Design of a Network Intrusion Detection System using Complex Deep Neuronal Networks

M. AL-Shabi[1]

[1]Department of Management Information System, College of Business Administration, Taibah University, Saudi Arabia.

***Abstract*:** Recent years have witnessed a tremendous development in various scientific and industrial fields. As a result, different types of networks are widely introduced which are vulnerable to intrusion. In view of the same, numerous studies have been devoted to detecting all types of intrusion and protect the networks from these penetrations. In this paper, a novel network intrusion detection system has been designed to detect cyber-attacks using complex deep neuronal networks. The developed system is trained and tested on the standard dataset KDDCUP99 via pycharm program. Relevant to existing intrusion detection methods with similar deep neuronal networks and traditional machine learning algorithms, the proposed detection system achieves better results in terms of detection accuracy.

***Keywords*:** Cyber security, Intrusion detection system, Complex deep neural networks, Deep learning, KDD CUP99 dataset, Computer network**.**

1. Introduction

Information and communications technology (ICT) systems deal with different user data that are vulnerable to various manual/automated attacks from internal and external hackers [21]. These attacks are diverse and constantly evolving with the advances in hardware/software and network architectures. Due to malicious cyber-attacks, serious security issues have raised that indeed require a flexible and reliable intrusion detection system (IDS) [21]. IDS is well-known technology utilized to detect internal/external interferences and anomalies that target network systems. The IDS system includes a set of tools and mechanisms for monitoring computer system and network traffic. In the last three years, deep learning methods have been extensively investigated and various machine learning approaches are introduced to detect anomaly-based paralysis. However, with the emergence of new and more complex attack scenarios, methods based on machine learning become no longer effective in dealing with the security challenges. On the other hand, deep learning techniques have shown their effectiveness in feature extraction and classification tasks. Besides, deep networks can automatically reduce the complexity of network traffic by finding the data correlation without human intervention. They also contribute to reducing the rate of type positives and increasing the detection rate in anomaly detection systems [3].

Considering the above discussion, this paper aims to design an intrusion detection system in computer networks and ICT systems, based on complex deep neural networks in order to obtain a better detection process. The main objective of this paper is securing networks and ICT systems and protecting these systems from intrusions and cyber-attacks. Besides, it aims to reducing the losses resulting from electronic attacks and malicious software. For this purpose, intrusion detection systems, complex deep neural networks, and database are employed in this study. Specifically, the pycharm program and its libraries in python language are utilized for programming the proposed solution. Moreover, architectural dataset and KDD CUP9 are employed to train and test the proposed model.

The remainder of this paper is organized as follows. Related studies on cybersecurity using AI, including NIDS and HIDS, are extensively reviewed and analyzed in Section 2. In Section 3, the proposed system is described, which includes the complex deep learning network, dataset pre-processing steps, and ordinary deep learning networks. Furthermore, the implementation and discussion of the experimental results are presented in Section 4. Finally, the paper is concluded in Section 5.

2. Related Works

Since the advent of computer architectures, there have been ongoing studies on security concerns including network-based intrusion detection system (NIDS) and host-dependent intrusion detection system (HIDS). Recent days have witnessed a remarkable interest among security researchers and specialists for developing several solutions based on machine learning to NIDS and HIDS. A survey on existing machine learning based solutions is presented in [12]. In this section, a panorama of largest study up to now is discussed, in which the machine learning and deep learning approaches that are applied to boost NIDS and HIDS, are investigated.

2.1 Cyber Security and Intrusion Detection System

Relative to information security, cyber security represents a broader concept as it includes securing data and information exchanged through internal or external networks. These are generally stored in servers inside/outside the company away from intrusions [8]. On the other hand, intrusion detection system (IDS) refers to a computer security program, application, or a combination of both. It aims to detect a wide range of security breaches by monitoring the systems and networks against any malicious activity [15]. The main functions of IDSs are to monitor straits and networks, analyze the behavior of computer systems, generate alerts, and respond to suspicious behavior. IDs are usually published near the protected network [21] [12].

The main components of the intrusion detection system are illustrated in Figure 1. Based on the information sources, the intrusion detection is categorized into a network-based intrusion detection system (NIDS) and host- dependent intrusion detection system (HIDS). In HIDS, log files are collected via local sensors [16]. On the other hand, the contents of each packet in the network traffic packets is inspected by NIDS [5].

