Anomaly based network intrusion detection for IoT attacks using deep learning technique

Sharma, Bhawana; Sharma, Lokesh; Lal, Chhagan; Roy, Satyabrata

DOI
10.1016/j.compeleceng.2023.108626

Publication date
2023

Document Version
Final published version

Published in
Computers and Electrical Engineering

Computers and Electrical Engineering 107 (2023) 108626

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Anomaly based network intrusion detection for IoT attacks using deep learning technique[☆]

Bhawana Sharma [a], Lokesh Sharma [a], Chhagan Lal [b], Satyabrata Roy [a,*]

[a] *Manipal University Jaipur, Jaipur, India*

[b] *Department of Intelligent Systems, Cybersecurity Group, TU Delft, Netherlands*

ARTICLE INFO

Keywords:
Intrusion detection system
Machine learning
Deep learning
GAN
DoS
Internet of Things

ABSTRACT

Internet of Things (IoT) applications are growing in popularity for being widely used in many real-world services. In an IoT ecosystem, many devices are connected with each other via internet, making IoT networks more vulnerable to various types of cyber attacks, thus a major concern in its deployment is network security and user privacy. To protect IoT networks against various attacks, an efficient and practical Intrusion Detection System (IDS) could be an effective solution. In this paper, a novel anomaly-based IDS system for IoT networks is proposed using Deep Learning technique. Particularly, a filter-based feature selection Deep Neural Network (DNN) model where highly correlated features are dropped has been presented. Further, the model is tuned with various parameters and hyper parameters. The UNSW-NB15 dataset comprising of four attack classes is utilized for this purpose. The proposed model achieved an accuracy of 84%. Generative Adversarial Networks (GANs) were used to generate synthetic data of minority attacks to resolve class imbalance issues in the dataset and achieved 91% accuracy with balanced class dataset.

1. Introduction

Internet of Things (IoT) is a popular technology due to its immense potential for applications ranging from the healthcare system to transportation and smart cities. IoT refers to the interconnected network of a large number of physical devices, which are termed as *things*. The devices connected in the network have limited computational power and storage capabilities. With an increased number of heterogeneous physical devices connected through the internet, an enormous amount of data is generated, and thus IoT networks becomes more lucrative targets for attackers. The challenges in IoT are mainly in terms of resource constrained (i.e., low computation, and storage capacity) devices, and cyber security [1] which makes it challenging to implement necessary security measures in them. Nowadays, for performing computing on a huge amount of data, Machine Learning (ML) and Deep Learning (DL) based techniques are considered appropriate and they provide better computation results. Best features are selected from the dataset using different ML techniques, and then the ML methods are applied to the features for classification. In contrast, in the deep learning methods, the best features are automatically extracted during the application of the technique on a dataset [2]. In IoT networks, addressing cyber security issues has become a top priority, and it may be achieved by planning and implementing an effective Intrusion Detection System (IDS) at edge nodes. Over the past few decades, both ML and DL-based IDSs gave good outcomes

[☆] This paper is for regular issues of CAEE. Reviews were processed by Associate Editor Dr. Saim Ghafoor and recommended for publication.

[*] Corresponding author.

E-mail addresses: bhawana2104@gmail.com (B. Sharma), lokesh.sharma@jaipur.manipal.edu (L. Sharma), c.lal@tudelft.nl (C. Lal), satyabrata.roy@jaipur.manipal.edu (S. Roy).

https://doi.org/10.1016/j.compeleceng.2023.108626
Received 29 August 2022; Received in revised form 31 January 2023; Accepted 7 February 2023
Available online 16 February 2023

in detecting the attacks in IoT networks, and as a result, in the realm of cyber security, it is becoming increasingly popular and suitable for detecting threats in such networks. An IDS can detect attacks within the network or host and are usually classified as:

- **Signature-based IDS:** It monitors the network and detects the anomalies by matching the specific signatures or patterns pre-stored in the memory. Since the pre-determined signatures or patterns need to be stored in the memory, the technique requires a large amount of storage and can only detect the pre-stored attacks. Therefore, this technique easily detects known attacks but cannot detect new attacks and thus requires timely updating of the database.
- **Anomaly-based IDS:** It is suitable to also detect new attacks depending on the behavior of the system. It detects the activity which is identified as different from the predefined normal behavior, and then it is could be considered an attack for further analysis. It can detect new attacks based on any variation from the predefined normal behavior, thus there can be false positives as well. [3].

In network security, anomaly detection is widely used for intrusion detection, and various deep learning techniques-based anomaly detection techniques are used for IDSs in IoT networks. In IoT systems, various threats or attacks occur daily, so there is a need to identify those attacks and find a way to mitigate attacks to safeguard the network. The IoT layers are namely the perception layer, network layer, and application layer.

In the perception layer, data is collected from sensors or IoT devices and then, after processing, transferred to network layers. Prominent security issues in the perception layer are a physical attack on the IoT devices and, tampering with the hardware components of devices, disturbance during the transmission of signals. IoT devices have limited storage capacity and power consumption, and computation capacity is also limited, making the devices vulnerable to attacks. Node capture, data injection attack, and replay attacks can hamper the confidentiality of data. At the perception layer, encryption of data from one end to another can be used to mitigate these issues.

In Network layers, data is transmitted using cloud computing platforms, gateways, switches, and routers. Recent communication technologies such as Bluetooth (BLE), low-power 6LoWPAN, ZigBee, ZWave, IEEE 802.15.4, WiFi, RFID, and short-range Near Field Communication (NFC) are widely used.

There is a high risk of a DoS attack by flooding the target with traffic which makes the device or network not accessible to its intended users, and a man-in-the-middle attack where the adversary can attack the privacy of data at the network layer by eavesdropping. To safeguard network objects, we need protocols and software to monitor and protect them from attacks.

In the application layer, application-specific service is provided and is responsible for evaluating, monitoring, and implementing the IoT Systems. Different applications need different authentication mechanisms, and thus privacy and authentication are difficult. There are large amounts of connected devices at the application layer sharing the data, so it is essential to monitor how and with whom the data is shared and manage the amount of data shared among users. Data is transmitted from the perception layer to the network layer, so there is a risk of SYN flooding, malicious scripts, phishing, and data transmission attacks.

Based on the layers, the threats are node capture at the perception layer, DDoS at the network layer, and code injection at the application layer. Researchers have identified different threats in IoT systems, such as DoS/DDoS, Sybil attacks, man-in-the-middle attacks, etc. DoS/DDoS attack is the most prominent in the IoT network layer [4,5]. Studies showed that the researchers had demonstrated the proposed model's performance with high accuracy. However, there are some issues while implementing such models.

Deep learning model provides high overall detection accuracy, but the performance declines with imbalanced data. The number of normal and attack class records is different, which results in an imbalanced dataset. Thus, the model is biased to the class having larger samples. Therefore, there is a need to balance the dataset before training the model. The dataset is balanced using undersampling and oversampling techniques. Undersampling decreases the number of majority classes, and oversampling increases the minority class, but that can lead to overfitting as the same data is oversampled. Generating synthetic data of minority attack classes can reduce the effect of the imbalanced dataset in training the model.

Dataset with a large number of features requires complex deep learning models with a large number of neurons and huge computation, so to reduce the computational cost and number of neurons, useful features need to be selected from the dataset. For classification, instead of processing with the complete set of features, we can reduce the dataset by selecting the best features, thus reducing the training time of the model as well.

The model with too low complexity can cause under fit model and may not be able to perform well on training data. In contrast, a too high complexity model can cause an overfitting problem, which can be removed using various regularization techniques.

In this paper, we propose a novel design based on a deep learning technique for intrusion detection. The Deep Neural Network (DNN) model classifies the multi-class with Normal and four attack types, namely Generic, Exploits, Fuzzers, and DoS. Specifically, this paper makes the following major contributions.

