International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 2, ISSN No. 2455-2143, Pages 312-318
Published Online June 2022 in IJEAST (http://www.ijeast.com)

NETWORK INTRUSION DETECTION SYSTEM USING ML AND NEURAL NETWORKS

Nithin, Nitin R Naik, Sameer Ahmed S, Sumeet Tompe, Dr. D B Srinivas
Department of ISE
Nitte Meenakshi Institute of Technology, Bengaluru,
Karnataka, India

***Abstract*- The largest threat to networking systems is the constantly changing nature of Internet attacks by unauthorised users. Intrusion Detection System (IDS) monitors networking systems for unauthorized activity, mostly from the Internet but also increasingly from hosts and Wi-Fi networks. These activities are typically the result of an attack being carried out by a hacker. IDS can spot harmful activity that firewalls typically do not, such as Trojan horses, worms, viruses, attacks on vulnerable services, unauthorized logins, privilege escalation, and application attacks. The unauthorized actions by unauthorized users are changing continuously over a period of time. The unauthorized activities that were carried out today are wholly different from those that were committed a year ago. The models developed using Defence Advanced Research Projects Agency (DARPA) IDS are not well suited to identifying modern attacks. Furthermore, the DARPA IDS attacks and their underlying problems are well known. NSL-KDD data sets are still in use but not suitable today's era of attacks. Recently, UNSW-NB15 was released, which might address the KDD Cup 99 and NSL-KDD issues. In our proposed work, we implement the IDS using the UNSWNB-15.**

I. INTRODUCTION

ICT systems have steadily become more and more crucial to businesses and people's daily lives over the years. The attacks on the ICT system are also highly diverse and are progressively becoming more severe. As a result, the ICT system sees a clear need for an upgraded, integrated network security solution.

A system called an intrusion detection system (IDS) watches network traffic for unfamiliar activity and sends out signals when it is detected. IDS is a portion of software that checks an entire system or network for malevolent policy violations. Any illegal or malicious activity is often stated to the administration or centralized information is gathered using security info and an event management system. A SIEM associations output from several sources & use various alarm clarifying mechanisms to differentiate between legitimate and erroneous alarms. While monitoring networks for potentially harmful activity or infractions, intrusion detection systems are also prone to raising false alarms. In light of this, businesses and consumers should adjust their IDS products after initial installation. In order to avoid triggering false alarms, it requires suggests that organizational intrusion detection systems differentiate between safe network traffic and malicious activity.

The terms intrusion detection and prevention system and intrusion prevention system are both acceptable terms. It is a network safety program that checks all system and network actions for malicious behaviour. The main responsibilities of intrusion prevention systems are to spot suspicious activity, gather data on it, report it, and make an effort to stop or block the problem. It is envisaged that intrusion prevention systems will work in addition to intrusion detection systems because both systems keep an eye on system operations and network traffic for various hostile acts. The information connected to system events is naturally recorded by an intrusion prevention system, which also alerts security administrators to significant observed occurrences and generates reports that are useful for future work. A lot of IPS can also react to a threat by making an effort to stop it from being successful in its objectives. They employ a variety of reaction policies, including the IPS preventing the occurrence of attack directly, altering the safety context, or altering the attacked content.

An intrusion detection system's objective is to identify unwanted access to a computer network by scanning network traffic for indications of malicious activity. Building a prediction model with the ability to differentiate between intrusions or assaults and regular network connections is the problem of intrusion detection. An effective intrusion detection system (IDS) monitors network traffic both inbound and outbound, continuously analyze patterns of activity, and immediately notifies the administrator of any unexpected network behaviour. Integral intrusion signatures are used by an intrusion detection system to spot any malicious activity that could hurt the network

The rest of the paper is organized as follows: in section 2 related work is explained.. Section 3 describes KDD CUP99 dataset and the steps taken to create the UNSW-NB15 dataset. Section 4 consists of methodology and

implementation. Section 5 explains results and finally in section 6 we conclude our work.

II. RELATED WORK

Since the invention of computer architectures, researchers have been studying ID in relation to network security. In recent years, using ML approaches and solutions to comprehensive IDS has gained popularity, although the amount of training data available is little and is frequently used mainly for benchmarking.

In order to create semi supervised data for feature selection, [1] exploited the homogeneousness of the data cluster and label. The effectiveness of the feature selection process is improved by this technique. To enhance the IDS, [2] suggested an inherent (genetic) algorithm and Super Vector Machine with a novel feature selection method and the time it takes to execute the new feature selection approach, which is based on a genetic algorithm with an advanced capability function to enhance the TP rate and decrease the FP rate instantaneously, is decreased. A feature selection method for detection was put out by [3]. Prior to categorization, feature selection is a critical step.

