An Effective and Intelligent Intrusion Detection System using Deep Auto-Encoders

T.POONGOTHAI[1], K.JAYARAJAN[2]

[2]Professor, St. Martin's Engineering College, Secunderabad, India

[1]Professor, Malla Reddy Engineering College for Women, Secunderabad, India

[1]poongothait@gmail.com, [2]jayarajinfoster@gmail.com

Abstract

Cyber threats or attacks are major security concern with the monumental growth and increased number of Internet connecting devices. Furthermore, the attackers exploit sophisticated attacks and persist for long period of time easily. The dynamic nature and large volume of cyber attacks require a responsive, adaptive and scalable protective mechanism. Many supervised and unsupervised learning methods have been developed from the domain of machine learning and data mining algorithms to detect and classify those attacks. The main aim of this proposed work is to examine the suitability of deep learning approach for intrusion detection system. In this work, IDS model is proposed based on Deep Auto Encoders (DAE). This model is trained and tested with NSL_KDD data set. The performance of the proposed system is compared with conventional machine learning algorithms namely, Logistic Regression (LR), Naïve Bayes (NB), K-Nearest Neighbor (KNN), Decision Tree (DT) and Random Forest (RF) methods. The experiment results show the improved accuracy of DAE in intrusion detection in comparison with classical machine learning algorithms.

Keywords

Intrusion Detection System, Cyber Security, Machine Learning, Deep Learning, Deep auto Encoder

1. Introduction

With the increased usage and monumental growth of Internet and computers and also its ubiquitous nature, cyber-attacks are growing immensely in Information and Communications Technology (ICT) systems and networks. Sensitive user data dealt with ICT systems are subject to various attacks by adversaries. The adversaries may be from internal or external. The attacks created have been used as a major weapon and causes serious damage to the physical systems and significant financial loss to the user. These attacks can create either manually or by using machines intelligently. Cyber attacks are created increasingly using sophisticated algorithms with the development of advanced technologies. The severity and frequency of these attacks have been increasing day by day. Such attacks pose major security issue and need an effective, reliable and intelligent Intrusion Detection System (IDS). IDS protect the system from various attacks. IDS also help to discover and detect the abnormal events in the system.

Artificial neural networks (ANNs) are kind of machine learning algorithms inspired by the nervous system of human body. ANNs are composed of a few layers of neurons connected by adaptive weights, and the adjacent network layers are usually fully connected. The universal approximation theorem for ANNs states that every continuous function that maps intervals of real numbers to some output interval of real numbers can be approximated arbitrarily closely by a multi-layer perceptron (type of ANN) with just one hidden layer. This means that an ANN with one hidden layer is capable of producing any non-linear continuous function, and as such much of the early research on ANNs concentrated on networks with just one hidden layer, trained using back-propagation. Activation functions are used to propagate the output of one layer's nodes forward to the next layer. Loss functions quantify how close a given neural network is to the ideal toward which it is training. The learning rate affects the amount by which you adjust parameters during optimization in order to minimize the error of neural network's guesses. In this algorithm, the feature extraction is performed separately. This learning is called shallow learning. Another class of learning called Deep Learning (DL) in which the neural networks consist of multiple hidden layers. In DL, feature extraction is performed by the first few layers of the deep network.

Deep learning has garnered great attention among researchers in recent years in a variety of application domains. It is a sub field of machine learning growing rapidly in all the areas such as image processing, computer vision, speech recognition, machine translation, object detection, medical imaging, medical information processing, robotics and control, bioinformatics, natural language processing, cyber security, and many others[18]. Various deep learning methods are proposed for identifying security attacks. Deep belief networks are widely used in intrusion detection process [22]. The success of deep autoencoder in various classification problems motivated to introduce deep autoencoder for intrusion detection problem. The contributions of deep autoencoder are as follows:

1. To develop deep auto encoder (DAE) based intrusion detection model to classify normal and abnormal events.
2. The performance of the deep auto encoder is evaluated by NSL KDD data set.
3. The performance of DAE model is compared with classical machine learning algorithms.

The remainder of the paper is structured as follows. Section II discusses some of the most important related works. We briefly review the basics of machine learning and deep learning in Section III. Section IV presents the proposed DAE-based approach and algorithm for anomaly detection. Section V describes the features of NSL-KDD dataset. Section VI discusses the various performance metrics used to evaluate the performance of proposed system. Results and discussion are presented in Section VII. Finally, Section VIII provides the conclusion of DAE based intrusion detection system.

2. Related Works

Yin et al. [8] proposed a deep learning approach for intrusion detection using neural networks. The performance of this approach is evaluated for binary and multi-class intrusion classification on the NSL-KDD dataset by varying number of neurons and learning rate. The results are compared with ML based approaches such as J48, SVM and ANN etc. This approach has accuracy of 83.28% in KDD-Test and 68.55% in KDD-Test-21 for binary classification, slightly higher than traditional ML approaches on 80 hidden nodes and 0.1 learning rate and has accuracy of 81.29% in KDD-Test and 66.67% in KDD-Test-21 for binary classification, significantly higher than traditional ML approaches on 80 hidden nodes and 0.5 learning rate. The results shown that RNNs outperform the shallow machine learning algorithms. These methods require extensive computing resources and large number of neurons.