Abuse detection uses pre-defined signatures and filters to detect the attacks and it relies on constantly updating the signature database. This method is accurate for known attacks.

However, it is no longer effective for unknown attacks. On the other hand, anomaly detection uses particular mechanisms to detect unknown malicious activities. It is worth mentioning that abnormality detection generally results in a high false positive rate [21].

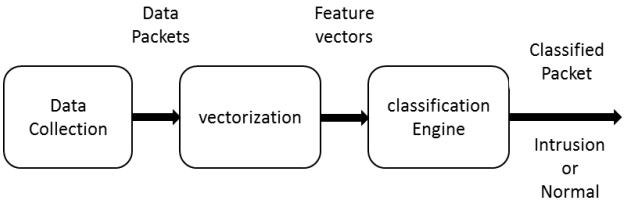

Figure 1. Main Components of Intrusion Detection System

2.2 Deep learning

Deep learning models consist of various deep networks, i.e., moderated, and unsupervised networks. Some examples of moderated networks include the deep neural networks (DNN), deep brief networks (DBNs), convolutional neural networks (CCNs) and recursive neural networks (RNNs). On the other hand, auto encoders, restricted Boltzmann machines (RBMs) and generative adversarial networks (GANs) represent examples of unsupervised networks (See [11] [10] [9] [18] for more details).

Table 1. Comparison of various deep learning models

Algorithms	Suitable Data Types	Supervised/ Unsupervised	Functions
Auto Encoder	Raw data; Feature vectors	Unsupervised	Feature extraction; Feature reduction; Denoising
RBM	Feature vectors	Unsupervised	Feature extraction; Feature reduction; Denoising
DBN	Feature vectors	Supervised	Feature extraction; Classification
DNN	Feature vectors	Supervised	Feature extraction; Classification
CNN	Raw data; Feature vectors; Matrices	Supervised	Feature extraction; Classification
RNN	*Raw data; Feature vectors; Sequence data*	*Supervised*	*Feature extraction; Classification*
GAN	*Raw data; Feature vectors*	*Unsupervised*	*Data augmentation; Adversarial training*

Deep learning models can directly learn the feature representations from the original data, such as images and text, without the need for manual feature engineering. Hence, machine learning methods can be implemented in a comprehensive manner. For a large data set, deep learning methods have a huge advantage in dealing with such scenarios. In view of deep learning, the main focus is on network engineering, hyperparameter selection, and optimization strategy. A comparison of different deep learning algorithms [15] is presented in Table 1.

2.3. Deep learning-based intrusion detection system

The detection system based on deep learning [2] is illustrated in Figure 2. It consists of several stages that are discussed in the following:

1. Income or traffic: Data is the primary component when evaluating any IDS. Data can be collected from various sources, including host logs, application data and network traffic.
2. Pre-processing or preparatory data: This is useful in removing redundant data, incomplete data, and converting the data into a standardized form. It usually covers the deleting of duplicate records and converting symbolic data into numeric data.
3. Feature extraction: It includes the analyzing of network traffic using specific tools that are used to create data sets, such as Argus.
4. Intrusion Detection Model: This model is designed using a deep learning algorithm, and then trained and tested in order to determine the type of connection log (either normal log or Anomaly).