1. Propose the design of an IDS for IoT systems based on deep learning techniques. Particularly, we apply Generative Adversarial Networks (GANs) to generate the synthetic data for minority attacks and balance the class load in the dataset.
2. Propose an approach to reduce the number of features using the filter-based method. One of each two highly correlated features is dropped from the dataset, which reduces the training time complexity of the model.
3. Evaluation of the proposed model using publicly available dataset is done. Moreover, tuning the model with various parameters, hyperparameters, and regularization techniques has been performed to further improve its performance.

The remaining part of the paper is arranged as follows. Section 2 provides the literature review of the state-of-the-art deep learning-based intrusion detection systems. Section 3 provides details about our proposed framework for intrusion detection. Section 4 presents details about the dataset used, evaluation approach taken, and result analysis of the proposed model. Lastly, Section 5 concludes the paper discussing some applications as well as limitations of the proposed model and thereby suggests some future scope of the study.

2. Related work

In this section, we provide a systematic literature survey of different DL techniques for detecting cyber-attacks in traditional and IoT networks. Different ML techniques require selecting the features from the dataset efficiently before the training of the model for classification, whereas the advantage of automatic extraction of features in DL techniques has made them more efficient and popular in the past few years. DL techniques are extensively used in IDSs in cyber security [6].

In [7], S. M. Kasongo et al. proposed a WFEU-FFDNN wireless IDS model. The authors selected the feature using wrapper-based method Trees algorithm, and for classification they constructed Feed-Forward DNN model. Experiments conducted on two different dataset, UNSW-NB15 and AWID, achieved an accuracy of 87.10% and 99.66% for binary, 77.16% and 99.77% for the multiclass.

In [8], G. Bae et al. proposed a DL anomaly detection model using AutoEncoder (AE) having three encoder and decoder layers. The experiment was conducted on KDD Cup '99 dataset and the Korea steel company's real dataset, they achieved the accuracy ranging 84% to 100% on KDD Cup ‘99 and 95% on the real dataset. In [9], M. A. Ferrag et al. constructed DL models and evaluated them on two dataset (i.e., CICIDS2018, and Bot-IoT). Discriminative convolutional neural network and generative/unsupervised deep autoencoder's model achieved an accuracy of 97.3% on CSE-CIC-IDS2018 dataset and 98.3% on Bot-IoT dataset.

In [10], Shone et al. proposed a model for unsupervised feature learning using an AE. The author utilized two different dataset, namely NSL-KDD and KDD Cup ‘99, to evaluate their model. Experimental results showed that the model achieved an accuracy of nearly 85% on the NSL-KDD dataset and 97.85% on the KDD Cup ‘99 dataset.

In [11], G. Thamilarasu et al. constructed a DNN model having three hidden layers and a ReLU activation function with 150 epochs. Experimental results confirmed that the model outperforms many other models with precision above 95% on different attack classes in the dataset. In [12], Ge, M., Syed et al. proposed a DL-based intrusion detection technique on BoT–IoT dataset. The model was constructed using two dense layers of 512 neurons in each layer and ReLU activation function. The last dense layer consists of two neurons and the Softmax function. The categorical features were encoded, and with the transfer learning concept, the same encoding used in binary classification was adopted for the multiclass model. Moreover, the model was tuned with different parameters and hyperparameters such as learning rates, the number of hidden layers, epochs, weight decay, and dropout regularization techniques. Feedforward neural networks (FNN) model for multiclass achieved an accuracy of 99.79% over four classes, and the FNN model for binary class achieved an accuracy of above 99%.

In [13], A. Nagisetty et al. proposed DL based different deep models CNN, DNN, MLP, and autoencoder for detecting an intruder in the IoT networks and tested the models on UNSW-NB15 and NSL-KDD99 dataset. Experimental results showed that the MLP model achieved the highest F1-score of 99.28% and 95.76% on respective dataset. DNN model obtained the highest accuracy of 99.24%. In [14], Y. Zhou et al. proposed Deep Feature Embedding Learning (DFEL) framework based on DL and reduced the dimensions/features of the dataset, including the transfer learning concept. The model achieved an accuracy of 93.13%.

In [15], Qiu, H., Dong et al. design a novel adversarial attack on DL-based IDS in the IoT network where a slight change in the attributes of the packet alters the prediction results of the model and leads to the DoS attacks. The model achieved an Attack Success Rate (ASR) higher than 95%. In [16], Xiao, Y et al. proposed a CNN-IDS model consisting of five hidden layers, pooling layer, and convolution layers. Redundant and unnecessary features were removed using dimensionality reduction methods. The model was trained on 10% of the KDDCup99 dataset, and tuned with different parameters. Experimental results showed that the model with a dropout rate of 0.3, 50 epochs, and 128 batch size achieved an accuracy of 94.0%.

In [17], Sun, P et al. developed a DL-based IDS model using the concept of hybrid method. LSTM-CNN model was constructed for extracting the features and classification. The weight optimization method was used to reduce the effect of the unbalanced data. The experiments was conducted using the CICIDS2017 dataset, and the model achieved an overall accuracy of 98.67% . In [18], Kim, J et al. constructed a deep learning-based CNN model for DoS attacks detection and compared it with the RNN model. The model was tested on the CICIDS2018 and KDD Cup99 datasets. Experimental results showed that the model achieved accuracy of 99% on the KDD Cup99 and 91.5% accuracy on the CICIDS2018 dataset.

In [19], Kasongo, S. M. et al. proposed a FNN model, where features were selected using the information gain filter method. The model was tuned with the different number of neurons, learning rates, and compared with other ML techniques such as SVM and DT. The authors utilized the NSL-KDD dataset for testing the model. The experiment conducted on the model having 3 hidden layers, 30 neurons in each layer, and 0.005 learning rate obtained an accuracy of nearly 99% and 88% on the training set and the testing dataset respectively for binary classification, and for multi-class classification, the model constructed with 150 neurons in each layer and 0.05 learning rate achieved an accuracy of nearly 99:5% and 86:19% on the training set and the testing set respectively.

In [20] Fenanir, S et al. proposed a lightweight IDS model. Features were selected using filter method, and then ML techniques were applied for classification. The authors compared different ML techniques and found that the Decision tree provides the best classification model on different dataset. Features were selected based on filter methods such as correlation filter method PCC, SCC, and KTC on various dataset using different threshold values. The author applied different ML techniques for classification, such as Logistic Regression (LR), Decision Tree (DT), and SVM, on different dataset, namely KDD99, NSL-KDD, and UNSW-NB15. By

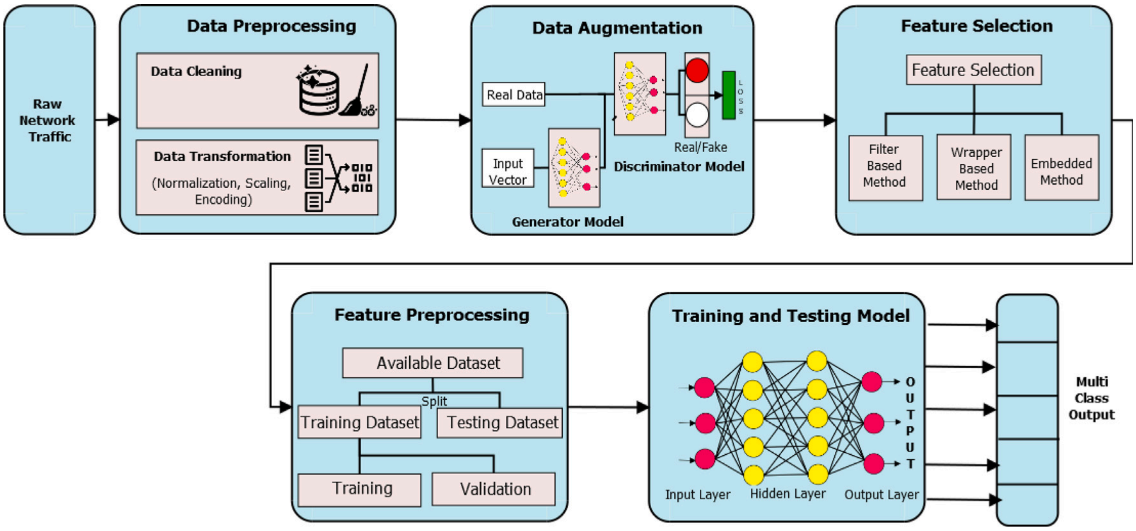


Fig. 1. Workflow of the proposed framework.