To improve the algorithm's accuracy, [4] suggested a successful "wrapper-based feature selection method". According to [5] anomaly-based detection is more effective than signature-based NID. The detection of anomalies does not follow patterns like signature-based detection. Deep belief networks (DBN) were employed as classifiers [6]in together with restricted Boltzmann machines for training and back propagation for KDDCup 99 fine tuning. In [7], authors have proposed anomaly based IDS using an Artificial Neural Network (ANN) model.

Support vector machines (SVM), a traditional machine learning classifier, and neural networks were combined in[8] to classify the connection records of the KDDCup 99. A number of studies, [9] have shown how changed Jordon design might convey rules for attack pattern patterns and increase detection rates. Deep learning methods for traffic recognition were proposed in[10]. The KDDCup'99 data sets characteristics and its classification methods were thoroughly researched utilizing the LSTM methodology in [11], [12] , and [13]. The main goal of this study is to make use of the prospect that an incoming cyber-attack might be random, undetectable to the human eye but filterable by adding an artificial intelligence layer to the network. Therefore, by using historical data on cyberattacks to train the neural network, it may quickly learn to predict incoming attacks and can either alert the system or launch a pre-planned response that may stop the attack in its tracks. As a result, by simply adding an extra layer to the security system, millions in aftershock collateral damage and costly data exposures can be avoided. The benchmarking dataset used to train the networks is outdated, and more recent data must be used for retraining before deploying in the field to improve the algorithm's real-time robustness. This paper's mandate is to teach the fundamentals of artificial neural networks to the extremely quickly developing topic of cybersecurity with machine learning. Section 5 concludes the work and future intentions.

III. DATA SET

Anomaly Detection Systems have gained further importance in identifying new assaults over the last few years than Signature-based Detection systems. Due to three key problems, using the KDD99 and NSLKDD benchmark data sets to evaluate NIDSs doesn't produce reasonable results: (i) due to their lack of recent low footmark attack styles, (ii) due to their non-existence of recent normal traffic situations, and (iii) dissimilar spreading of training and testing sets. To overcome these matters, UNSW-NB15 has been newly produced.

a. KDD CUP99

The Lincoln laboratories at MIT University created the DARPA98 set and carried out the replication using regular and irregular traffic in an armed network environment. The training data consisted of flattened binary tcpdump files from 7 weeks of network activity, which had a size of approximately 4 Giga Bytes. The simulation finished with 9 weeks of raw tcpdump files. This was then converted into a little over 5 million connection records. The simulation produced test data for two weeks, consisting of 2 million connection records. Using the Bro-IDS tool, the simulation produced 41 features for each link along with the class label, after upgrading the comprehensiveness of the DARAP98 network data features and using the same environment. The KDDCUP99 is the updated version of the DARAP98. The entire collection of extracted features from the KDDCUP99 data set has been separated into three categories: intrinsic features, content features, and traffic features. Additionally, the assault records in this data collection are divided into four categories. The test data of KDDCUP'99 comprises 15 attack types, whereas the training set has 22 attack types. These datasets have been used by many IDS researchers since they are accessible to the general public. However, numerous studies have noted that the transparency of the IDS estimation might be impacted by three significant drawbacks of these datasets. Initial time-to-live values (TTL) for each attack data packet are 126 or 253, whereas TTLs for traffic packets are typically maybe 127 and 254. TTL numbers 126 and 253 are non-existent in the attack's training records in the dataset, though. As a result of including additional attack records in testing dataset, the testing set's probability distribution is different from the training set's probability distribution. As a result, some records are favoured by skewed or biased classification algorithms rather than a balance between different attack types and standard annotations. Last but not least, the data set does not provide a complete picture of the forecasts for newly reported low footprint attacks.

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 2, ISSN No. 2455-2143, Pages 312-318
Published Online June 2022 in IJEAST (http://www.ijeast.com)

b. UNSW-NB15

The specifics of UNSW-NB15 dataset are presented in this segment. The segment contains information on the steps taken to create the UNSW-NB15 dataset, as well as information on its features and records distribution.

i. Usage of IXIA tool Testbed Configuration

The dataset was created using the IXIA traffic generator. According to Fig. 1, it is set up with 3 virtual servers. Here, servers 1 and 3 are set up for the typical distribution of traffic, whereas server 2 is for any unusual or malicious network traffic activity. There are 2 virtual interfaces that are used to connect the servers together and collect traffic from the public and private networks. Afterward, two routers are used to connect the servers to the hosts. These routers are also connected to a firewall device, which is configured to transmit all traffic, abnormal or legitimate, via it. Pcap or raw data of the simulation uptime should be recorded, the tcpdump utility is installed on router 1.