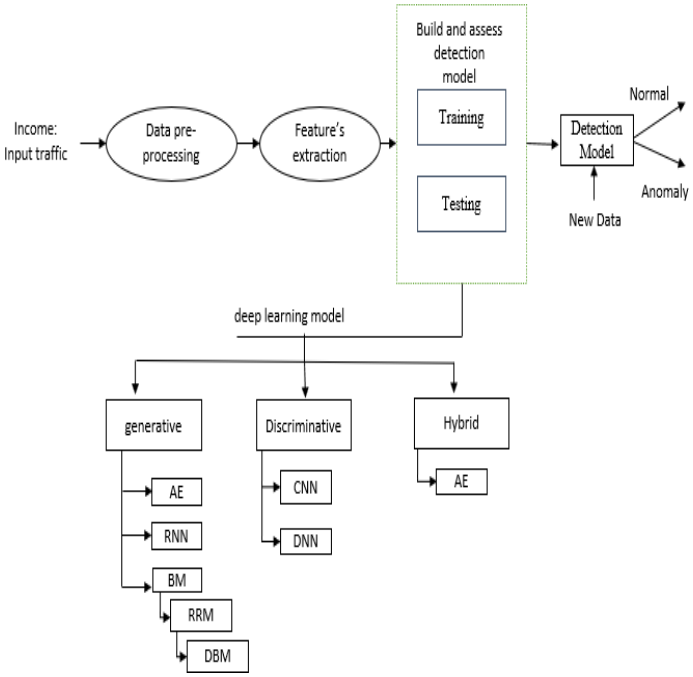

Figure 2. Deep learning-based IDS Architecture

The effectiveness of an IDS is evaluated by its classification capabilities, i.e., the ability to correctly determine to which class the contact record belongs (either normal or anomaly). Four different cases are observed when comparing the classification result of the record with the actual reality. These cases are presented in Table 2. It expresses the known disturbance matrix which is one of the most important means used in the process of performance evaluation of IDS [12].

Table 2. Confusion Matrix

	Predicted Positive	Predicted Negative
Active Positive	TP	FN
Actual Negative	FP	TN

In the following, the most utilized measures in the evaluation process are presented.

Accuracy: It estimates the ratio of correctly recognized contact records to the entire test data set, as given in (1).

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

where:

- TP (true positive) represents the number of connection records correctly classified to the normal class.

- TN (true negative) represents the number of connection records correctly classified to the attack class.
- FP (false positive) represents the number of normal connection records wrongly classified to the attack connection record.
- FN (false negative) represents the number of attack connection records wrongly classified to the normal connection record [1].

Precision: It estimates the ratio of the correctly identified attack connection records to the number of all identified attack connection records, as given in (2).

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

Where TP is the True Positive and FP is False positive.

True Positive Rate (TPR): It is also known as Recall. It estimates the ratio of the correctly classified attack connection records to the total number of attack connection records, as defined in (3).

$$TPR = \frac{TP}{TP+FN} \quad (3)$$

False Positive Rate (FPR): It estimates the ratio of the normal connection records flagged as attacks to the total number of normal connection records. The FPR is also known as false alarm rate, and it is calculated as in (4).

$$FPR = \frac{FP}{TP+FP} \quad (4)$$

F1-Score (Also known as F-measure):It is defined as the harmonic average of the precision and the recall, as given in (5).

$$F1 - Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right) \quad (5)$$

2.4. KDD CUP99 dataset

This dataset was designed as a 1998 simulation dataset using 1,000 UNIX machines and 100 access users [13]. It is created by MIT Lincon Lab due to the need for a suitable dataset for testing intrusion detection systems. In general, the KDDCUP99 standard dataset includes about 5 million connection records that are divided into training and test records. Each connection record includes 41 features that can be categorized as follows: The first 9 features are basic features of a packet, 10-22 are content features, 23-31 are traffic features, and 32-41 are host-based features. The records in this dataset can be categorized into 5 main categories (4 are attack and 1 is normal/non-attack type data). The attacks are 22 types, and each one belongs to the following attack category: DoS (Denial of Service), Probe (Probe Attacks), R2L (Remote to Local) and U2R (User to Root) [14].

KDD CUP99 contains numeric data (in binary and real number format) and text information (characters) about the requirement classes. Additionally, this data has one additional feature at the end in order to show the label of the data (whether it is from intrusion or not) [15] [3]. This database is deeply rooted and detailed in Reference [16].

Natural Record:

0,tcp,http,SF,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.0

Record Attack:

0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,292,18,1.00,1.00,0.00,0.00,0.06,0.05,0

The training connection records with the KDD CUP99 dataset and their test are summarized in Table 3.

Table 3. Training and Testing connection records from KDDCup 99 and NSL-KDD datasets

Attack Category	Description	Data instances-10% data			
		KDDCup 99		NSL-KDD	
		Train	Test	Train	Test
Normal	Normal connection records	97,278	60,593	67,343	9,710
DoS	Attacker aims at making network resources down	391,458	229,853	45,927	7,458
Probe	Obtaining detailed statistics of system and network configuration details	4,107	4,166	11,656	2,422
R2L	Illegal access from remote computer	1,126	16,189	995	2,887
U2R	Obtaining the root or super-user access on a particular computer	52	228	52	67
Total		494,021	311,029	125,973	22,544

Recently, a comprehensive literature survey on machine learning based ID with KDD Cup99 dataset was conducted [17] [18].