Experiments they found that the DT classification model outperform others with an accuracy rate of 98% and a low false-positive rate.

In [21], Almaiah, M et al. studied the Shamoon attack behavior using FPSO (Frequency Particle Swarm Optimization). Fog nodes deliver three types of data: industrial, medical, and educational. Shamoon primarily targets industrial data. The author evaluated the method by estimating its fitness value and best cost. The author also observed the distribution of Shamoon attacks, tracked their movement, and concluded that Shamoon attacks take the shortest path possible. So by locating the first node, the source of the attack can be found.

In [22], the author proposed smart health monitoring system using IoT, where the three key health parameters: blood oxygen, heart rate, and body temperature are measured using sensors, and then the data collected is encrypted using Advanced Encryption Standard (AES) algorithm to protect the data from internal and external attacks in the cloud. Encrypted data is then transferred to a medical organization, where servers receive and then decrypt the data. The experimental result showed that the proposed method has a 95% confidence interval.

In [23], the author proposed a blockchain-based deep learning model for IIoT. Improved the consensus algorithm for blockchain and compared it with the existing consensus protocol. The experimental result showed that the proposed model provides advancement compared to the benchmark models. In [24], the author proposed a blockchain-based healthcare system and implemented blockchain with a homomorphic encryption technique for the privacy of patient health data. A graph theory-based binary spring search(BSS) consisting of hybrid DNN is implemented for intrusion detection in the IoT, and used Hyperledger Calliper for blockchain network and compared the proposed method with the benchmark models. The experimental result showed that the confirmation time of the proposed model is less and provides security with less computational cost. A jamming attack is the most common type of DoS attack. In [25], the author proposed an improved PSO (Particle Swarm Optimization) algorithm for the detection of a jamming attack and compared the proposed algorithm with the existing PSO algorithm and other optimization techniques. The experimental result showed that the proposed method outperforms in terms of minimal fitness value and coverage area.

The literature study showed that in the proposed methods, the overall accuracy might be high, but the detection accuracy of each class in multiclass classification is low. It is mainly due to the class imbalance problem. Particularly, model training accuracy is biased to the class having a larger sample size. Therefore, there is a need to resolve the class imbalance issue and improve the accuracy of each class by increasing the samples of classes having smaller samples. First, we propose the use of GAN technique to generate synthetic data for the minority classes. Second, we reduce the features by removing irrelevant and redundant features, which decreases the training runtime of the model. We can apply different feature selection techniques to select relevant features. In particular, we have applied a filter-based method where we dropped one of the two highly correlated features to reduce the training runtime of the Deep learning model while achieving higher model accuracy.

3. Proposed framework

In this section, we propose DNN-based framework for detecting vulnerabilities and attacks in IoT networks. Firstly, we discuss the primary workflow of the framework and subsequently we discuss the evaluation of each phase and the key contributions of each phase thereafter.

Table 1
Dataset statistics showing number of packets in each category.

Category	No of packetsr	Category	No of packets
Normal	93000	Reconnaissance	13987
Generic	58871	Analysis	2677
Exploits	44525	Backdoor	2329
Fuzzers	24246	Shellcode	1511
DoS	16353	Worms	174
		Total Packets	**257673**

3.1. Workflow of the proposed framework

It is composed of five main steps: Data Preprocessing for extraction of the data, Data augmentation for class balancing, Feature selection for identifying the best suited features, Feature preprocessing for encoding and splitting the dataset, and Training and testing of the DNN model. Fig. 1 depicts the workflow of the proposed framework. Below we explain working principles of each of these steps in detail.

- **Data Preprocessing:** First raw network traffic is collected using the network analyzer tool, and then we extract the features from the packets. Redundant packets are dropped in the dataset, and then we collect the samples of classes in the dataset. We encoded the symbolic data into integer values using the encoding technique and standardized the numerical values in the dataset using min–max standardization [21].
- **Data Augmentation:** Imbalance data is the condition when the number of samples of each class is not equally distributed, creating the problem of data skewness where the data of one class is larger and makes the model biased to the class having a larger number of samples.
  Before classification, training data is resampled to avoid class imbalance. Resampling is done by oversampling when the number of minority class samples is increased, and undersampling is when the number of majority class samples is decreased. Researchers are using the oversampling method, but it causes the problem of overfitting as the same data learned by oversampling [22]. The synthetic minor oversampling technique (SMOTE) selects the samples of the minority class and generates the new samples, but the problem of class overlap and noise is generated.
  Recently GANs have been used based on the neural network, which generates synthetic data similar to original data [23]. GANs are better than other conventional methods such as SMOTE as it avoids the overfitting, class overlap, and noise problem and allows for tuning the model, which helps to develop an accurate model [24].
  We generate new samples from the existing dataset to have a balanced dataset. We increase the number of packets of the minority attack classes using GANs. Generating the synthetic data makes the dataset more balanced. GANs, introduced by Goodfellow et al. [26] consists of two different neural networks which work together, one generates the synthetic data and another detects the real data and synthetic data. The generative networks generate the synthetic data from an input dataset, and the discriminator network detects the real data deriving from the input set and the fake data deriving from the generative network.
- **Feature Selection:** Feature selection improves the efficiency of storage and reduces the computing cost [27]. Feature selection is categorized into the following techniques:
  - **Filter Methods:** It measures the relationship of features by their correlation scores. Features are selected based on scores of statistical methods, and we select the features depending on the scores and the threshold value. The most common techniques are correlation, information gain, and the Chi-Square test.
  - **Wrapper Methods:** A subset of features is selected, and the ML model is trained on it. Based on the accuracy of the model, the features in the subset are updated by the addition or removal of features from it. The most common types of such methods are backward elimination and forward selection.
  - **Embedded Methods:** The benefits of both the methods (i.e., filter and wrapper) are combined in this method which extracts the best features and maintains the computational cost. The most common types of this method are LASSO regularization [1] and Random Forest [25].

  Filter Method is much faster and requires less computation as compared to the other two methods, therefore we use it for our experiments.
- **Feature Preprocessing:** After selecting the features, and dropping and encoding the features, we split the processed data into three different sets, namely training, validation, and testing, containing the labels of both normal and attack type classes.
- **Training and Testing the dataset:** In the training phase, the DNN model is trained on the processed data coming from the training set. The trained model is then tested with the data from the testing set and classifies the data as normal and attack types.

Table 2
Features of dataset.

S.No	Features names	S.No	Features names
1	dur	23	Stcpb
2	dpkts	24	dtcpb
3	spkts	25	dwin
4	dbytes	26	tcprtt
5	sbytes	27	synack
6	rate	28	ackdat
7	Sttl	29	smean
8	dttl	30	dmean
9	sload	31	trans_depth
10	dload	32	response_body_len
11	sloss	33	ct_srv_src
12	dloss	34	is_ftp_login
13	sinpkt	35	ct_dst_ltm
14	dinpkt	36	ct_dst_sport_ltm
15	sjit	37	ct_src_dport_ltm
16	djit	38	ct_state_ttl
17	ct_dst_src_ltm	39	ct_flw_http_mthd
18	ct_ftp_cmd	40	ct_srv_dst
19	ct_src_ltm	41	proto
20	is_sm_ips_ports	42	service
21	swin	43	state
22	attack_cat	44	label

3.2. Evaluation and analysis of proposed framework

Here we discuss in detail the work done in each phase with its analysis.