The primary goal of the entire testbed was to record any usual or anomalous traffic that remained distributed around network nodes and came from the IXIA tool. Prominently, the IXIA tool was used to generate attack traffic in addition to regular traffic, and attack actions were sourced as of the CVE site with the goal of accurately simulating a contemporary risk environment.

In order to gather the first 50 GigaBytes of data during the initial simulation, one assault is generated by the IXIA tool per second. Additionally, another simulation was set up to launch 10 attacks every sec to abstract an additional 50 GigaBytes on the other side. This is due to how quickly network traffic moves and how modern exploits exploit it.

Fig. 1: Testbed Visualization [5]

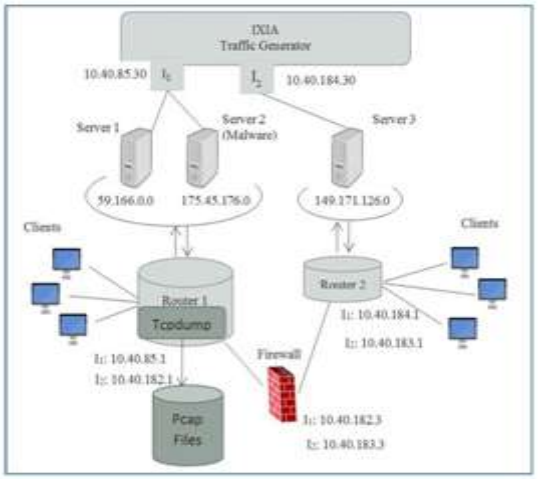

Table 1: Dataset Statistics [5]

Statistical features		16 hours	15 hours
No. of flows		987,627	976,882
Src bytes		4,860,168,866	5,940,523,728
Des bytes		44,743,560,943	44,303,195,509
Src Pkts		41,168,425	41,129,810
Dst pkts		53,402,915	52,585,462
Protocol types	TCP	771,488	720,665
	UDP	301,528	688,616
	ICMP	150	374
	Others	150	374
Label	Normal	1,064,987	1,153,774
	Attack	22,215	299,068
Unique	Src ip	40	41
	Dst ip	44	45

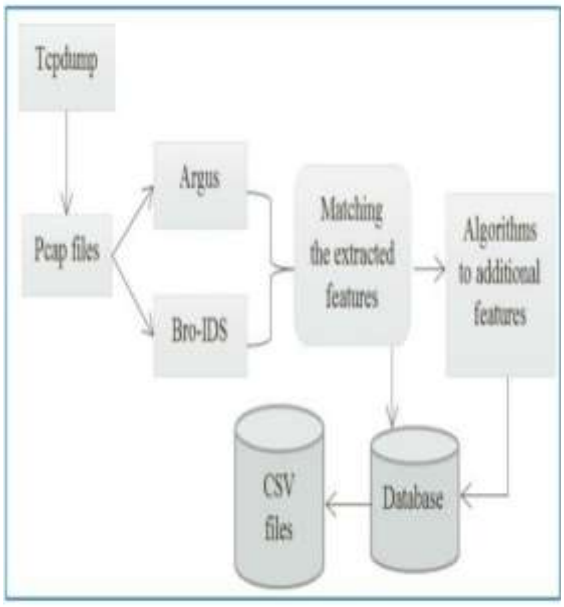

Fig. 2: Architecture for generating dataset [5]

ii. Feature Selection:

Feature selection was done using the Argus and Bro-IDS tools. The output files of these two different tools are then stored in the SQL server database to match the generated features by using the flow features. After the feature selection, the selected 49 features are shown in the figures. The features from 1-35 represent the combined gathered information from data packets. And from 1-5 consists of FLOW FEATURES, 6-18 consists of BASIC FEATURES, 19-26 consists of CONTENT FEATURES, 27-35 of TIME

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 2, ISSN No. 2455-2143, Pages 312-318
Published Online June 2022 in IJEAST (http://www.ijeast.com)

FEATURES, 36-47 consists of GENERAL FEATURES. The extra features of the UNSW-NB15 dataset are features 36 through 47. There are two sections to these twelve characteristics. Features 36 through 40 are connection features, while features 41 through 47 are general-purpose features. 48 and 49 are designated as features. Attack categories (i.e., attack cat) and labels of 0 or 1 for each record indicate whether the record is normal or attacked in the data set.

iii. Dataset Records Distribution

Table2 displays the distribution of each account in the UNSW-NB15 dataset.