2.5 Previous studies

Yin et al. [19] presented a modeling of a deep learning-based intrusion detection system using RNN. The model was evaluated on the NSL-KDD dataset, for which the highest accuracy is obtained with 80 hidden nodes and 0.1 learning rate. The accuracy reached 80% for five-class classifications and 81.2% for the case of binary classification. However, it is worth mentioning that the presented model requires large training time, and the results of R2L and U2R classes show lower detection.

Lin et al. [20] proposed an intrusion detection system to classify network attacks using convergent neural networks (CNNs) based on 5-LeNet. The model was trained and tested on the dataset and KDD9, in which the obtained results achieved 96% detection accuracy rate. Nevertheless, it is worth mentioning that not all features in the data set were taken in the proposed model, in which just 32 features out of 41 were considered.

Vigneswaran et al. [21] used deep neural networks (DNNs) to predict attacks on the network intrusion detection system (N-IDS), for which the KDD Cup99 dataset was used. The results showed that the DNN architecture with 3 hidden layers has superior performance over all other classical algorithms and learning algorithms, in which the detection accuracy, recall and f1-score were 0.92, 0.91 and 0.95, respectively.

Vigneswaran et al. [1] used a deep neural network DNN ID3 development for detecting and classifying cyber-attacks using the KDD Cup99 dataset. This work proposes a DNN

architecture consisting of an input layer, hidden layers, and an output layer. The training accuracy ranges from 95% to 99% for most of the DNN typologies. However, the proposed model requires a lot of training time to obtain an optimal network topology in its structure.

Alsughayyir et al. [22] used deep learning technique to develop a network attack detection system. The auto-encoder technology was used to classify normal behavior from network anomalies based on the NSL-KDD dataset. The results showed that the proposal outperforms the classical methods with an accuracy of 99% for training and 91.28% for the testing phase. Nevertheless, more deep learning methods and algorithms are indeed required for network traffic in real time.

Belavagi et al. [23] used Quantal Response Equilibrium-based Game Model and Rule-based Classification to Improved Intrusion Detection System. The results showed that all the attacks are detected with good detection rate and their approach provides optimal usage of IDS.

3. Proposed model

In this paper, a complex deep learning network is designed to obtain a better intrusion detection process than ordinary deep learning networks. In the proposed design, the data entry is considered according to its importance and sequence into deep neural networks, which consist of a set of hidden layers unequally for all data. The data, which is circulated through the network, needs to be pre-processed regardless of its type. Then, the features are extracted and transferred to the classification process to decide whether it is normal or abnormal. In fact, creating a database for a specific manufacturer or company and extracting features from this database cannot be used for comparison purpose in global intrusion detection systems. Therefore, a huge database has been created which simulates a computer network. This network consists of 1000 computers connected via TCP/IP protocol and it contains all kinds of attacks that can threaten the networks in either training or testing set. Based on the created database, the features are then extracted. Each connection record has 41 features, and each feature expresses a different type. The first nine features (from #1 to #9) represent the basic features such as the duration of the connection, type of protocol, number of transferred bytes, and a flag indicating the state of the connection (either normal or error). These attributes provide information for the purposes of protocol analysis. The next 13 features (from #10 to #22) represent the content features which reflect the snooping behavior such as the number of login failures from the data content. The last 19 types (from #23 to #41) are the time-based traffic features. These attributes reflect the connections between the current and the previous records. This information is crucial for data transmission, for which the feature extraction process plays a major role in the classification/detection phase with the utilization of a deep learning model as shown in Figure 3. Based on that, same standard data set, that is KDD CUP9, is selected for the evaluation of the classification/detection models. In view of the detection stage using deep learning, a new intrusion detection system was designed in this research using deep neural networks based on the KDD CUP99 database. Rather than introducing all the 41 features of the database into a deep learning system at once, the features are divided into basic features (from #1 to #9) and content features (from #10 to #22) and traffic features (from #23 to #41). This results in a new method of design, in which the input process is phased. Since the features are divided into three sections, it is suggested to introduce them in three stages depending on the sequence and importance of these features. As the basic features are crucial, they are entered first into the system to be partially processed in the first data processing stage.

