIVE ADVERSARIAL NETWORKS (GAN) AND BIDIRECTIONAL GENERATIVE ADVERSARIAL NETWORKS (BIGAN)

Generative Adversarial Networks (GAN)

The benefits of GAN in the area of intrusion detection are as follows. The first is that any data distribution may be accurately imitated by GAN, enabling it to provide real data that an IDS can exploit to its advantage since it is more easily accessible. Second, when confronted with adversarial attacks like attackers creating malicious traffic and making it look like regular traffic in an effort to deceive IDS into classifying it in the incorrect class, GAN has inherent advantages over other deep learning algorithms. Frequently, GAN is used in studies to enhance IDS or create novel attack patterns, such as producing adversarial malware instances. However, there are still few studies on GANs used for breach detection. This study will examine the effects of different GAN and BiGAN (Bidirectional Generative Adversarial Networks) factors on the efficiency of intrusion detection. [2-3]

In-depth coverage of the aforementioned topics will be provided in this paper, together with pertinent research on intrusion detection utilizing deep learning techniques, GAN, and its version. Identification of anomalies using BiGAN and GAN. IDS is an essential tool for network systems to use while looking for vulnerabilities in the network. Based on how an attack is found, IDS may be divided into two groups: anomaly detection-based IDS (ADIDS) and signature-based (misuse) IDS (SIDS). Misuse detection performs much worse when dealing with novel and unidentified attacks despite having a low false alarm rate along with high detection accuracy when dealing with known assaults. Anomaly detection, in comparison, handles new and unidentified attacks very well.

According to the annual Internet Security Threat Report (ISTR) from Symantec Corporation, hundreds of millions of new malware variants have been discovered annually in recent years, reaching a peak of 670 million variants in 2017. Moreover, compared to 2017, when 1 in 16 URLs were determined to be malevolent, 1 in 10 URLs were infected in 2018. Many researchers are concentrating more on anomaly identification these days due to the abundance of new malicious attacks that constantly emerge. Typically, the development of ADIDS involves two phases: the training phase, where new datasets are used to establish the system's reliable generalization ability against unknown intrusions, and the testing phase, where previous datasets are used to learn and construct behavior models that are deemed normal from typical trace data. When the ADIDS is doing a detection task, any behavior that is noticed to considerably deviate from the normal behavior model is termed an anomaly, which is another word for the intrusion. The premise of this technology is that malevolent behavior differs from ordinary user behavior.

Today, ADIDS have been developed using a range of machine learning algorithms that have learned from intrusion datasets. Finding patterns in vast quantities of data is the process of machine learning. A complex collection of "transfer functions" is what makes up machine learning models, and they can be used to recognize or forecast behavior. Effective use of machine learning algorithms can increase detection accuracy while lowering the need for human expertise. Generally speaking, there are two types of machine learning algorithms: supervised learning and unsupervised learning. The pertinent features and categories of marked training data are identified by supervised learning algorithms, which then learn and create data patterns. Each record in supervised learning IDS is made up of a data source and a label designating it as either invasive or normal. In addition, feature selection can be used in supervised learning-based IDS to eliminate unimportant features from the training data before instructing the classifier to comprehend the internal relationship between input data and labelled output values.

Algorithms for unsupervised learning construct joint density models from a set of random variables without class labels and draw out relevant information. In supervised learning IDS, the label of the output data is given and used to train the model to manage the unknown data; in unsupervised learning IDS, the label of the output data is unknown, and instead, the data is automatically classified into distinct classes during the learning process. The records in other tiny clusters will be classified as malicious attack data because of the performance disparities between malicious records and normal records, which cause them to correspond to various clusters. Normal records will form huge clusters.

One of the various methods developed by IDS researchers using machine learning algorithms is semi-supervised learning, which mixes supervised and unsupervised learning. It can be used to effectively cut down on the time and expenses needed when applied to IDS by combining its performance with that of a few select labelled data classifiers. Currently, numerous earlier works have put forth numerous different semi-supervised learning methods. Additionally, employing integrated approaches like improved integration, bagging integration, and stack integration, a number of machine learning algorithms are combined to enhance prediction performance.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

The significance of generative models has significantly grown due to their high adaptability in a variety of fields. They are trying to figure out how real data are distributed precisely for models. However, the majority of conventional generative models use the maximum likelihood principle to train the model in order to parameterize the model in a way that approximates the distribution of the actual data as closely as possible, which renders these models ineffective for handling the complexity of high-dimensional data. In order to address the shortcomings of other generative models, GAN uses the adversarial learning idea rather than maximum likelihood.[3-4]

Even though GAN offers a number of advantages and theoretical support, many flaws have been found, such as the discriminator's capacity issue and the inability to learn the inverse mapping. Due to these flaws, GAN is unable to reach its full potential. As a result, multiple studies have developed various GAN versions by changing the objective function, the structure, etc. to address these problems. BiGAN refers to one of the variations that deforms the structure.