Zhang et al. [9] proposed a deep learning-based approach for network intrusion detection using denoising auto-encoder (DAE)with a weighted loss function for feature selection. This method mainly consists of two deep-learning based components to perform feature selection and classification. The selection is performed by a DAE, where a key processis to add weights to its loss function, which helps improve the selection results by placing more emphasis on the attack samples. The classification is then realized by an MLP with a minimized number of parameters while still achieving a highlevel of performance. The experiments are conducted using UNSW-NB dataset. Results show that the feature selection yields satisfactory detection performance with low memory and computing power requirements.

Farahnakian and Heikkonen [10] employed a denoising auto-encoder (DAE) model to discover important feature representations from the imbalanced training data and generate a model to detect normal and abnormal behaviors. This model is pre-trained using an unsupervised learning algorithm to avoid overfitting and local optima. A softmax classifier is added on the top of the model to represent the desired outputs. The performance of this DAE is evaluated byKDD-CUP'99 dataset. This method achieved detection accuracy of 94.71% on the total 10% KDDCUP99 test data set.

Shone et al. [11] presents a novel deep learning technique for intrusion detection which uses nonsymmetric deep autoencoder (NDAE) for unsupervised feature learning. Further, this model utilizes stacked NDAEs and the RF classification algorithm. This model is evaluated using GPU-enabled TensorFlow and obtained promising results from analysing the KDD Cup '99 and NSL-KDD datasets. The results demonstrated that this approach offers high levels of accuracy, precision and recall together with reduced training time.

Lin et al. [12] focused on network intrusion detection using convolutional neural networks (CNNs) based on LeNet-5 to classify the network threats. This method consists of three sub phases namely, the feature extraction phase, the model training phase, and the model verification phase. This method is evaluated using KDD Cup 99 dataset using TensorFlow tool.

Ye et al. [13] proposed a heterogeneous deep learning framework composed of an Autoencoder stacked up with multilayer restricted Boltzmann machines (RBMs) and a layer of associative memory for malware detection. This model employs a greedy layer-wise training operation for unsupervised feature learning in addition to supervised parameter tuning. Data is collected from Comodo Cloud Security Center to evaluate the performance of this model in malware detection. The performance of this approach is compared with shallow learning-based classification methods such as artificial neural network (ANN), support vector machine (SVM), Naive Bayes (NB), and decision tree (DT). The experimental results demonstrate that this approach outperforms all other methods.

Vinayakumar et al. [14] modeled a deep neural network to detect and classify unforeseen and unpredictable cyber-attacks. Both Network based intrusion detection systems (NIDS) and Host based intrusion detection systems (HIDS) are developed and experimented with different data sets like KDDCup 99, NSL-KDD, UNSW-NB15, Kyoto, WSN-DS and CICIDS 2017. In addition, a scalable hybrid intrusion detection framework called SHIA is introduced to process large amount of network level and host-level events to automatically identify malicious characteristics in order to provide appropriate alerts to the network admin.

Vinayakumar et al. [15] proposed deep learning architectures based on Static analysis, Dynamic analysis and image processing techniques formalware detection and designed a highly scalable framework called ScaleMalNet to detect, classify and categorize zerodaymalwares. They have used two data sets for conducting experiments. In the first stage, malwares are detected using static and dynamic analysis. In the second stage, malware is categorized into their corresponding categories. The performance of deep learning method is compared with machine learning approach. The performances obtained by deep learning architectures outperformed the classical machine learning approaches.

Naseer et al. [16] proposed anomaly detection models based on the Deep Neural Network architectures convolutional neural networks, autoencoders and recurrent neural networks. These models are trained using NSL-KDD training dataset and tested using NSLKDDTest+ and NSLKDDTest21. This model is implemented using GPU powered test-bed using Keras with Theano backend. The performance of this DNN model is compared with conventional machine learning models. Experimental results represents that the DCNN and LSTM achieved an accuracy of 85% and 89% on test dataset.

3. Machine Learning and Deep Learning

Machine Learning is a subset of Artificial Intelligence, where machines learn to perform tasks without being programmed explicitly. In this the machine can learn from data and experience. It uses statistical methods that enable machines to improve with experience. ML mainly focuses on classification and prediction based on known values previously learned from the training data. Machine learning approach consists of two phases namely training and testing. Typically, the following steps are performed in machine learning process

- Collection of data
- Identification of features and its class from training data
- Feature Selection
- Learn the model using training data
- Make predictions

Machine learning can be categorized into

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

In supervised learning, the training data is already labeled by a human expert or through other means. Supervised learning includes classification algorithms and regression algorithms [1]. Various supervised learning algorithms are Naïve Bayes, Logistic Regression, Support Vector Machines, Random Forest, Hidden Markov Models, Fuzzy systems, Decision trees, K-Nearest Neighbour and Neural networks.

In unsupervised learning, the training data is neither classified nor labeled. The unsupervised learning algorithms are clustering and association.

In reinforcement learning the training data is intermediate between supervised and unsupervised learning In this method, an agent learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly. The agent learns without intervention from a human by maximizing its reward and minimizing its penalty.

Machine learning is having a significant effect on various fields of technology, commerce and science [2]. ML can be applied in health care, robotics, natural language processing, computer vision, recommendation systems, manufacturing, education, financial modeling, policing, and marketing.