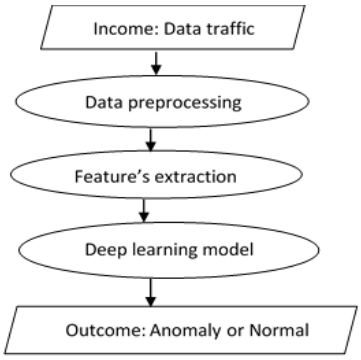

Figure 3. Intrusion detection system block

This stage is represented by the first and second hidden layers which consist of 16 and 64 neurons, respectively. In the second partial data processing stage that is represented by the third hidden layer, the content and basic features are inserted, in which the hidden layer becomes an input layer with 86 neurons. After that, the output of the second partial processing stage is then entered into the third partial processing stage, which is represented by the fourth and fifth hidden layers with 86 and 128 neurons respectively.

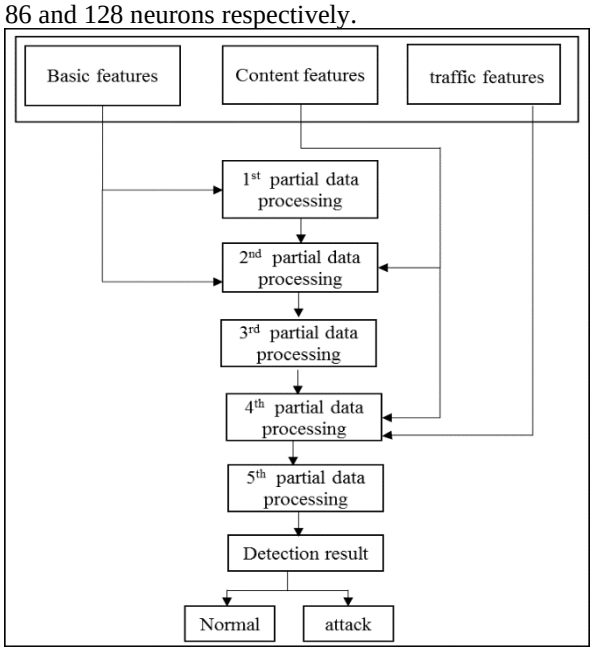

Figure 4. Deep learning-based intrusion detection system

Then, the features coming out from the third partial processing stage along with the traffic and content features are entered into the fourth partial processing stage represented by the sixth hidden layer with 160 neurons. The output of the fourth partial processing stage is entered directly into the fifth partial processing stage that has the seventh and eighth hidden layers with 128 and 256 neurons, respectively. This is the last processing process. With the completion of the data processing process, the classification stage is now introduced which consists of a one neuron. According to the result of this neuron, the decision is made as either an intrusion or an attack as shown in Figure 4. The output of each neuron at every processing stage is represented by the input values of that neuron multiplied by its weights. If the weighted sum of the

input values is greater than a certain value called the threshold, the neuron activates and sends a signal based on the activation function of the neuron. The pycharm program was used to handle the deep learn design in the proposed architecture. Figure 6 illustrates the proposed complex deep neuron structure, the number of neurons in each layer, and the input and output of each layer. The ***relu*** function has been used in the hidden layers as it is more efficient and has the ability to speed up the entire training process. Moreover, the ***sigmoid*** activation is utilized in the output layer due to its nature in the binary separation, in which two values (0 or 1) are returned in the output. Since there are nine basic features, the number of neurons in the hidden layer is doubled by 16, which is the nearest number to 2^n. Further, to reduce the network complexity and training time, one hidden layer was shortened. Consequently, the number of neurons in the next hidden layer was directly doubled to 64. It is worth mentioning that the 32-neuron layer was finally decided to be shortened, as the results were not satisfactory when we have tried to shorten the 64-neuron layer and rest of hidden layers. On the other hand, the learning is kept constant at 0.01, while other parameters are optimized. The proposed scheme was trained 10 times and the training process took approximately one hour on the CORE i3 processor. In the proposed structure, the input, dense and output represent the input layer, hidden layer, and output layer, respectively. Moreover, the question mark (?) denotes the model before the data was entered into it.