Table 1 An well-organized performance assessment of numerous intelligent security mechanisms

Sr. No.	Paper & Year	Datasets / Real-time data	Algorithm used	Average Classification Accuracy	Shortcomings / Future work
1	R. Braga et al [2], 2010	real time network traffic	SOM (Self Organizing Maps)	98.61%	1. Only DDoS attack considered. 2. Can apply other types of attack detection for heterogeneous domains.
2	L. Grimaudo et al [3], 2013	data collected from the switches in the network for the DDoS attack	KNN	97.00%	1. Attack detection was done using few ports only. 2. Many port related attacks can be examined.
			SVM	82.00%	
			Naive Bayes	83.00%	
3	Y. Hong et al [4], 2015	real time dataset for the detection of DDoS attack	KNN (K Nearest Neighbors)	90.00%	1. Only 12 features used for intrusion classification. 2. Can use heuristic approach + SVM for increasing accuracy
			Naive Bayes	94.00%	
			K means	86.00%	
			K mediods	88.00%	
4	A. S. da Silva et al [5], 2016	real time network traffic	SVM	88.70%	1. Only anomaly detection done using few features of flow subset selection 2. New classification mechanism using combination of various network classifiers can be devised.
5	M. Nobakht et al [6], 2016	real time network traffic	SVM	96.20%	1. Intrusion detection and mitigation done for only IoT environment. 2. Statistically verification of intelligent method can be done with some standard datasets.
6	L. Barki et al [7], 2016	real time network traffic	Naive bayes	94.00%	1. Advanced IDS based on DDoS attack detection only. 2. Classification algorithm like HMM can be used for the same.
			k-NN	90.00%	
			k-means	86.00%	
			k-medoids	88.00%	
7	T. A. Tang et al [8], 2016	Deep NN (Neural network) algorithm with the help of NSL-KDD dataset	Deep NN	75.75%	1. Flow based anomaly detection only. 2. The proposed technique can be tested in real time traffic environment. 3. Work on more parameters can be done like latency and throughput.
			Decision tree	74.00%	
			SVM	70.90%	
			Bayes Theory	45.00%	
8	S. Nanda et al [9], 2016	intrusion detection mechanism	Decision Tree	86.19%	1. Verification of the proposed attack detection algorithms in real-time network 2. Verification of proposed intelligent mechanism with more standard datasets is required.
			BayesNet	91.68%	
			Decision Table	88.52%	
			Naive Bayes	87.78%	
9	P. Wang et al [10], 2016	network intrusion detection system	DT + SVM	97.55%	1. Network intrusion detection system for DDoS attack only. 2. Other combinations of classification algorithms can be used for feature selection.
10	T. Hurley et al [11], 2016	real time network traffic	HMM (Hidden Markov Models)	88.00%	1. Different feature vectors for classification can be used for Network intrusion detection system. 2. Fuzzy logic with HMM can be applied to increase the accuracy.
11	C. Song et al [12], 2017	KDD 99 dataset	Decision tree	82.48%	1. Signature based network intrusion detection system only. 2. Signatures need to be updated for increasing thee accuracy. 3. Multiple classifiers can be considered in future.
			Random Forest	98.75%	
12	I. Alawe et al [13], 2018	Public dataset for predicting network attack patterns	Decision Tree	86.19%	1. Deep Neural Network (DNN) on .VNF only for traffic classification using RNN and DNN. 2. This proposed solution can be made applicable for 5G network with minor modifications in virtualized environment.
			Bayes Net	91.68%	
			Decision table	88.52%.	
13	J. Xu t al [14], 2018	NSL-KDD Datasets for the anomaly based intrusion detection	KNN	98.14%	1. Deep Neural Network (DNN) on .VNF only for traffic classification. 2. Unclassified traffic can be identified using some other algorithm apart from Dijkstra's algorithm.
			SVM	91.04%	
			Naive Bayes	64.16%	
			Decision tree	99.70%	
			RF	99.70%	
			Bagging Trees	99.33%	
			RUSBoost	99.19%	
			AdaBoost	99.03%	
14	T. Tang et al [15], 2018	NSL-KDD dataset	RNN	89.00%	1. Gated Recurrent unit Recurrent Neural Network (GRU RNN) applied for intrusion detection only to standard dataset. 2. Proposed solution need to be tested in real time traffic environment. 2. More features need to be added for achieving more accuracy.
			SVM	65.67%	
15	N. Shone et al [16], 2018	KDD 99, NSL KDD dataset	Deep NN + RF	85.42% (NSL KDD) , 97.85% (KDD)	1. The proposed Non Symmetric Deep Auto Encoder (NDAE) network intrusion detection system can be tested in real time traffic.
16	AlEroud A. et al [17], , 2020	real time network traffic	SDN GAN deep NN	62.00%	1. Intrusion detection system for DDoS is only considered. 2. Combining the machine learning and deep learning classification algorithms to achieve the accuracy in future.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