Deep learning (DL) is a new research field in machine learning. It uses the concept of neural networks and mimics the human brain mechanism to interpret the data. Deep learning is a subset of machine learning, which is a subfield of AI. Deep learning is a machine learning algorithm with more than two layers of neural networks, useful for modelling complex concepts and relationships. Automatic feature extraction is the main advantage of deep learning compared with traditional machine learning algorithms. Deep learning reduces the human effort to perform many tasks [1]. Deep learning achieved disruptive results in various fields.

DL allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [3]. Here higher higher-level concepts are defined from lower level-concepts. Feature extraction is performed by lower layers of deep neural networks. But in machine learning, feature extraction is performed separately. DL works based on the principle of gradient-based optimization algorithms to fine-tune parameters in a multilayered network based on error at its output. The main advantage of deep-learning is the capability of automatically discovering the representations needed for detection or classification. But DL needs high computational power to train the models [6]. The main difference between deep neural networks and other neural networks are the number of layers. Deep Neural Networks (DNN) have multiple hidden layers where as ordinary neural networks have at most one hidden layer [4]. In deep learning, the process of learning does not require on features that are crafted by human [17]. Deng [19] classified deep learning into three categories namely, generative, discriminative and hybrid architectures. The main goal of deep learning is learning features from lower level to higher level. Figure1 represents the classification of given input with respect to machine learning and deep learning.

DNNs are composed of an input layer followed by many hidden layers and an output layer. The input layer represents the input which is to be classified. The output layer is responsible for producing the class of corresponding given input [5]. Figure 1.1 illustrates the structure of DNN with two hidden layers. The information from one layer is transformed to another layer in forward direction. Neurons in each layer are fully connected. Each node in the model acting as a neuron, receives output from previous layer. Each hidden layer uses the non-linear activation function for its computation. The activation function of each hidden layer is represented as

$$h_i(x) = f(w_i^T(x) + b_i) \qquad (1)$$

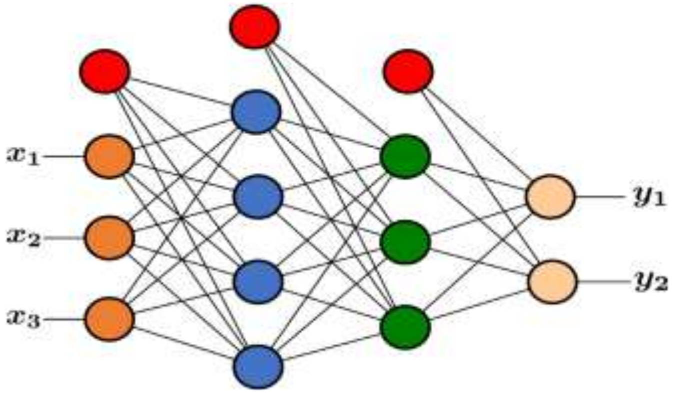

Figure 1.1 Structure of DNN

Machine learning algorithms works well when the data set contains less number (hundreds) of features.
If the data is unstructured, the size of the data is huge and contains very large number of features usage of machine learning is unfeasible. Deep learning is applied on the following scenarios

- Absence of human expert
- The nature of the problem is dynamic
- The nature of problem is very complex

- The problems in which the humans are unable to involve

Deep Learning Algorithms

Deep learning algorithms work with several "layers" of neural network algorithms, each of which passes a simplified representation of the data to the succeeding layer. The algorithms available in deep learning are based on deep neural networks. These algorithms are categorized in to three classes, namely supervised DL algorithm, unsupervised DL algorithm and deep reinforcement learning [30].

- **Supervised DL algorithms**
    - Fully connected Deep Neural Networks (FNN)

      Deep Neural Networks consists of directed graph composed of nodes and edges. FNN passes information from one node to another without forming a cycle. In this network, every neuron is connected to all the neurons in the previous layer
    - Convolutional Deep Neural Networks (CNN)

      Convolutional Deep Neural Networks is a variant of classical deep neural network in which each neuron receives its input only from a subset ofneurons of the previous layer. The CNN architecture consists of three layers namely convolution layers, pooling layers and classification layer. CNN efficiently works in image processing.
    - Recurrent Deep Neural Networks (RNN)

      This network uses the feature of back propagation. This network is mainly used in sequence input. RNNs are best suited for natural language and speech processing. Long Short-Term Memory (LSTM) is a variant of RNN.
- **Unsupervised DL algorithms**
    - Deep Belief Networks (DBN)

      These networks are modeled as a stack of Restricted Boltzmann Machines [24, 25] with classifier as a last layer. It is mainly used in pre-training stage because it performs well for feature extraction.
    - Deep Auto-Encoders (DAE)

      Deep Auto-Encoders consists of multiple autoencoders where the input and output consist of same number of neurons. The auto encoder consists of encoder, code and decoder. They are mainly used for dimensionality reduction.
    - Generative Adversarial Networks (GAN)

      In GAN, two neural networks compete against each other in a zero-sum game. Here, one network acts as a generator and another network act as a discriminator. The generator takes in input data and generates output data with the same characteristics as real data. The discriminator takes in real data and data from the generator and tries to distinguish whether the input is real or fake. When training has finished, the generator is capable of generating new data that is not distinguishable
- **Deep Reinforcement Learning**

  Learning by trial-and-error, exclusively from rewards and punishments is known as reinforcement learning. It takes best actions from past experience. In deep reinforcement learning, agents construct and learn their own knowledge directly from raw inputs, such as vision, without any hand-engineered features or domain heuristics.