The block diagram of the proposed intrusion detection system is illustrated in Figure 5.

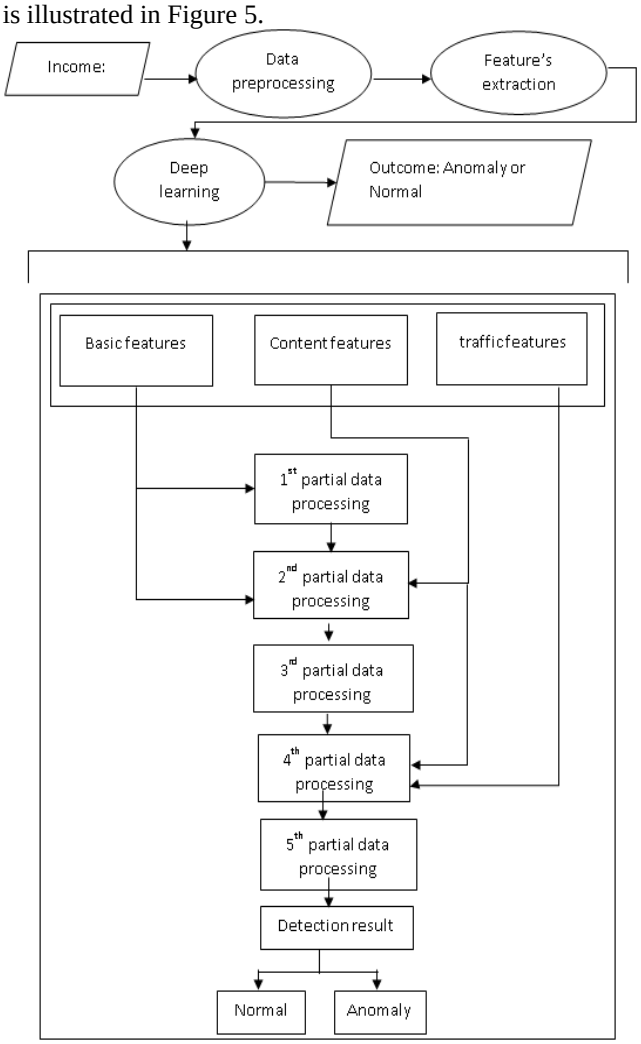

Figure 5. Proposed Model Intrusion Detection System

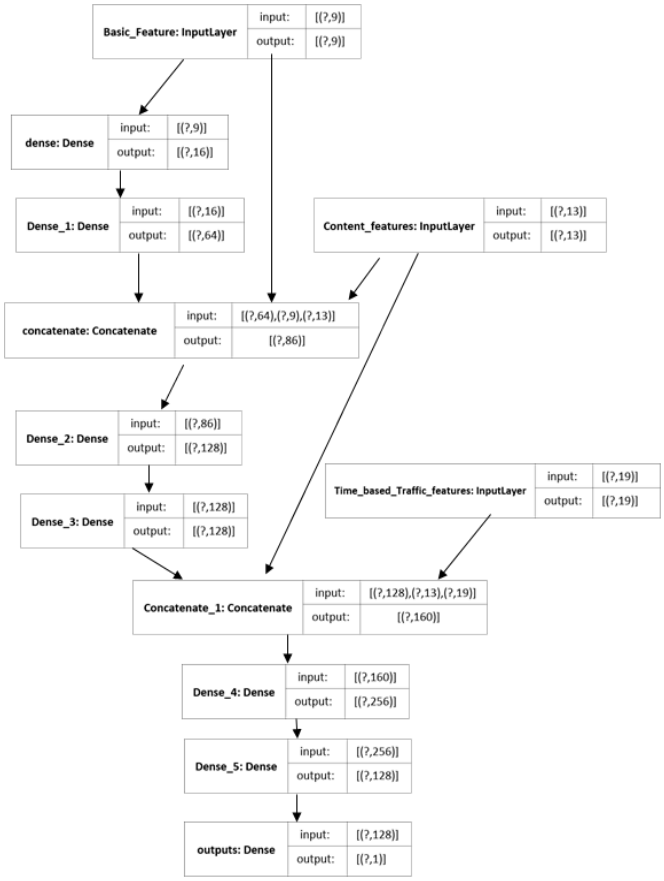

Figure 6. Neural Network deep Architecture.