Bidirectional Generative Adversarial Networks (BiGAN)

Donahue et al. introduced BiGAN, a brand-new unsupervised feature learning framework, in 2016. This framework expanded the standard GAN model by adding an inference network, allowing the discriminators to take into account inputs from both the data and latent spaces in addition to the data space alone. The latent representation is learned by combining autoencoder structure with a standard GAN architecture. There are three neural networks in BiGAN.

III. DATASET

The InSDN dataset and dataset which is created by us are in synchronization with that of the KDD and NSL KDD datasets. We will elaborate the features first and afterwards we will see the method to generate these features. [5-6] We will categories those into different types of attributes.

1. Attributes based on Network identifiers

Table 1 Attributes based on Network identifiers

Feature	Description
flow-id	Id of the flow
Src-IP	Source IP Address
Src-Port	Source port number
Dst-IP	Destination IP address
Dst-Port	Destination port number
timestamp	timestamp
Protocol-type	types of protocol e.g. TCP, UDP, etc

2. Attributes based on Bytes

Table 2 Attributes based on Bytes

Feature	Description
FwD-Header-Len	Total bytes used for the header in the Forwarding direction
BwD-Header-Len	Total bytes used for the header in the backward direction

3. Attributes based on Packets

Table 3 Attributes based on Packets

Feature	Description
Tot-Fwd-Pkts	Total packets in forwarding direction
Tot-Bwd-Pkts	Total packets in Backward direction
TotLen-Fwd-Pkts	Total Size of packets in the forward direction
TotLen-Bwd-Pkts	Total Size of packets in the backward direction
Fwd-Pkts-Len-Max	Max of the size of the packet in the forward direction
Fwd-Pkts-Len-Min	Min of the size of the packet in the forward direction
Fwd-Pkts-Len-Mean	Mean of the size of the packet in the forward direction
Fwd-Pkts-Len-Std	Standard deviation of the size of the packet in the forward direction
Bwd-Pkts-Len-Max	Max of the size of the packet in the backward direction
Bwd-Pkts-Len-Min	Min of the size of the packet in the backward direction
Bwd-Pkts-Len-Mean	Mean of the size of the packet in the backward direction
Bwd-Pkts-Len-Std	Standard deviation of the size of the packet in the backward direction
Pkts-Len-Max	Max of the length of a packet
Pkts-Len-Min	Min of the length of a packet
Pkts-Len-Mean	Mean of the length of a packet
Pkts-Len-Std	Standard deviation of the length of a packet
Pkts-Len-Var	Variance of the length of a packet
Pkts-Size-Avg	Average size of a packet

4. Attributes based on interarrival times

Table 4 Attributes based on interarrival times

Feature	Description
Duration	Duration of the flow in microseconds
Flow-IAT-Min	Min of the time between two packets sent in a flow
Flow-IAT-Max	Max of the time between two packets sent in a flow
Flow-IAT-Mean	Mean of the time between two packets sent in a flow
Flow-IAT-Std	standard deviation of the time between two packets sent in a flow
Fwd-IAT-Min	Min of the time between two packets sent in the forward direction
Fwd-IAT-Max	Max of the time between two packets sent in the forward direction
Fwd-IAT-Mean	Mean of the time between two packets sent in the forward direction
Fwd-IAT-Std	Standard deviation of the time between two packets sent in the forward direction
Fwd-IAT-Tot	a total of the time between two packets sent in the forward direction
Bwd-IAT-Min	Min of the time between two packets sent in the backward direction
Bwd-IAT-Max	Max of the time between two packets sent in the backward direction
Bwd-IAT-Mean	Mean of the time between two packets sent in the backward direction
Bwd-IAT-Std	standard deviation of the time between two packets sent in the backward direction
Bwd-IAT-Tot	a total of the time between two packets sent in the backward direction

5. Attributes based on Flow timers

Table 5 Attributes based on Flow timers

Feature	Description
Active-Min	Min of the time flow was active before becoming idle
Active-Max	Max of the time flow was active before becoming idle
Active-Mean	Mean of the time flow was active before becoming idle
Active-Std	Standard deviation of the time flow was active before becoming idle
Idle-Min	Min time flow was idle before becoming active
Idle-Max	Max time flow was idle before becoming active
Idle-mean	Mean time flow was idle before becoming active
Idle-Std	standard deviation time flow was idle before becoming active