- **Dataset Description:** In this work, we used a dataset available publicly, namely UNSW-NB15, as a source of raw network traffic. Deployment of the real testbed, realistic traffic generation, and labeled data are some oft the major causes of selecting this dataset. UNSW-NB15 dataset was produced in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS), and it contains the normal and attack behavior of network traffic. The research team generated the data from live network traffic using two servers in a hybrid mode and the IXIA Perfect Storm tool, normal activities were generated by one server. The malicious activities were generated by another server in the network. The data generated was collected and stored in Tcpdump format, and the Pcap files are used to extract the features by Argus and Bro-IDS tools in Linux Ubuntu 14.0.4. This dataset consists of one normal class and nine attacks type classes, namely, Exploits, Generic, DoS, Analysis, Fuzzers, Backdoors, Worms, Reconnaissance, and Shellcode. The data set includes 42 features and one label that indicates whether an attack is normal or attack. The dataset contains a total of 257 673 with 93 000 records of the normal label and 164 673 records from the different attack classes. Table 1 shows the dataset statistics with respect to the number of packets in each class/category. The complete description of the dataset, as well as testbed setup, is presented in [28].
- **Data Pre-processing**
  We encode and normalize the dataset features, which are then fed into the DL model. We uploaded the dataset stored in a .csv file on Google Colab, which provides 12 GB of RAM and a single Tesla K80 GPU. The dataset consists of total features equal to 44, including 39 numerical values, four categorical values, and one label indicating normal/attacks type class as shown in Table 2. We dropped columns of redundant labels and encoded the categorical features into integer values using label encoding. Symbolic features are 'proto' 'service', 'state', and 'attack_cat' having (133,13,11,10) values respectively, are converted into integer values using label encoding. Dataset is normalized using min–max normalization as shown in Eq. (1) to change the values of numeric columns having features of different ranges to a common scale. X_{new} denotes the new normalized value in Eq. (1), where X_{min} and X_{max} are the minimum and maximum values in the X column.

  $$X_{new} = (X - X_{min})/(X_{max} - X_{min}) \tag{1}$$

  The complete range of values of the X column, from X_{min} to X_{max}, are mapped within the range of 0 to 1. Normalization is needed to avoid huge variance in the convergence problem and bring all the features on the same scale so that optimization is possible. After the data pre-processing operations, we obtain the dataset, which is rescaled.
- **Data augmentation**
  In our experiment, the dataset is imbalanced, so to avoid the imbalance issue, we resampled the dataset before classification. We undersample the majority class and oversample the minority class to balance the dataset. We considered only 5 classes in our experiment where the majority class are Normal, Generic, Exploits, and Fuzzers classes, and we under-sampled the majority classes by randomly selecting the 25 K packets, and for the minority DoS class, we generated the new data using GAN. We randomly selected the sample of size 25 K packets for the class having the number of packets larger than 25 K size, and for the class having the number of packets lower than 25 K sample size, all packets are selected. In specific, for each class with more than 25 K packets, we selected 25k size packets randomly from the dataset. The total number of packets extracted

from the dataset is 115 599, including normal and four attack type classes. The number of DoS packets is lower than 25k, so we applied GANs to the samples and generated the packets using the generator and discriminator model, as shown in Fig. 2. Then generated packets are concatenated with the original packets, making the number of packets 25 K in size.
The GAN model invented by Ian Goodfellow consists of generator neural network G, which generates synthetic data, and discriminator neural network D, which identifies the real and generated data. The generator's aim is to create data that is similar to real data, and the aim of the discriminator model is to identify the real data and generated data. Working of GAN is

- Generator G takes the input noise z and generates the data.
- The generated data is fed to the discriminator D along with the real data.
- Discriminator D identifies the real and generated data and returns 0 for generated data and 1 for real data.
- The aim of the generator model is to minimize the loss function, whereas the aim of the discriminator is to maximize the loss function.

GAN loss function (Min (G) Max(D)) can be defined as Eq. (2):

$$V(D,G) = E_{x-P_{data(x)}}[\log D(x)] + E_{z-P_z}[\log(1 - D(G(z)))], \tag{2}$$

where, G = Generator, D = Discriminator, x = sample from real data, z = sample from generator data, $P_{data(x)}$ = Distribution of real data, P_z = Distribution of generator data, G(z) = Generator network, D(x) = Discriminator generator

- G(z) is the generator that tries to generate the distribution P_g from the distribution of noise P_z and makes P_g close to the distribution of real data $P_{data(x)}$. Discriminator tries to identify real and generated data.
- GAN model continues to train and adjust G and D such that D is not able to identify the difference between real and generated data. The Optimal discriminator is when $P_g = P_{data(x)}$

The generator model consists of three Dense hidden layers of 32, 64, and 128 neurons and LeakyReLu activation function, and the last layer consists of 39 neurons and sigmoid activation function. The discriminator model consists of 4 Dense hidden layers of 128,64,32,16 neurons and LeakyReLu activation function, and the last layer consists of one neuron and sigmoid activation function. During model compilation and weight update, we applied binary_crossentropy loss function and Adam Optimizer and trained the model for 1000 epochs. The dataset is balanced, with each class having the number of packets equal to 25 K. The dataset contains 44 columns having 39 numerical values, four categorical values, and one label indicating normal or attacks type class. Generator loss and discriminator loss plot is shown in Fig. 3 where blue shows the generator loss and orange shows the discriminator loss.

- **Feature Selection:**
In the field of ML and DL, feature selection is crucial. It increases the storage efficiency and reduces the computational cost, thus improves the performance of the model. In this paper, we selected the features from the dataset using the Pearson's Correlation Coefficient (PCC) filter method.

  - **Pearson's correlation coefficient:** The PCC ρ measures the dependence between two random variables X and Y and is given by the equation:

$$\rho = cov(X,Y)/\sigma(X)\sigma(Y) \tag{3}$$

where, cov and σ denote the covariance and the standard deviation, respectively, and p denoting the value of Pearson correlation coefficient ranges from -1 to 1. If two random variables, X and Y, are highly related, then the value of σ is nearly close to the values -1 and 1, whereas if X and Y are totally uncorrelated, then the value of σ is 0. Thus, highly correlated features are redundant features, and one of the features can be dropped. The correlation between features in the dataset is obtained and the features having high negative or positive values are considered as highly correlated. We derived the Correlation map showing the highly correlated features, and the features having a value greater than the threshold value are dropped. The correlation matrix of the features of the dataset is shown in Fig. 4.

During model building, features having a strong correlation with a threshold value of 0.95 are selected, and any other features has been dropped. Features 'dloss', 'dwin', 'ct_ftp_cmd', 'ct_src_dport_ltm', 'ct_srv_dst' and 'label' are dropped in UNNB_SW15 dataset. After dropping the columns, the new dataset has 38 features.

- **Training and Testing**
After feature selection and preprocessing, we split the dataset into 75%–25% training and testing sets. We used the training set to train the DNN model. It is tested using a testing set. Both the sets contain different labels showing normal or attack classes. The accuracy achieved by training the model is then verified by the testing dataset. For multi-class classification, we build a DNN model with three dense layers, and each layer has different neurons, where a range of normal and different attack classes are treated as separate classes.

4. Experiment and result analysis

In this section first the experimental set up and detailed specifications are described. Then a detailed analysis of the result is presented. Further, the comparison of result with other existing techniques have been presented.