4. Results and comparison

For evaluation and comparison purposes, the classical algorithms are re-applied. Besides, the normal DNN and the complex DNN modeled on the 99-KDDCup dataset are trained. After that, all models are re-compared.

The comparison results of the proposed new model with other algorithms are given in Table 4.

Table 4. The results of new model and comparing it with other algorithms.

Algorithm	Accuracy	Precision	Recall	f1-score	FPR
DNN-New	0.949	0.999	0.915	0.955	0.001
DNN-3	0.928	0.999	0.915	0.956	0.001
Ada Boost	0.925	0.995	0.911	0.951	0.005
Decision Tree	0.931	0.999	0.915	0.955	0.001
K-Nearest Neighboor	0.929	0.998	0.913	0.954	0.002
Linear Regression	0.846	0.988	0.819	0.896	0.012
Navie Bayes	0.929	0.988	0.923	0.955	0.012
Random Forest	0.9.27	0.999	0.910	0.953	0.001
SVM-Linear	0.811	0.994	0.770	0.868	0.006
SVM-rbt	0.811	0.992	0.772	0.868	0.008

It can be observed from Table 4 that the decision tree algorithm performs better in terms of accuracy compared to the normal deep neural network, despite the same number of training times used in the previous study [21]. However, it is also observed that the new deep network is superior to the normal deep network and all other classic machine learning algorithms. This is due to the DNNs ability to extract data and features with high abstraction, as well as the income distribution that ease the burden on the network. The advantage of network non-linearity is also considered compared to other algorithms. It gave the best accuracy despite the few training (10 times) which requires less training time. On the other hand, the normal deep neural network was trained 1000 times to obtain an accuracy of 0.93 with a large training time. Due to the distribution of income, the designed network has a small probability of its collapse no matter how many times the training is increased. In the following, a comparison for each of the variables is illustrated.

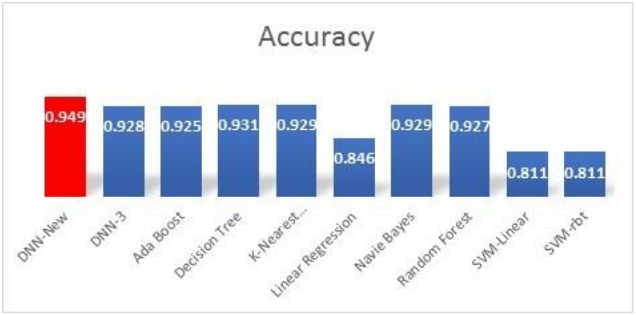

Figure 7. Accuracy comparison

It is evident from Figure 7, that the proposed model obtained the best accuracy of 0.94 compared to other algorithms.

It can be observed from Figure 8, that the proposed model obtained a very low false- positive rate which is 0.001. Similar value is achieved by random forest and decision tree.

Although the Naive Bayes algorithm outperformed in obtaining the highest true positive rate as shown in Figure 9, the proposed model achieved a value of 0.915 which is very close to the one obtained by Naive Bayes algorithm.

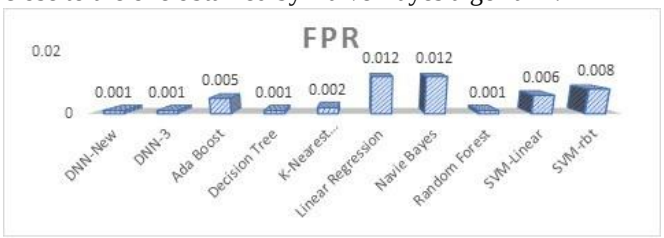

Figure 8. False positive rate (FPR) comparison

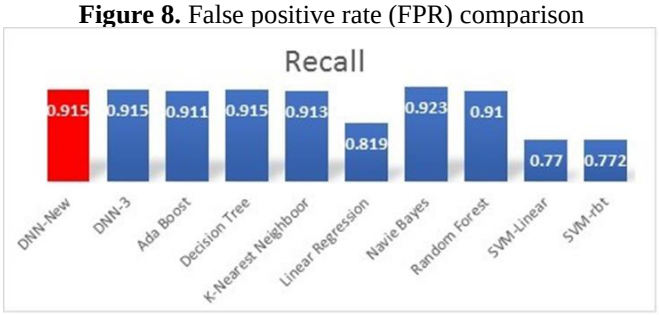

Figure 9. Recall comparison

5. Conclusion

Deep learning networks have proven to be effective in intrusion detection systems for detecting the network attacks. It achieved high detection accuracy compared to machine learning methods. The proposed complex deep web-based detection system has a significant ability to distinguish normal traffic from anomalous traffic. However, the proposed model requires more training in order to increase the accuracy of the system.