6. Attributes based on Flag

Table 6 Attributes based on Flag

Feature	Description
Down/up-ratio	Download and upload ratio
Fwd-bytes/b-Avg	Average number of bytes bulk rate in the forward direction
Fwd-pkts/b-Avg	Average number of packets bulk rate in the forward direction
Fwd-Bulk-rate-Avg	Average number of bulk rate in the forward direction
Bwd-bytes/b-Avg	Average number of bytes bulk rate in the backward direction
Bwd-pkts/b-Avg	Average number of packets bulk rate in the backward direction
Bwd-Bulk-rate-Avg	Average number of bulk rate in the forward direction
Fwd-Seg-Size-Avg	Average size observed in the forward direction
Bwd-Seg-Size-Avg	Average size observed in the backward direction
Init-Fwd-Win-Byts	The total number of bytes sent in the initial window in the forward direction.
Init-Bwd-Win-Byts	The total number of bytes sent in the initial window in the backward direction.
Fwd-Act-Data-Pkts	count of packets with at least 1 byte of TCP data payload in the forward direction
Fwd-Seg-Size-Min	Minimum segment size observed in the forward direction
Flow-Byts/s	number of flow bytes per second
Flow-Pkts/s	number of flow packets per second
Fwd-Pkts/s	number of the forward packets per second
Bwd-Pkts/s	number of the backward packets per second

7. Attributes based on Subflow

Table 7 Attributes based on Subflow

Feature	Description
Subflow-Fwd-Pkts	the average number of packets in a subflow in the forward direction
Subflow-Fwd-Byts	the average number of bytes in a subflow in the forward direction
Subflow-Bwd-Pkts	the average number of packets in a subflow in the backward direction
Subflow-Bwd-Byts	the average number of bytes in a subflow in the backward direction

8. Attributes based on Flag counts

Table 8 Attributes based on Flag counts

Feature	Description
Fwd-PSH-Flags	number of PSH flags was set in packets traveling in the forward direction.
Bwd-PSH-Flags	number of PSH flags was set in packets traveling in the backward direction. (0 for UDP)
Fwd-URG-Flags	number of URG flags was set in packets traveling in the forward direction. (0 for UDP)
Bwd-URG-Flags	number of URG flags was set in packets traveling in the backward direction. (0 for UDP)
FIN-Flag-Cnt	number of packets with FIN
SYN-Flag-Cnt	number of packets with SYN
RST-Flag-Cnt	number of packets with RST
PSH-Flag-Cnt	number of packets with PSH
ACK-Flag-Cnt	number of packets with ACK
URG-Flag-Cnt	number of packets with URG
CWE-Flag-Cnt	number of packets with CWE
ECE-Flag-Cnt	number of packets with ECE

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

The above mentioned categories are useful for future direction and result discussion at various feature engineering or selection research.

IV. METHODS

GAN and BiGAN model training is the first step. At this point, G (Generator) is used to create examples of adversarial normal traffic using noise variables made up of random values that are uniformly distributed in the (0,1) range. These generated samples will be fed into the discriminator in the GAN-based IDS together with typical traffic examples from the KDD, NSL KDD, and InSDN datasets that have been de-labeled and turned to numbers. Joint pairs made up of real examples of regular traffic and encoded data created by the encoder compressing actual examples of regular traffic will be sent into D along with the generated examples from the IDS using BiGAN that have been mixed with the noise variables. D will then complete the training process by analysing these traffic records. Following the completion of this training process for a predetermined number of repetitions, the system moves on to the second level. All data in the KDD, NSL KDD and InSDN dataset, including examples of legitimate and fraudulent traffic, are pre-processed as training data were previously, and are then de-labeled and digitalized. This pre-processed data will be sent to D in x_{text} format for anomaly detection, which is distinct in other IDS and is unique to GAN-based IDS.

In the BiGAN-based IDS, x_{text} is compressed by the encoder to encoded data $x_{encoded}$, combined with $x_{encoded}$ to create a joint pair ($x_{encoded}$, x_{text}), and sent to D for detection rather than x_{text}.