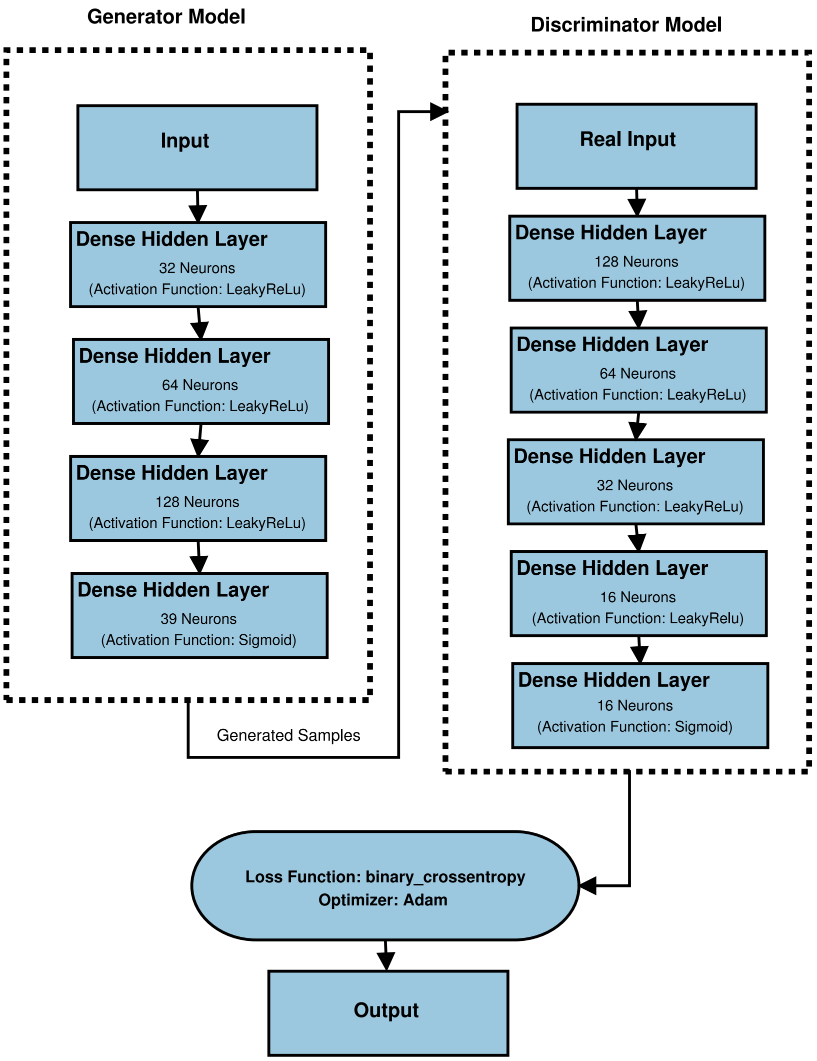


Fig. 2. Architecture of GAN model.

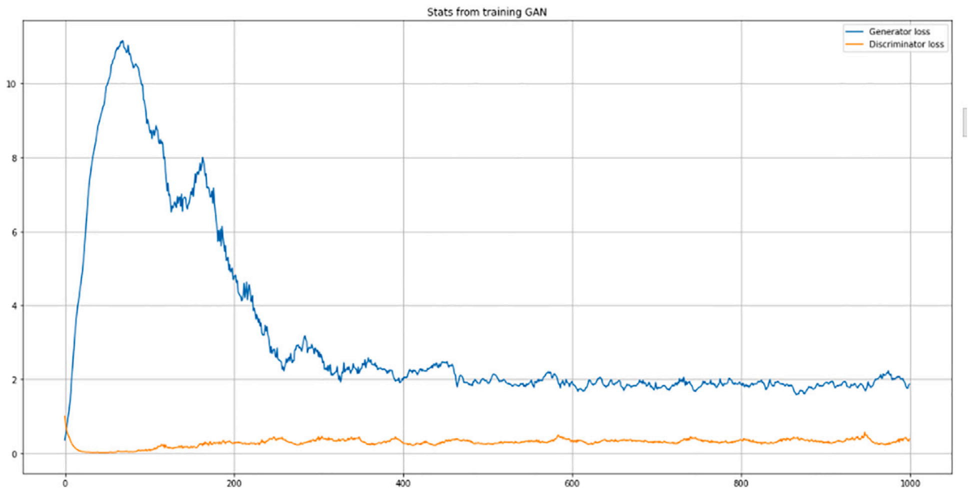


Fig. 3. GAN loss.

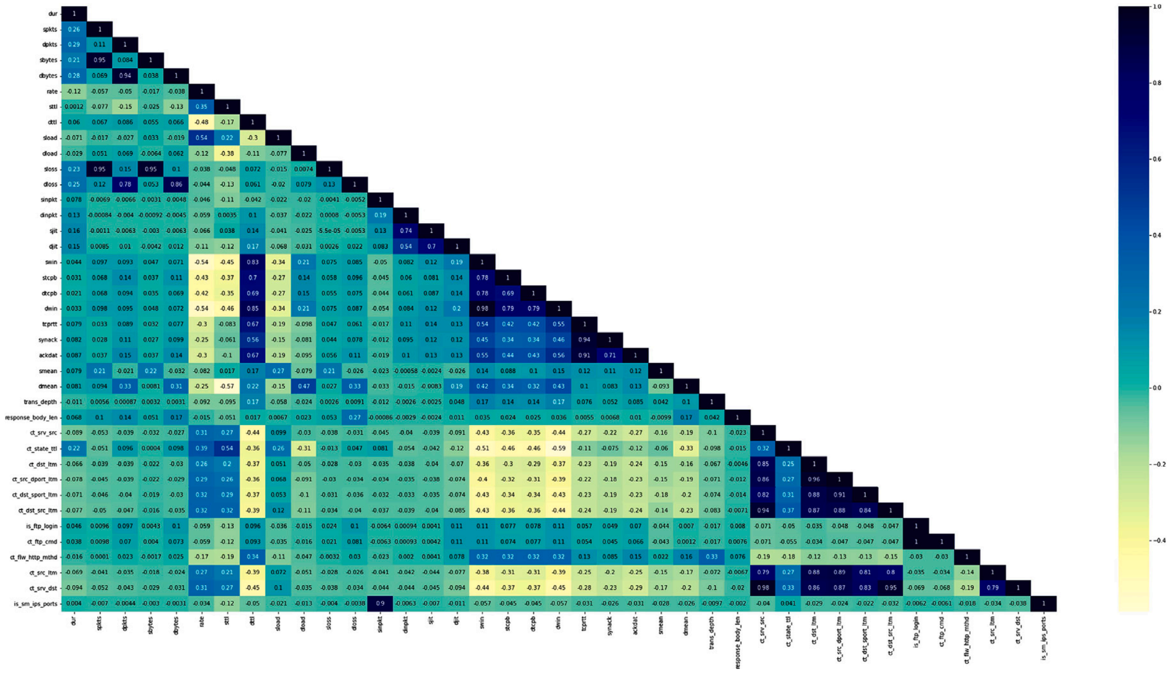


Fig. 4. Correlation matrix of the dataset.

4.1. Experimental setup

We conducted our experiment with the split of 60% for training, 15% for validation and remaining 25% for testing sets respectively using Google Colab and TensorFlow library on our dataset. To train the DNN model, we used training and validation data. The DNN model had three dense hidden layers, each with 64 neurons, and the final layer has five neurons, equal to the number of classes in the dataset. We used the ReLU activation function on each dense hidden layer, and on the model's final layer, we used the Softmax activation function. During the model compilation, we applied Adam optimizer and a sparse categorical cross-entropy loss function for weight update. The DNN model's architecture, including the Adam optimizer and loss function, is shown Fig. 5. We also tuned the DNN model using various parameters and hyperparameters to increase the classification accuracy and to avoid the overfitting of the model [29]. For tuning the model, we used the $L2$ regularization technique with 0.0001 weight decay and 0.0001 dropout rate, and the different number of hidden layers and neurons in each layer. Since our dataset is large, the classification accuracy does not improve with different regularization techniques.