4. Proposed Methodologies

Deep Learning provide a new approach for handling various security issues [21]. Deep Auto Encoder is a promising approach in intrusion detection system.

Auto Encoders

An Auto-Encoder (AE) is a type of neural networks with the same number of neurons in both input and output layer [28]. It is mainly used for dimensionality reduction for better representation of data. Auto-Encoder is an unsupervised learning model and applies backpropagation. The nodes in the hidden layer represent features which are low-dimensional [20]. An AE consists of encoder and decoder. Encoder compress the input data into a low-dimensional representation and decoder reconstruct the information from low-dimension representation. The conceptual structure of AE is represented in Fig.4.1.

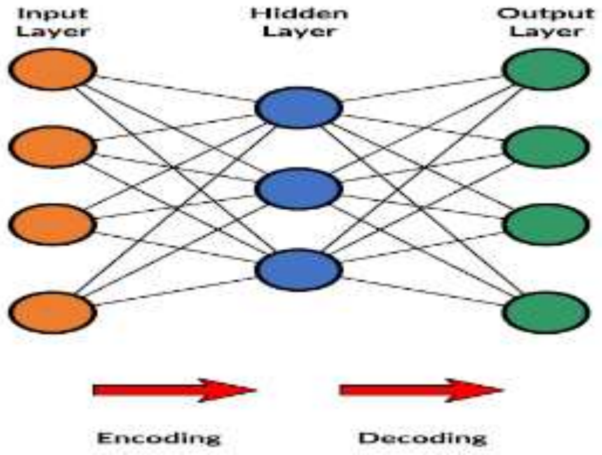

Figure 4.1. Conceptual structure of an AE

The input and output layer consist of N nodes and hidden layer consist of K nodes. Hidden layer of AE is known as abstract layer. For a given training data set X with m samples, the encoder performs the mapping of input vector to hidden vector using mapping function *f*.

$$h = f(Wx+b)$$

where f is an activation function, W is a weight matrix and b is the bias vector.

The decoder maps the h to x'

$$x' = f'(W'h + b')$$

Where f', W' and b' are the parameters for the reconstruction of x. AEs are trained to minimize the difference between input and output.

There are several extensions of AE namely, Stacked AE, Sparse AE and Denoising AE. A Stacked Auto-Encoder (SAE) consisting of multiple layers of AE in which the output value of each layer is connected to the inputs of the next layer to gradually compress the information more and more to construct a new representation [26]. In sparse AE the hidden units have sparsity constraints and only a portion of the hidden units are activated at a given time. Denoising auto-encoder reconstructs the original data from the corrupted input, which helps to discover the robust representations and prevent it from learning the less important identity. In DAE input is a corrupted with noise [27]. The proposed approach uses deep auto-encoder (DAE) for intrusion detection. Intrusion detection system consists of two phases training and testing. In the training phase, the training data set is used to create a DAE model. Then the performance of this model is evaluated using test data. In DAE loss function is used for detecting intrusions.

Loss function $$L(x,y) = \frac{1}{m}\sum_{i=1}^{m} ||x_i - y_i||^2$$

The hidden layer 1 of DAE takes features from input data. The output of hidden layer h is used as an input of hidden layer h + 1. Based on the layer-wise algorithm [14], an auto-encoder at layer h + 1is trained after the completion of training an auto-encoder at layer h. The last hidden layer is a supervised layer which classifies attack classes by using the softmax classifier. Finally, the output layer is used as an output of the entire DAE model.

Algorithm1: Training Deep Autoencoder

```
Input: Dataset D= {$x_1$, $x_2$, …....$x_m$} with $m$ samples, number of hidden layers $L$
Output: Output of each hidden unit
Step 1: for $l \in [1, L]$ do
Step 2: initialize $W_l= 0$, $W_l^{'}= 0$, $b_l= 0$, $b_l^{'}= 0$
Step 3: define the $l$-th hidden layer representation vector $h_l= f(W_l\, h_{l-1}+ b_l)$
Step 4: define the $l$-th hidden layer output $x_l{'}= f( W_l{'}h_l+b_l)$
Step 5: while not stopping criterion do
Step 6:         calculate $h_l$ from $h_{l-1}$
Step 7:         calculate $yl$
Step 8:         calculate the loss function
Step 9:         update layer parameters $\theta_l= (W_l, b_l)$ and $\theta_l{'}= (W_l^{'}, b_l^{'})$
Step 10: end while
Step 11: end for
Step 12: Initialize $(W_{l+1}, b_{l+1})$ at the supervised layer
Step 13: calculate the labels for each sample $xi$ of the training dataset $D$
Step 14: perform BP in a supervised way to tune parameter of all layers;
```

5. Experiment

Data Set

NSL-KDD is the refined version of KDDCup99 intrusion data. NSL-KDD is derived from KDD Cup 99 dataset [23]. It is generated in the year 2009 and widely used data set for network intrusion detection. Many researchers use NSL-KDD as the benchmark dataset which deal with inherent problems of the KDD Cup 1999 dataset which contain too many redundant records. It does not include any redundant and duplicate records. It partitions all the records in the KDD Cup dataset into various difficulty levels based on the number of learning algorithms that can correctly classify there cords. The number of records in the training and test sets is reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion [7].