Algorithm 1: IDS based on GAN

Input: Original normal traffic examples x_{normal} from the training set KDD, NSL KDD, InSDN; The random variable noise n;
Output: Detection result
Initialize the generator G and the discriminator D;
for Initialized GAN do
  for i = 1; :::; training times do
    for G do
      G generates the fake normal traffic examples $x_{generated}$ from n based on x_{normal};
      Send $x_{generated}$ to D;
    end
    for D do
      D classifies dataset including $x_{generated}$ and x_{normal};
    end
  end
end
for Trained D do
  D classifies the testing set KDD, NSL KDD, InSDN, getting predicted labels;
End

Algorithm 2: IDS based on BiGAN
Input: Original normal traffic examples x_{normal} from the training set KDD, NSL KDD, InSDN; The random variable noise n;
Output: Detection result
Initialize the generator G, the encoder E and the discriminator D;
for Initialized BiGAN do
  for i = 1; :::; training times do
    for G do
      G generates the adversarial normal traffic examples $x_{generated}$ from n based on x_{normal};
      Send joint pair (n, $x_{generated}$) to D;
    end
    for E do
      E compresses x_{normal} into latent space by encoding x_{normal} to $x_{encoded}$;
      Send joint pair ($x_{encoded}$, x_{normal}) to D;
    end
    for D do
      D classifies dataset including joint pairs (n, $x_{generated}$) and ($x_{encoded}$, x_{normal});
    end
  end
end
for Trained D do
  E encodes all traffic examples x_{test} in the testing set KDD, NSL KDD and InSDN to x_{test}(encoded);
  D classifies the joint pair dataset of (x_{test}(encoded), x_{test})+, getting predicted labels;
end

V. EXPERIMENTATION

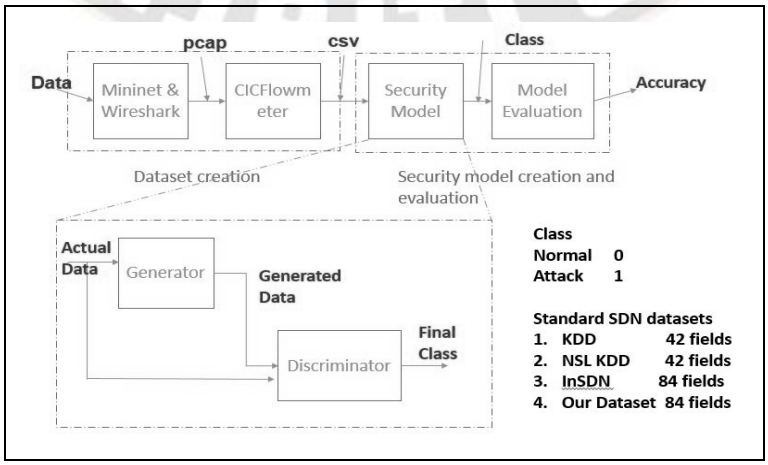

Fig. 1. An Intelligent Security mechanism for 5G network

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

To address this issue, the authors proposed an intelligent security approach. This solution can provide robust security for smart cities with various types of topologies and scenarios. The Figure gives an overview where irrespective of no physical device also the data can be generated at maximum possibilities on our own using the mentioned technique. The budget of the solutions is minimum and the cost of the companies will be reduced drastically. The accuracy, generator losses, discriminator losses, and the overall losses, precision, and recall values are taken into consideration for the naïve method used for the naïve results generated in this experimentation which will be discussed in detail at the end.[7-12]

We have made an effort to offer an open-source solution, as shown in Fig. 1. We developed an open-source solution that includes Wireshark, Google Colab, mininet, CICFlowmeter, and Kali Linux. Mininet, CICFlowmeter, and Wireshark were installed after the Kali Linux operating system in VMware. On the Mininet, we developed the Topology. Additionally, several topologies can be made. utilising different traffic creation commands, a significant amount of traffic may be generated utilising 5G technology.

Wireshark is used to track and record the same traffic. Wireshark must be used to capture the data, and then the data must be saved in a pcap file. The CICFlowmeter needs to receive the pcap file. The pcap file is transformed into a CSV file by the CICFlowmeter. The resulting CSV file is used to create datasets. It is feasible for several forms of attacks inside Kali Linux to repeat the same situation. Using the two cases mentioned above, we can build the normal and attack datasets. We followed this procedure to produce the dataset.