We build different architectures of DNN with various dense hidden layers and numbers of neurons in each layer to improve accuracy. The best performance is obtained with the DNN model, having three dense hidden layers of 64 neurons in each dense layer. The complexity of the model depends on the number of dense hidden layers and neurons. If the number is small, then the model has small complexity and is underfit model, whereas if the number is large, then the model has large complexity and is overfit model. We applied different learning rates using Adam optimizer and found that the default learning rate of 0.001 gives better accuracy.

4.2. Analysis of results

We examine the outcome of the proposed DNN model for different attack types in the dataset in terms of evaluation metrics namely accuracy, precision, recall, and F1 score, and we present confusion matrices for the classification. The model performance with two class types C1 and C2, is evaluated using the following metrics in the classification.

- True positives (TP): The packets of C1 in the dataset is correctly identified as packets of C1.
- True negatives (TN): The packets of C2 in the dataset is correctly identified as packets of C2.
- False positives (FP): The packets of C2 in the dataset is identified as packets of C1.
- False negatives (FN): The packets of C2 in the dataset is identified as packets of C1.
- Precision (P): The fraction of number of packets which are correctly identified as packet belonging to class C1 to the total number of packets which are identified as packets of class C1 including false positives.
- Recall (R): The fraction of number of packets of class C1 that are identified correctly as true positive.
- F-measure (F1-score): Computes the harmonic mean of P and R to show the number of packets identified incorrectly.

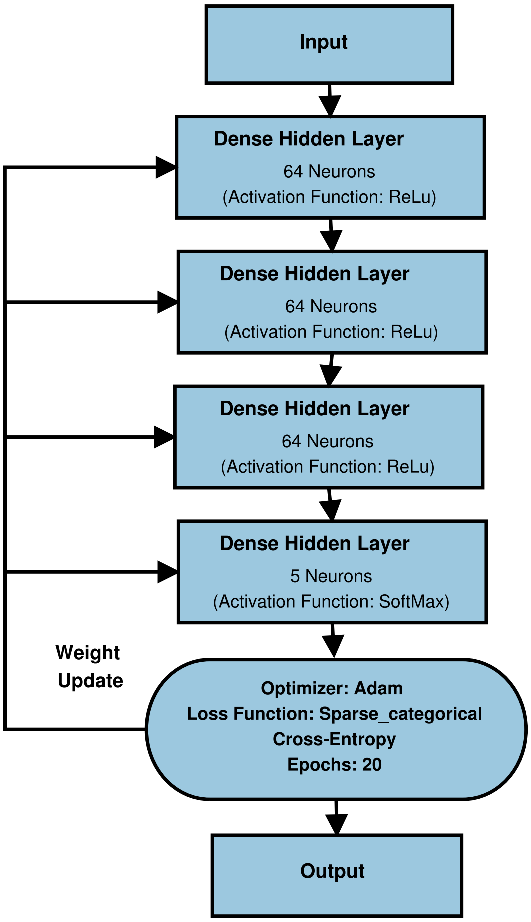


Fig. 5. Architecture of the DNN Model.

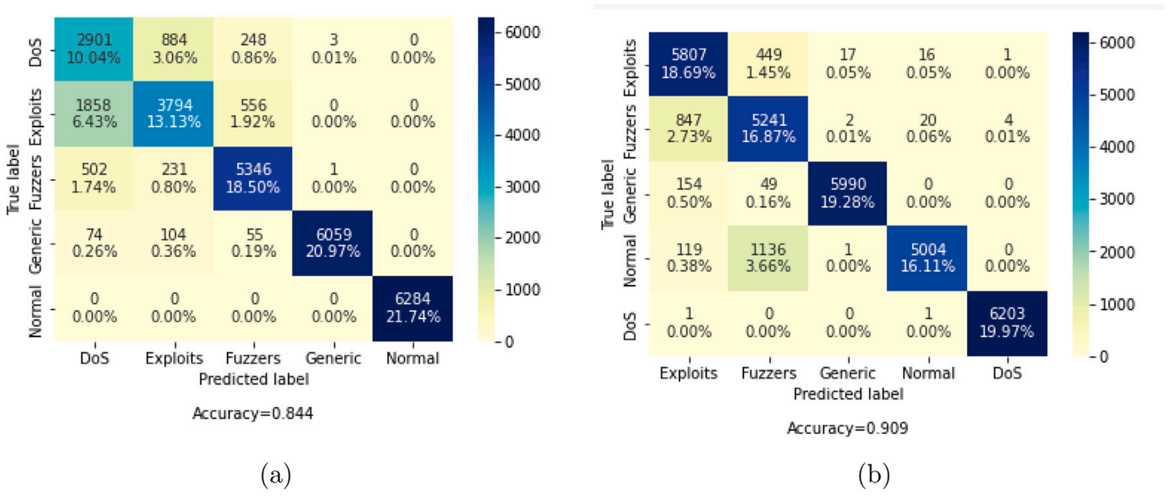


Fig. 6. (a) Confusion matrix of DNN and (b) Confusion matrix of GAN-DNN model for the multi-class classification.

- Accuracy: Estimates the fraction of packets in the dataset that are identified correctly as class C1 and C2. Definitions of the metrics are shown in Table 3.

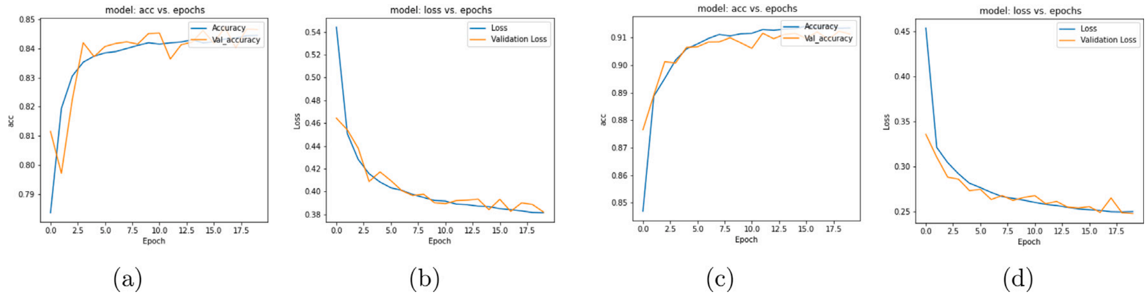


Fig. 7. (a) Accuracy vs. Epochs of DNN model, (b) Loss vs Epochs of DNN model, (c) Accuracy vs. Epochs of GAN-DNN model with increased number of packets using GAN, (d) Loss vs. Epochs of GAN-DNN model with increased number of packets using GAN.

Table 3
Evaluation metrics for the classification.

Precision	P = TP/(TP+FP)
Recall	R = TP/(TP+FN)
F1 Score	F = 2PR/(P+R)
Accuracy	A = (TP+TN)/(TP+TN+FP+FN)