It contains essential records of the complete KDD data set. NSL-KDD dataset includes three sub-files and others listed in table: KDDTrain+, KDDTest+ and KDDTest–21which has different normal records and four different types of attack records. There are 125,973 network traffic samples in the KDDTrain+ dataset, 22,554 network traffic samples in the KDDTest+ dataset and 11850 network traffic samples in the KDDTest–21 dataset. The KDDTest−21 dataset is a subset of the KDDTest+ and is more difficult to classify [29]. There are 41 features and 1 class label for each traffic record. The features include basic features (No.1-No.10), content features (No.11 - No.22), and traffic features (No.23 - No.41). A class label was provided with each record, which identified the network traffic instance either as normal or an attack. The details of training and testing records are reported in table1.

TABLE 1. Features of NSL-KDD dataset.

Attack Category	Description	KDDTrain+	KDDTest+	KDDTest–21
Normal	Connections are normal	67,343	9,710	2152
DoS	Denying access to unauthorized users by making network resources down	45,927	7,458	4342
Probe	Obtaining system and network configuration details	11,656	2,422	2402
R2L	Illegal access from remote computer	995	2,754	2754
U2R	Obtaining the access of super user	52	200	200
Total		**125,973**	**22,544**	**11,850**

TABLE 2. Features of NSL-KDD dataset.

No.	Features	Types
1.	Duration	Numeric
2.	Protocol_type	Nominal
3.	Service	Nominal
4.	Flag	Nominal
5.	src_bytes	Numeric
6.	dst_bytes	Numeric
7.	Land	Binary
8.	wrong_fragment	Numeric
9.	urgent	Numeric
10.	hot	Numeric
11.	num_failed_logins	Numeric
12.	logged_in	Binary
13.	num_compromised	Numeric
14.	root_shell	Binary
15.	su_attempted	Binary
16.	num_root	Numeric
17.	num_file_creations	Numeric
18.	num_shells	Numeric

19.	num_access_files	Numeric
20.	num_outbound_cmds	Numeric
21.	is_host_login	Binary
22.	is_guest_login	Binary
23.	count	Numeric
24.	srv_count	Numeric
25.	serror_rate	Numeric
26.	srv_serror_rate	Numeric
27.	rerror_rate	Numeric
28.	srv_rerror_rate	Numeric
29.	same_srv_rate	Numeric
30.	diff_srv_rate	Numeric
31.	srv_diff_host_rate	Numeric
32.	dst_host_count	Numeric
33.	dst_host_srv_count	Numeric
34.	dst_host_same_srv_rate	Numeric
35.	dst_host_diff_srv_rate	Numeric
36.	dst_host_same_src_port_rate	Numeric
37.	dst_host_srv_diff_host_rate	Numeric
38.	dst_host_serror_rate	Numeric
39.	dst_host_srv_serror_rate	Numeric
40.	dst_host_rerror_rate	Numeric
41.	dst_host_srv_rerror_rate	Numeric

6. Performance Metrics

In order to evaluate the performance of the proposed system Accuracy, Precision, Recall, False Alarm and F- score is considered as a performance metrics. These metrics are calculated based on True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

True Positive (TP) - Number of anomaly records that are correctly identified as anomaly

False Positive (FP) - Number of normal records that are incorrectly identified as anomaly.

True Negative (TN) - Number of normal records that are correctly identified as normal.

False Negative (FN) - Number of anomaly records that are incorrectly identified as normal.

The metrics Accuracy, Precision, Recall, False Alarm and F- score are defined as follows

Accuracy estimates the ratio of the correctly recognized connection records to the entire test dataset.

If the accuracy is higher, the machine learning model is better.

$$Accuracy = \frac{\#TP + \#TN}{\#TP + \#TN + \#FP + FN} \qquad (1)$$

Precision estimates the ratio of the correctly identified attack connection records to the number of all identified attack connection records. If the Precision is higher, the machine learning model is better

$$Precision = \frac{\#TP}{\#TP + \#FP} \qquad (2)$$

The recall measures the ratio of correct classifications penalized by the number of missed entries.

$$Recall = \frac{\#TP}{\#TP + \#FN} (3)$$

The false alarm rate measures the proportion of benign events incorrectly classified as malicious.

$$False\ Alarm\ Rate = \frac{\#FP}{\#FP + \#TN} (4)$$

The F-score measures the harmonic mean of precision and recall, which serves as a derived effectiveness measurement.

$$F - score = 2\text{ X } \left(\frac{Precision\ X\ Recall}{Precision + Recall}\right) \qquad (5)$$

7. Results and Discussion

The proposed research uses Python TensorFlow toolkit for implementing deep auto encoders. The conventional machine learning algorithms are implemented using Scikit-learn machine learning library. The performance of Deep auto encoder is compared with classical machine learning algorithms namely Logistic Regression (LR), Naïve Bayes (NB), K-Nearest Neighbor (KNN), Decision Tree (DT) and Random Forest (RF). In this approach, KDDTrain+ is used as training set and KDDTest+ and KDDTest-21 used as testing test.