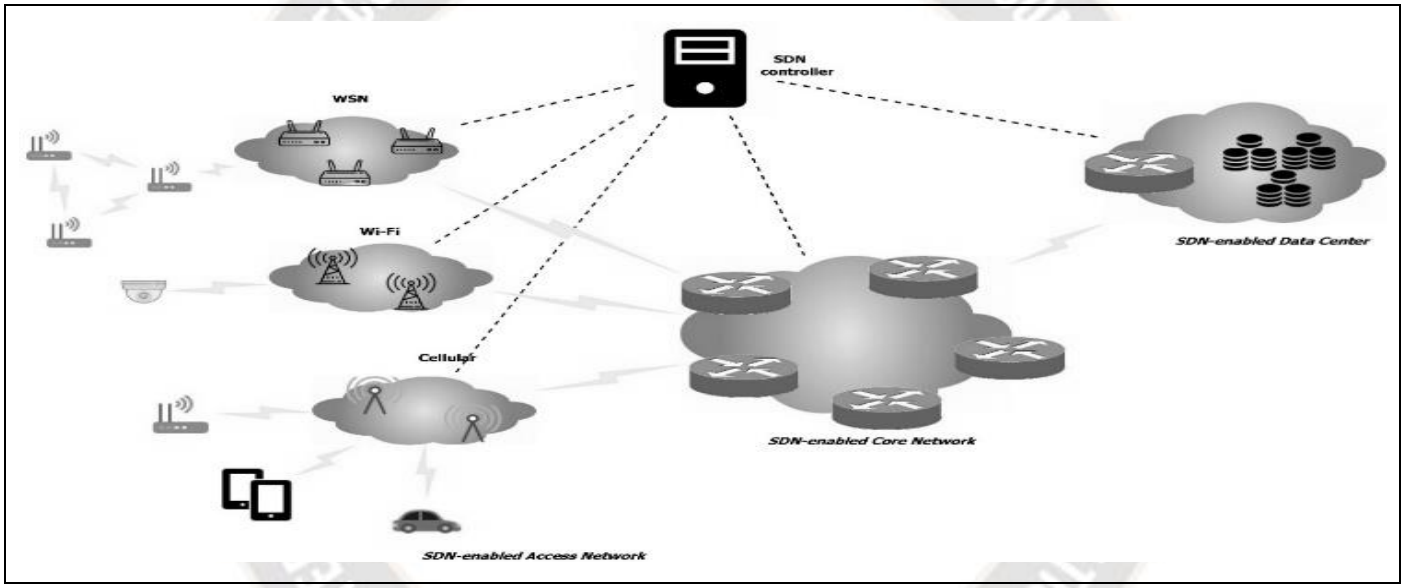

Fig. 2. Software Defined Network for Campus network

Two neural networks that are in competition with one another make up the intelligent security system for 5G. All types of data, whether original or created by the neural network, are generated using the first neural network. The second neural network aids in the detection of various forms of hostile and legitimate traffic during intrusions. It must also be able to recognize the traffic that the first neural network created.

Our main objective is to offer security in an open-source, economical way. The expense of developing testbeds will increase in the future. The expense will decrease in the near future.

VI. RESULTS AND DISCUSSION

The statistical study of two techniques, GAN and BiGAN, on three distinct datasets—KDD, NSL KDD, and InSDN datasets—is shown in Tables 9 and 10. To cross-check whether the findings are coming in correctly and consistently, two parameters—feature matching and cross entropy—were used. The tables show that the same accuracy can be achieved by cross-checking the findings with the two parameters.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

Table 9 Statistical analysis of GAN and BiGAN on three different datasets for feature matching

Dataset	InSDN	InSDN	InSDN	InSDN	KDD	KDD	NSL KDD	NSL KDD
Files used	Ovs	ovs	Ovs + metasploit	Ovs + metasploit				
techniques	GAN	BiGAN	GAN	BiGAN	GAN	BiGAN	GAN	BiGAN
No of epochs	200	200	200	200	200	200	200	200
features	77	77	77	77	122	122	127	127
Mode	Fm	fm	fm	Fm	fm	fm	fm	Fm
L norm	1	1	1	1	1	1	1	1
learning_rate	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001
batch_size	50	50	50	50	50	50	50	50
no of records	173073	173073	309669	309669	494021	494021	148516	148516
mean inference time	1.2756	0.0012	1.1817	0.0012	0.8569	0.7771	0.9052	0.8168
Precision	0.3201	0.3781	0.3063	0.5493	0.8497	0.8703	0.5651	0.4949
recall	0.3201	0.3781	0.308	0.5522	0.8632	0.8841	0.5651	0.866
F1 score	0.3201	0.3781	0.3071	0.5508	0.8564	0.8772	0.5651	0.6298
Accuracy	0.7281	0.7512	0.7236	0.8208	0.7964	0.9512	0.6462	0.7964

Table 10 Statistical analysis of GAN and BiGAN on three different datasets for cross Entropy