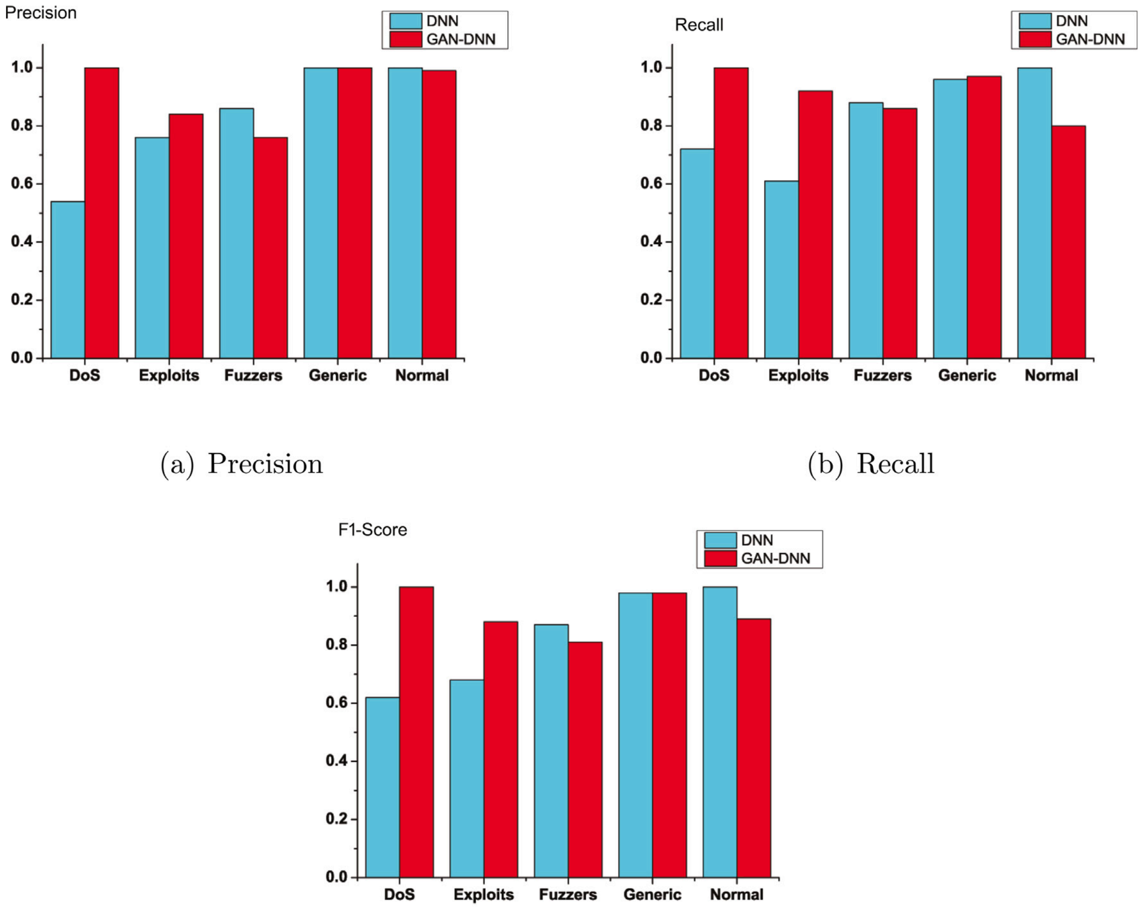
Table 4
Evaluation metrics of DNN and GAN-DNN model for multi-class classification.

Classification report						
	DNN model			GAN-DNN model		
	Precision	Recall	F1-score	Precision	Recall	F1-score
DoS	0.54	0.72	0.62	1	1	1
Exploits	0.76	0.61	0.68	0.84	0.92	0.88
Fuzzers	0.86	0.88	0.87	0.76	0.86	0.81
Generic	1	0.96	0.98	1	0.97	0.98
Normal	1	1	1	0.99	0.8	0.89
Accuracy			**0.84**			**0.91**

4.3. Comparison of results

We trained the DNN model using original samples having 86 699 training dataset for 20 epochs and verified the model accuracy with the 28 900 testing dataset for multi-class classification. We increased the number of DoS attacks using GAN and trained the same DNN model using increased samples having 93 184 training dataset and 31 062 testing dataset for 20 epochs, and to evaluate the performance of the model, we used the following evaluation metrics:

1. **Confusion matrix:** We obtained the confusion matrix in terms of the number and percentage of different class packets as it is shown in Fig. 6. The confusion matrix of DNN shows that the number of TP of DoS attacks is 2901, which is equal to 10.04% of the testing dataset, whereas the confusion matrix of GAN-DNN shows that the number of TP of DoS attacks is 6203 which is equal to 19.97% of the testing dataset. The number of samples has increased, the TP of DoS attacks by 10%.
2. **Precision, recall, and F1-Score:** We obtained the evaluation metrics for multi-class classification as shown in Table 4. The DNN model gives the precision, recall, and F1-Score average value of 0.84, whereas the GAN-DNN model gives the average value of 0.91. With the increase in the number of DoS samples, the precision, recall, and F1-Score are increased by 0.46, 0.28, and 0.38, respectively. A high precision value is achieved when the number of FPs is less, and a high recall value shows less number of FNs. High F1-Score shows the efficacy of correctly classifying the test dataset with the respective class. Comparisons of Precision, Recall, and F1-Score of DNN and GAN-DNN model is shown in Fig. 8.
3. **Accuracy and Loss:** The proposed DNN model has achieved an accuracy of 84.4% for the classification of normal and packets with four attack categories. By increasing the number of packets of minority attacks using GANs model, we can increase the accuracy. In our dataset, DoS attack category have a smaller number of packets, and with the increased number of DoS attacks using GANs model, we achieved an accuracy of 91% with reduced loss as it is shown in Fig. 9. The loss function value indicates the error in the predicted value, and depending upon the loss value, the weights are updated. Loss function value decreases with the increase in epochs, and training accuracy increases with the increase in epochs. Training and validation accuracy graph plot of DNN model is shown in Fig. 7. Fig. 7(a) shows that with the increase in epochs, both types of accuracy increase to 0.844, and they overlap with each other. Training and validation loss graph plotted in Fig. 7(b) shows that the loss value decreases to 0.38 after 20 epochs. Graph plots in Figs. 7(c) and 7(d) show that the training and validation accuracy of the GAN-DNN model increases to 0.91, and they overlap with each other, whereas the loss decreases to 0.25 value after



(a) Precision

(b) Recall

(c) F1-Score

Fig. 8. (a) Precision, (b) Recall, and (c) F1-Score of DNN and GAN-DNN model.

20 epochs. We applied the testing dataset on the DNN model and achieved an accuracy of 84%, and the GAN-DNN model achieved an accuracy of 90.9%.

We compared the proposed model with the other ML techniques, such as KNN, Decision Tree (DT), RandomForest, and SVM model, in terms of accuracy of the multi-class classification as illustrated in Fig. 9. We used the same UNSW-NB dataset, randomly selected the samples of size 25 K if the packets are larger than the sample size, encoded the categorical features using label encoding, and normalized the dataset. We applied the Pearson correlation coefficient to reduce the number of features in the dataset and implemented different machine learning models, namely KNN, Decision Tree, and SVM model using the Support Vector Classifier (SVC). Fig. 9 illustrates the accuracy of different models for multi-class classification. We conclude that the proposed DNN model achieved the highest accuracy of 91% with high precision, recall and F1-Score for DOS attacks.

5. Conclusion and future works

In the recent years, researches have been carried out on network security and privacy in the IoT domain. Various ML and DL-based solutions for designing IDSs have also been proposed. Deep learning is widely used for intrusion detection in IoT networks. In this paper, we proposed a framework using DNN, including feature selection and GANs for increasing the number of packets of minority attack classes to resolve the class imbalance issue in the dataset and tested the framework's efficacy for the dataset in the IoT networks. In multi-class classification, the prediction accuracy of 85% is reported by the classifier, and using GANs, 91% accuracy is yielded by the classifier. The results achieved are significant in the field of cyber security. By feature selection and increasing the number of packets of DoS attacks, we can classify the normal and attacks categories with high accuracy. However, we encountered several issues. The proposed model can be applied to other datasets with imbalanced class labels, and GANs can

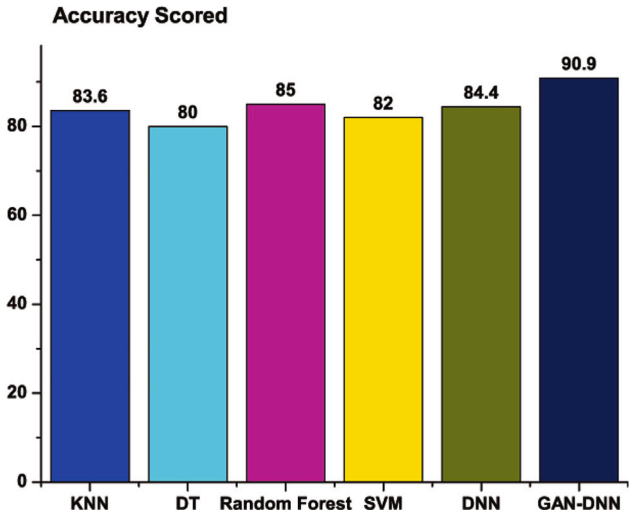


Fig. 9. Comparisons of Accuracy of different models.

be adopted to generate traffic of the minority attacks in the dataset. The feature selection technique has reduced the number of features, and thus decreased the cost of the model.