References

[1] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. "Venkatraman, Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[2] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," Knowledge-Based Syst., vol. 189, p. 105124, 2020.

[3] S. Dua and X. Du, "Data mining and machine learning in cybersecurity," CRC press, 2016.

[4] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," Appl. Sci., vol. 9, no. 20, p. 4396, 2019.

[5] G. Karatas, O. Demir, and O. K. Sahingoz, "Deep learning in intrusion detection systems," in International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), pp. 113–116, 2018.

[6] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," IEEE Commun. Surv. Tutorials, vol. 21, no. 1, pp. 686–728, 2018.

[7] A. Azab, M. Alazab, and M. Aiash, "Machine learning based botnet identification traffic," in IEEE Trustcom/BigDataSE/ISPA, pp. 1788–1794, 2016.

[8] G. E. Hinton, "A practical guide to training restricted Boltzmann machines, in Neural networks: Tricks of the trade," Springer, pp. 599–619, 2012.

[9] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in IEEE international conference on acoustics, speech and signal processing, pp. 6645–6649, 2013.

[10] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in international conference on machine learning, pp. 1764–1772, 2014.

[11] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to sequence learning with neural networks," in Advances in neural information processing systems, pp. 3104–3112, 2014.

[12] R. I. Dr. Hassan Alahmad, "Using Neural Networks to Build an Intrusion Detection System based on Standard Dataset (KDD99)," Tishreen Univ. J. Res. Sci. Stud. - Eng. Sci. Ser., vol. 39, no. 5, pp. 287–310, 2017.

[13] V. N. Tiwari, S. Rathore, and K. Patidar, "Enhanced Method for Intrusion Detection over KDD Cup 99 Dataset," in International Journal of Current Trends in Engineering & Technology, vol.02, no.02 ,2016.

[14] W. Lee, S. J. Stolfo, and K. W. Mok, "Adaptive intrusion detection: A data mining approach," Artif. Intell. Rev., vol. 14, no. 6, pp. 533–567, 2000.

[15] Irvine, "KDD Cup 1999 Data.," The UCI KDD Archive Information and Computer Science, University of California, Irvine, [Online]. Available: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Accessed: 28-Oct-1999].

[16] Eastmount, "Network Security Self-study," [Online]. Available: https://github.com/eastmountyxz/NetworkSecuritySelf-study. [Accessed: 16-Oct-2019].

[17] A. Özgür and H. Erdem, A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015, PeerJ Prepr., vol. 4, p. e1954v1, 2016.

[18] M. A. Al-Shabi, "Credit card fraud detection using autoencoder model in unbalanced datasets," J. Adv. Math. Comput. Sci., pp. 1–16, 2019.

[19] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017.

[20] W.-H. Lin, H.-C. Lin, P. Wang, B.-H. Wu, and J.-Y. Tsai, “Using convolutional neural networks to network intrusion detection for cyber threats,” in IEEE International Conference on Applied System Invention (ICASI), pp. 1107–1110, 2018.

[21] R. K. Vigneswaran, R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security,” in 9th International conference on computing, communication and networking technologies (ICCCNT, pp. 1–6), 2018.

[22] B. Alsughayyir, A. M. Qamar, and R. Khan, “Developing a Network Attack Detection System Using Deep Learning,” in International Conference on Computer and Information Sciences (ICCIS, pp. 1–5), 2019.

[23] M.C. Belavagi, and B. Muniyal. "Improved Intrusion Detection System using Quantal Response Equilibrium-based Game Model and Rule-based Classification," International Journal of Communication Networks and Information Security(IJCNIS), vol.13, no.1, pp.1-8,2021.