Dataset	InSDN	InSDN	InSDN	InSDN	KDD	KDD	NSL KDD	NSL KDD
Files used	Ovs	ovs	Ovs + metasploit	Ovs + metasploit				
techniques	GAN	BiGAN	GAN	BiGAN	GAN	BiGAN	GAN	BiGAN
No of epochs	200	200	200	200	200	200	200	200
features	77	77	77	77	122	122	127	127
mode	cross-e	cross-e	cross-e	cross-e	cross-e	cross-e	cross-e	cross-e
L norm	1	1	1	1	1	1	1	1
learning_rate	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001
batch_size	50	50	50	50	50	50	50	50
no of records	173073	173073	309669	309669	494021	494021	148516	148516
mean inference time	1.2575	0.001	1.2204	0.001	0.8217	0.7541	0.8579	0.7863
Precision	0.3047	0.3781	0.3052	0.5544	0.8503	0.7822	0.5662	0.4866
recall	0.3048	0.3781	0.3068	0.5574	0.8638	0.7946	0.5662	0.8516
F1 score	0.3048	0.3781	0.306	0.5559	0.857	0.7884	0.5662	0.6193
Accuracy	0.7219	0.7512	0.7231	0.8228	0.7906	0.916	0.6462	0.7906

VII. CONCLUSION

The significance of security has grown along with society's dependence on information technology and the associated risks. Intrusion detection technology is a crucial component of information security technology that is quickly evolving. Deep learning algorithms have been extensively used in intruder detection, especially with the growth and acceptance of neural network algorithms. The generative adversarial learning algorithm is a deep learning algorithm with a fresh idea, lots of potential, and a track record of success in other

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

fields such as anomaly detection. It has significant potential in the realm of network security intrusion monitoring.

Despite having a lot of promise, there aren't many IDS that use the generative adversarial networks class of algorithm. This is certainly due to the fact that this kind of algorithm is still in its infancy and that the hardware requirements are stringent, but one of the key factors is also the dearth of pertinent supplemental research data.

This study has shown how neural network settings can affect how well GAN-based IDS works. The experimental results show that the discriminator's epoch and number of hidden layers significantly affect how well it performs, and through the use of latent space, a GAN and autoencoder combination can dramatically increase IDS stability.

REFERENCES

[1] Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie, “A Survey on Software-Defined Networking”, IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 1, FIRST QUARTER 2015.

[2] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using NOX/OpenFlow,” in Proc. IEEE LCN’10, Denver, CO, USA, Oct. 2010, pp. 408–415.

[3] L. Grimaudo, M. Mellia, E. Baralis, and R. Keralapura,``Self-learning classifier for Internet traffic," in Proc. IEEE INFOCOM, Apr. 2013, pp. 3381_3386.

[4] Prof. Vaishali Sarangpure. (2018). Hybrid Hand-off Scheme for Performance Improvisation of Wireless Networks. International Journal of New Practices in Management and Engineering, 7(03), 08 - 14. https://doi.org/10.17762/ijnpme.v7i03.67

[5] Y. Hong, C. Huang, B. Nandy, and N. Seddigh,``Iterative-tuning support vector machine for network traffic classification," in Proc. IFIP/IEEE IM, May 2015, pp. 458_466.

[6] A. S. da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, “ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN,” in Proc. IEEE NOMS’16, Istanbul, Turkey, April. 2016, pp. 27–35.

[7] M. Nobakht, V. Sivaraman, and R. Boreli, “A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow,” in Proc. IEEE ARES’16, Salzburg, Austria, Aug. 2016, pp. 147–156.

[8] L. Barki, A. Shidling, N. Meti, D. G. Narayan, and M. M. Mulla, “Detection of distributed denial of service attacks in software defined networks,” in Proc. IEEE ICACCI’16, Jaipur, India, Sept. 2016, pp. 2576–2581.

[9] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, ``Deep learning approach for network intrusion detection in software defined networking," in Proc. IEEE WINCOM, Oct. 2016, pp. 258_263.

[10] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, ``Predicting network attack patterns in SDN using machine learning approach," in Proc. IEEE NFV-SDN, Nov. 2016, pp. 167_172.

[11] P. Wang, K. M. Chao, H. C. Lin, W. H. Lin, and C. C. Lo, “An efficient flow control approach for SDN-based network threat detection and migration using support vector machine,” in Proc. IEEE ICEBE’16, Macau, China, Nov. 2016, pp. 56–63.

[12] T. Hurley, J. E. Perdomo, and A. Perez-Pons, “HMM-based intrusion detection system for software defined networking,” in Proc. IEEE ICMLA’16, Anaheim, CA, USA, Dec. 2016, pp. 617–621.

[13] C. Song, Y. Park, K. Golani, Y. Kim, K. Bhatt, and K. Goswami, ``Machine-learning based threat-aware system in software defined networks,' in Proc. IEEE ICCCN, Jul./Aug. 2017, pp. 1_9.

[14] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, ``Improving traffic forecasting for 5G core network scalability: A machine learning approach," IEEE/ACM Trans. Netw., vol. 32, no. 6, pp. 42_49, Nov./Dec. 2018.

[15] J. Xu, J. Wang, Q. Qi, H. Sun, and B. He,``Deep neural networks for application awareness in SDN-based network," in Proc. IEEE MLSP, Sep. 2018, pp. 1_6.