Various feature selection and extraction techniques could be implemented to reduce the number of features before the classification. In multi-class classification, we considered only five types of traffic in this work. Other minority attack classes can be categorized in the dataset in future and a new DNN-based classifier with increased accuracy and reduced loss in terms of FN and FP predictions can be developed. Further, a DNN model to operate in real-time for detection of intrusion in IoT networks can be a future scope of this work.

CRediT authorship contribution statement

Bhawana Sharma: Conceptualization, Methodology, Software, Writing – original draft. **Lokesh Sharma:** Visualization, Investigation, Supervision. **Chhagan Lal:** Supervision, Software. **Satyabrata Roy:** Writing – review & editing, Supervision, Investigation.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.compeleceng.2023.108626.

Data availability

Data will be made available on request.

References

[1] Khan MN, Rahman HU, Almaiah MA, Khan MZ, Khan A, Raza M, et al. Improving energy efficiency with content-based adaptive and dynamic scheduling in wireless sensor networks. IEEE Access 2020;8:176495–520.

[2] Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol 2021;32(1):e4150.

[3] Sharma B, Sharma L, Lal C. Anomaly detection techniques using deep learning in IoT: A survey. In: 2019 international conference on computational intelligence and knowledge economy. 2019, p. 146–9. http://dx.doi.org/10.1109/ICCIKE47802.2019.9004362.

[4] Altulaihan E, Almaiah MA, Aljughaiman A. Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions. Electronics 2022;11(20):3330.

[5] Al Nafea R, Almaiah MA. Cyber security threats in cloud: Literature review. In: 2021 international conference on information technology. IEEE; 2021, p. 779–86.

[6] Ma W. Analysis of anomaly detection method for Internet of Things based on deep learning. Trans Emerg Telecommun Technol 2020;31(12):e3893.

[7] Kasongo SM, Sun Y. A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Comput Secur 2020;92:101752.

[8] Bae G, Jang S, Kim M, Joe I. Autoencoder-based on anomaly detection with intrusion scoring for smart factory environments. In: International conference on parallel and distributed computing: Applications and technologies. Springer; 2018, p. 414–23.

[9] Ferrag MA, Maglaras L, Moschoyiannis S, Janicke H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J Inf Sec Appl 2020;50:102419.

[10] Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE Trans Emerg Top Comput Intell 2018;2(1):41–50.

[11] Thamilarasu G, Chawla S. Towards deep-learning-driven intrusion detection for the Internet of Things. Sensors 2019;19(9):1977.

[12] Ge M, Syed NF, Fu X, Baig Z, Robles-Kelly A. Towards a deep learning-driven intrusion detection approach for Internet of Things. Comput Netw 2021;186:107784.
[13] Nagisetty A, Gupta GP. Framework for detection of malicious activities in IoT networks using keras deep learning library. In: 2019 3rd international conference on computing methodologies and communication. IEEE; 2019, p. 633–7.
[14] Zhou Y, Han M, Liu L, He JS, Wang Y. Deep learning approach for cyberattack detection. In: IEEE INFOCOM 2018-IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE; 2018, p. 262–7.
[15] Qiu H, Dong T, Zhang T, Lu J, Memmi G, Qiu M. Adversarial attacks against network intrusion detection in IoT systems. IEEE Internet Things J 2020.
[16] Xiao Y, Xing C, Zhang T, Zhao Z. An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access 2019;7:42210–9.
[17] Sun P, Liu P, Li Q, Liu C, Lu X, Hao R, et al. DL-IDS: extracting features using CNN-LSTM hybrid network for intrusion detection system. Secur Commun Netw 2020;2020.
[18] Kim J, Kim J, Kim H, Shim M, Choi E. CNN-based network intrusion detection against denial-of-service attacks. Electronics 2020;9(6):916.
[19] Kasongo SM, Sun Y. A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access 2019;7:38597–607.
[20] Fenanir S, Semchedine F, Baadache A. A machine learning-based lightweight intrusion detection system for the Internet of Things. Rev D'Intelligence Artif 2019;33(3):203–11.
[21] Almaiah A, Almomani O. An investigation of digital forensics for shamoon attack behaviour in FOG computing and threat intelligence for incident response. J Theor Appl Inf Technol 2020;15:98.
[22] Siam AI, Almaiah MA, Al-Zahrani A, Elazm AA, El Banby GM, El-Shafai W, et al. Secure health monitoring communication systems based on IoT and cloud computing for medical emergency applications. Comput Intell Neurosci 2021;2021.
[23] Almaiah MA, Hajjej F, Ali A, Pasha MF, Almomani O. A novel hybrid trustworthy decentralized authentication and data preservation model for digital healthcare IoT based CPS. Sensors 2022;22(4):1448.
[24] Ali A, Almaiah MA, Hajjej F, Pasha MF, Fang OH, Khan R, et al. An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network. Sensors 2022;22(2):572.
[25] Al Hwaitat AK, Almaiah MA, Almomani O, Al-Zahrani M, Al-Sayed RM, Asaifi RM, et al. Improved security particle swarm optimization (PSO) algorithm to detect radio jamming attacks in mobile networks. Int J Adv Comput Sci Appl 2020;11(4).
[26] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016, http://www.deeplearningbook.org.
[27] Sharma B, Sharma L, Lal C. Feature selection and deep learning technique for intrusion detection system in IoT. In: Proceedings of international conference on computational intelligence. Springer; 2022, p. 253–61.
[28] UNSW-NB15 Dataset. 2015, https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ [Online; accessed 19-Oct-2021].
[29] Sharma B, Sharma L, Lal C. Anomaly based network intrusion detection for IoT attacks using convolution neural network. In: 2022 IEEE 7th international conference for convergence in technology (I2CT). 2022, p. 1–6. http://dx.doi.org/10.1109/I2CT54291.2022.9824229.

Bhawana Sharma received her B. Tech. degree in Computer Science and Engineering from University of Rajasthan in 2006 and M. Tech. degree from Malaviya National Institute of Technology in 2009. She is currently pursuing her Ph.D. from Manipal University Jaipur. She is currently working on network security, intrusion detection system and software defined networks.

Lokesh Sharma received his M. Tech. and Ph.D. degrees from Rajasthan Technical University and Manipal University Jaipur in 2009 and 2018 respectively. He is currently working as Associate Professor in Manipal University Jaipur. He is a professional member of IEEE and ACM. His research interest includes Software Define Networks and IoT. He is author of many high impact research papers.

Chhagan Lal received his M. Tech. and Ph.D. degrees from Indian Institute of Information Technology, Allahabad and Malaviya National Institute of Technology. Currently, he is a senior researcher in CyberSecurity at Department of Intelligent Systems, Delft University of Technology, Netherlands. He is a member of IEEE and authored several research articles. His research interest includes Network security and machine learning.

Satyabrata Roy received his M. Tech. and Ph.D. degrees from KIIT University and Manipal University Jaipur in 2014 and 2020 respectively. He is currently working as Associate professor in Manipal University Jaipur. He is a senior member of IEEE and professional member of ACM. He has authored several research articles. His research interest includes information security, IoT and Cellular Automata.