Table 3. Accuracy for each model

	LR	**NB**	**KNN**	**DT**	**RF**	**DAE**
KDDTest+	81.51	81.93	89.57	91.66	93.70	95.26
KDDTest–21	79.36	78.34	85.64	87.68	88.41	94.38

Table 4. Precision for each model

	LR	**NB**	**KNN**	**DT**	**RF**	**DAE**
KDDTest+	83.24%	82.32%	91.75%	92.17%	91.78%	95.46%
KDDTest–21	80.35%	79.57%	87.44%	91.28%	89.74%	95.00%

Table 5. Recall for each model

	LR	**NB**	**KNN**	**DT**	**RF**	**DAE**
KDDTest+	81.34%	84.58%	86.01%	87.56 %	81.17%	90.58%
KDDTest–21	79.08%	82.11%	82.98%	82.36%	79.08%	89.74%

Table 6. False Alarm rate for each model

	LR	NB	KNN	DT	RF	DAE
KDDTest+	29.89%	26.62%	24.12%	22.45%	11.04%	12.98%
KDDTest–21	32.58%	30.62%	25.06%	24.17%	13.63%	15.77%

Table 7. F-Score for each model

	LR	NB	KNN	DT	RF	DAE
KDDTest+	80.34%	83.58%	91.14%	91.74%	91.47%	95.56%
KDDTest–21	83.75%	82.11%	82.98%	82.36%	79.08%	93.76%

Table 3,4,5,6 and 7 shows the comparison of experiment results. From table 3, it is observed that the overall accuracy of deep auto encoder is better compared with other methods in KDDTest+ and KDDTest-21. It is also observed that the random forest method outperforms the other conventional results. In all the cases, the accuracy of KDDTest+ is high compared with KDDTest-21. The table 4 indicates the precision of each model. From the results it is inferred that, the precision of DAE is improved by 4% compared with other methods. Table5 demonstrates the improvement of recall in DAE method. The results shown that 10% improvement in the performance of recall.

Table 6 shows the results of false alarm rate. The results demonstrate that the values of Random forest and deep auto encoder are less in comparing with the results of remaining methods. The false alarm rate of DAE is high in comparison with RF. The table 7 shows the result of F-score. From the results, it is observed that Logistic regression method performs better in KDDTest-21 dataset compared with KDDTest-21 dataset. The overall results show the performance improvement in proposed deep auto encoder method.

The figure 7.1 shows the comparison of Accuracy, Precision, Recall and F-Score of KDDTest+. The figure indicates that the performance of KNN, DT and RF is better compared with LR and NB. By comparing the performance of DAE and other machine learning algorithms with respect to KDDTest+, DAE outperforms compared with classical algorithms. The figure 7.2 shows the comparison of Accuracy, Precision, Recall and F-Score of KDDTest-21. From the figure, we infer that the performance measures of KDDTest-21 using all the techniques are not good compared with KDDTest+. The figure 7.3 shows the comparison of false alarm rate between KDDTest+ and KDDTest-21. The figure demonstrates that the false alarm rate of KDDTest+ is less compared with KDDTest-21. The figure also represents that, the false alarm rate of Random Forest method is less comparing with Deep autoencoders and other machine learning techniques. The LR method gives high false alarm rate of 32.58% with KDDTest-21data set.