[16] T. Tang, S. A. R. Zaidi, D. McLernon, L. Mhamdi, and M. Ghogho, “Deep recurrent neural network for intrusion detection in SDN-based networks,” in Proc. IEEE NetSoft’18, Montreal, Canada, 2018.

[17] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” IEEE Trans. Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, Feb 2018.

[18] AlEroud A., Karabatis G. (2020) SDN-GAN: Generative Adversarial Deep NNs for Synthesizing Cyber Attacks on Software Defined Networks. In: Debruyne C. et al. (eds) On the Move to Meaningful Internet Systems: OTM 2019 Workshops. OTM 2019. Lecture Notes in Computer Science, vol 11878.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems, 2014, pp. 2672– 2680.

[20] Shailesh Pramod Bendale, Jayashree Rajesh Prasad, Rajesh Shardanand Prasad, Chapter 27 - State of the art for edge security in software-defined networks, Editor(s): Rajiv Pandey, Sunil Kumar Khatri, Neeraj kumar Singh, Parul Verma, Artificial Intelligence and Machine Learning for EDGE Computing, Academic Press, 2022, Pages 411-424, ISBN 9780128240540, https://doi.org/10.1016/B978-0-12-824054-0.00010-1.

[21] S. P. Bendale and J. Rajesh Prasad, "Security Threats and Challenges in Future Mobile Wireless Networks," 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN), Lonavala, India, 2018, pp. 146-150, doi: 10.1109/GCWCN.2018.8668635.

[22] S. P. Bendale and G. V. Chowdhary, "Stable path selection and safe backup routing for Optical Border Gateway Protocol (OBGP) and Extended Optical Border Gateway Protocol (OBGP+)," 2012 International Conference on Communication, Information & Computing Technology (ICCICT), Mumbai, 2012, pp. 1-6, doi: 10.1109/ICCICT.2012.6398201.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.7011
***Article Received:** 04 April 2023 **Revised:** 22 May 2023 **Accepted:** 03 June 2023*

[23] Bendale, Shailesh and Prasad, Jayashree Rajesh, Preliminary Study of Software Defined Network on COVID-19 Pandemic Use Cases (May 28, 2020). Available at SSRN: https://ssrn.com/abstract=3612815 or http://dx.doi.org/10.2139/ssrn.3612815

[24] Shailesh Pramod Bendale, Jayashree Rajesh Prasad, "Security Challenges to provide Intelligence in SDN with the help of Machine Learning or Deep Learning", IJAST, vol. 29, no. 05, pp. 356 - 363, Apr. 2020.

[25] Smith, J., Ivanov, G., Petrović, M., Silva, J., & García, A. Detecting Fake News: A Machine Learning Approach. Kuwait Journal of Machine Learning, 1(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/142

[26] M. A. R. S. Mr. Shailesh P. Bendale, "Implications and Application of Artificial Intelligence and Machine Learning Concepts on Software Defined Network and Its Future Prospects.", IJAST, vol. 29, no. 4s, pp. 1142 - 1152, Mar. 2020.

[27] S. Shah and S. Pramod Bendale, "An Intuitive Study: Intrusion Detection Systems and Anomalies, How AI can be used as a tool to enable the majority, in 5G era," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-8, doi: 10.1109/ICCUBEA47591.2019.9128786.

[28] Prasad, J.R., Bendale, S.P., Prasad, R.S. (2021). Semantic Internet of Things (IoT) Interoperability Using Software Defined Network (SDN) and Network Function Virtualization (NFV). In: Pandey, R., Paprzycki, M., Srivastava, N., Bhalla, S., Wasielewska-Michniewska, K. (eds) Semantic IoT: Theory and Applications. Studies in Computational Intelligence, vol 941. Springer, Cham. https://doi.org/10.1007/978-3-030-64619-6_18

[29] Tamboli, D. N., & Bendale, S. P. (2022). Crowdfunded Assassinations and Propaganda by Dark Web Cyber Criminals. In R. Rawat, S. Telang, P. William, U. Kaur, & O. C.U. (Ed.), Dark Web Pattern Recognition and Crime Analysis Using Machine Intelligence (pp. 74-84). IGI Global. https://doi.org/10.4018/978-1-6684-3942- 5.ch006

[30] Gawali, N. N., & Bendale, S. (2022). Artifical Intelligence and Machine Learning Algorithms in Dark Web Crime Recognition. In R. Rawat, U. Kaur, S. Khan, R. Sikarwar, & K. Sankaran (Ed.), Using Computational Intelligence for the Dark Web and Illicit Behavior Detection (pp. 126-149). IGI Global. https://doi.org/10.4018/978-1- 6684-6444-1.ch007