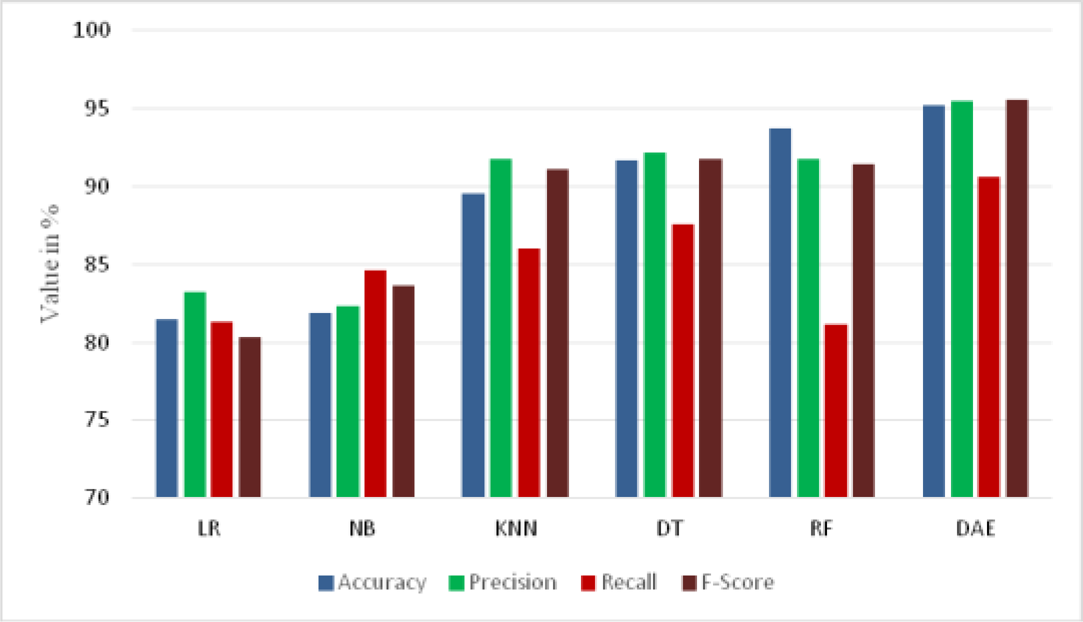

Figure 7.1 Accuracy, Precision, Recall and F-Score of KDDTest+

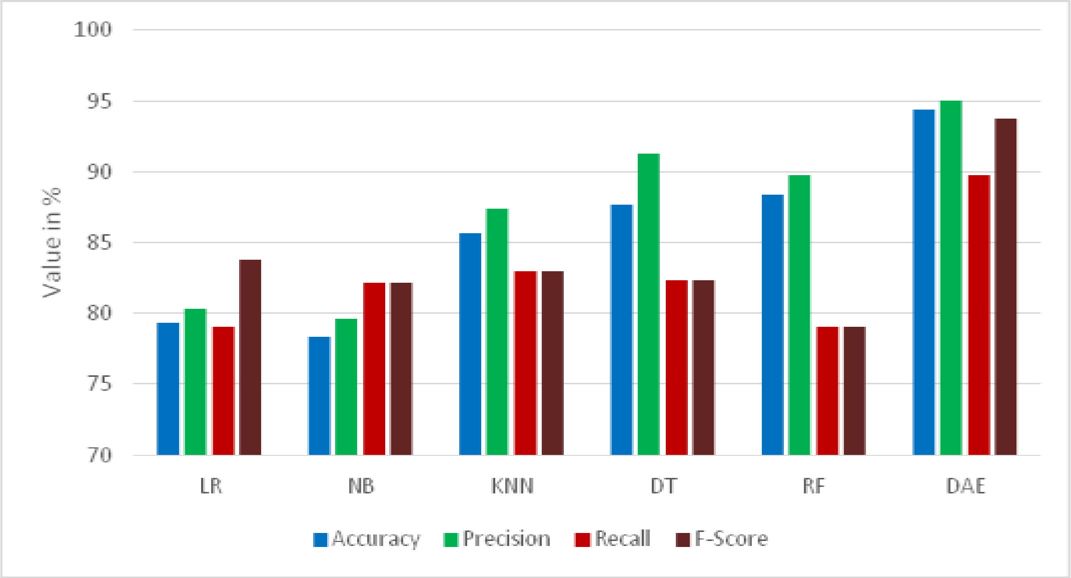

Figure 7.2 Accuracy, Precision, Recall and F-Score of KDDTest-21

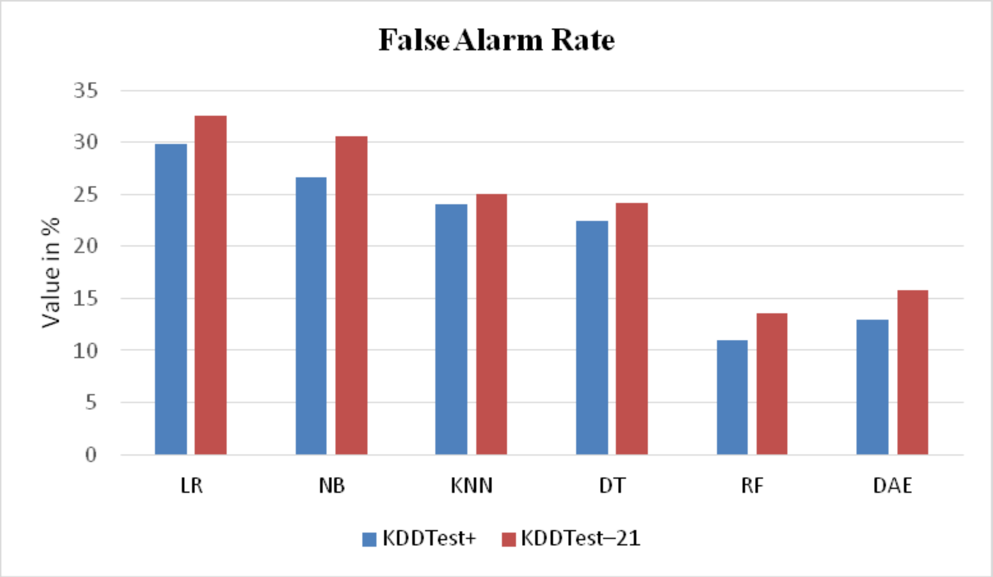

Figure 7.3 False alarm rate of KDDTest+ and KDDTest-21

8. Conclusion

Developing effective intrusion detection systems attracted the researchers in recent years due to the massive evolution of network attacks. In this paper, a deep auto-encoder based intrusion detection system has been proposed and its performance is compared with classical machine learning algorithms on the NSL-KDD dataset. This model can effectively increase the accuracy, precision, recall and F-score of intrusion detection and reduces the false alarm. In the future research, various deep learning methods such as Convolutional Neural Networks, Recurrent Neural Networks, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are used to improve the accuracy of intrusion detection system.

REFERENCES

1. P. Louridas and C. Ebert, “Machine Learning,” IEEE Software., vol. 33, no. 5, pp. 110–115, 2016.
2. M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” Science, vol. 349, no. 6245, pp. 255–260, 2015.
3. Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
4. Deng, L.; Yu, D. Deep learning: Methods and applications. Found. Trends Signal Process. 2014, 7, 197–387.
5. W. Huang, J.W. Stokes, MtNet: A multi-task neural network for dynamic malware classification in: Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2016, pp. 399–418.
6. Adam Gibson, Josh Patterson, Deep Learning Practitioner’s Approach, O’REILLY, 2017.
7. S. Revathi, A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, Int. J. Eng. Res. Technol. ESRSA Publ. 2 (2013) 1848–1854.
8. C. Yin, Y.Zhu, J.Fei and X.He "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Network," IEEE Access, vol. 5, pp. 21954-21961, 2017.
9. H.Zhang, Q. Wu, S.Gao, Z.Wang, Y.Xu and Y.Liu, "An Effective Deep Learning Based Scheme for Network Intrusion Detection," in 2018 24th International Conference on Pattern Recognition(ICPR), Bejing,China, 2018.
10. F. Farahnakian, J. Heikkonen, A deep auto-encoder based approach for intrusion detection system, in: Int. Conf. on Advanced Communication Technology (ICACT), 2018, pp. 178–183.

11. Nathan Shone , Tran Nguyen Ngoc, Vu Dinh Phai and Qi Shi , A Deep Learning Approach to Network Intrusion Detection, IEEE Transactions on Emerging topics in Computational Intelligence, vol. 2, no. 1, February 2018.
12. W. Lin, H. Lin, P. Wang, B. Wu, J. Tsai, Using convolutional neural networks to network intrusion detection for cyber threats, in: IEEE Int.Conf. on Applied Syst. Invention (ICASI), 2018, pp. 1107–1110,
13. Y. Ye , L. Chen , S. Hou , W. Hardy , X. Li , DeepAM: A heterogeneous deep learning framework for intelligent malware detection, Knowledge and Information Systems 54 (2) (2018) 265–285 .
14. Vinayakumar, R., AlazabMamoun, Soman, K. P., Poornachandran Prabaharan, Al-Nemrat, A. and Venkatraman Sitalakshmi. Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access. vol.7, pp. 41525-41550, 2019.
15. Vinayakumar, R., AlazabMamoun, Soman, K. P., Poornachandran Prabaharan, and Venkatraman Sitalakshmi, Robust Intelligent Malware Detection Using Deep Learning, IEEE Access. vol.7, pp. 46717-46738, 2019.
16. Sheraz naseer, yasir saleem, shehzadkhalid, Muhammad khawarbashir,Jihunhan, Muhammad munwariqbal, and kijunhan, Enhanced Network Anomaly Detection Based on Deep Neural Networks, IEEE Access, vol.6, pp. 48231 – 48246, 2017.
17. B. Dong, X. Wang, Comparison deep learning method to traditional methods using for network intrusion detection, in: 8th IEEE International Conference on Communication Software and Networks, 2016, pp. 581–585.
18. K. Kim, M.E. Aminanto, Deep learning in intrusion detection perspective: overview and further challenges, International Workshop on Big Data and Information Security (IWBIS) (2017), pp.5–10.
19. L. Deng, “A tutorial survey of architectures, algorithms, and applications for deep learning,” APSIPA Transactions on Signal and Information Processing, vol. 3, pp.1-29, 2014.
20. Geoffrey E. Hinton, Ruslan R. Salakhutdino, Reducing the dimensionality of data with neural network, Science, 313 (2006) 504–507.
21. A. Mahdavifar and AA Ghorbani, Application of deep learning to cyber security: A survey, Neurocomputing, 347, pp.149-176, 2019.
22. N. Gao, L. Gao, Q. Gao, H. Wang, An intrusion detection model based on deep belief networks, in: Proceedings of the 2nd International Conference on Advanced Cloud and Big Data (CBD), IEEE, 2014, pp. 247–252.
23. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, A deep learning approach for network intrusion detection system, in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies, 2016, pp. 21–26.
24. J. Yang, J. Deng, S. Li, Y. Hao, Improved traffic detection with support vector machine based on restricted boltzmann machine, Soft Computing, Vol. 21, No.11. pp. 3101–3112, 2017.
25. U. Fiore, F. Palmieri, A. Castiglione, A. De Santis, Network anomaly detection with the restricted boltzmann machine, Neurocomputing 122 (2013) 13–23.
26. Berman, D. S., Buczak, A. L., Chavis, J. S., and Corbett, C. L. A survey of deep learning methods for cyber security. Information, vol.10, no.4, pp.1-35, 2019.
27. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol,Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of Machine Learning Research, vol. 11, no. 12, pp. 3371-3408, 2010.
28. P. Madani and N. Vlajic, Robustness of deep autoencoder in intrusion detection under adversarial contamination, in Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security, no.1, pp. 1-8, 2018.
29. Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, C. Wang, Machine learning and deep learning methods for cybersecurity, IEEE Access, vol.6, pp. 35365-35381, 2018.
30. Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M. On the effectiveness of machine and deep learning for cyber security. IEEE International Conference on Cyber Conflict (CyCon), pp. 371-390, 2018.