Table 2: Dataset Record distribution [5]

Type	No. Records	Description
Normal	2,218,761	Natural transaction data.
Fuzzers	24,246	Attempting to cause a program or n-etwork suspended by feeding it the randomly generated data.
Analysis	2,677	It contains different attacks of port scan, spam and html files penetrations.
Backdoors	2,329	A technique in which a system security mechanism is bypassed stealthily to access a computer or its data.
DoS	16,353	A malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the Internet.
Exploits	44,525	The attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability.
Generic	215,481	A technique works against all block-ciphers (with a given block and key size), without consideration about the structure of the block-cipher.
Reconnaissa-nce	13,987	Contains all Strikes that can simulate attacks that gather information.
Shellcode	1,511	A small piece of code used as the payload in the exploitation of software vulnerability.
Worms	174	Attacker replicates itself in order to spread to other computers. Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it.

IV. METHODOLOGY

The Fig. 3 shows the complete details of the methodology used for our project "Predicting the Cyber Attacks/crime". The dataset here we used is UNSW-NB15, and we have discussed this in the previous section. There are many reasons to choose UNSW-NB15 over the other datasets like KDDCUP99, NSLKDD, and other datasets which are used for NIDS.

a. Importing Libraries:

In order to perform data pre-processing, selecting classifiers, and important tasks, Some preconfigured libraries need to be imported. Specific tasks are carried out using these libraries. The following specific libraries will be used by our model to carry out tasks and work:

Any type of mathematical operation can be included in the programming using Numpy. Python's 2D charting library Matplotlib requires the import of a sub-library called pyplot. Any type of chart can be plotted in Python using this package. One of the most well-known Python libraries, Pandas is an open-source data analysis and manipulation framework used for importing and maintaining datasets.

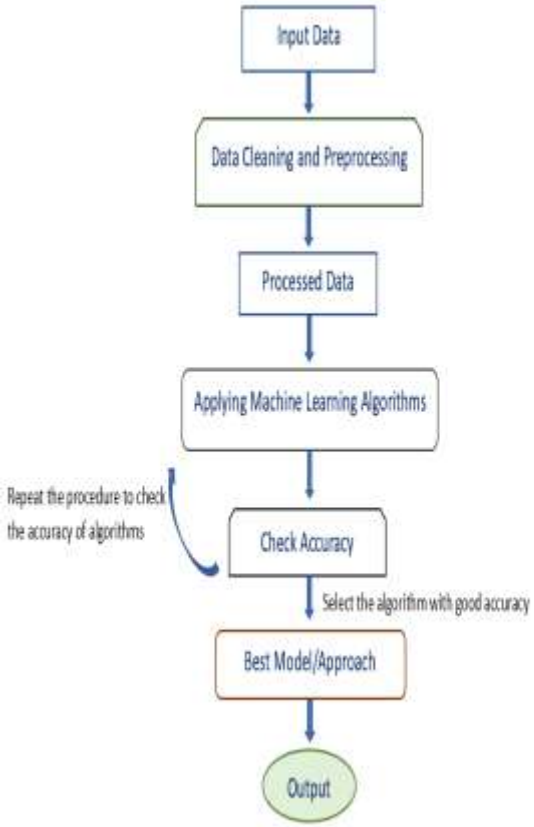

Fig. 3: Proposed Architecture

Sklearn is an open-source toolkit that implements a range of ml, dl, pre-processing, cross validation, and visualization techniques using a standardized user interface. Additionally, it is a quick and effective tool for data mining analysis. It comprises a variety of clustering, regression, and classification algorithms, such as random forests and support vector machines. The process of getting raw data ready for machine learning models is known as data pre-processing. The first and most crucial stage in developing a machine learning model is this one. It is not always the case that we find clean, well-formatted data while building a

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 2, ISSN No. 2455-2143, Pages 312-318
Published Online June 2022 in IJEAST (http://www.ijeast.com)

model. And in working with data, it is necessary to clean it up and organize it so that it can be used easily and produce accurate results. Our machine learning models cannot be directly applied to real-world data since it commonly contains noise, and missing numbers, and may be in an unfavorable format. The pre-processing procedure includes acquiring the dataset, importing libraries, importing datasets, identifying missing data, encoding categorical data, splitting the dataset into a training and test set, and feature scaling.

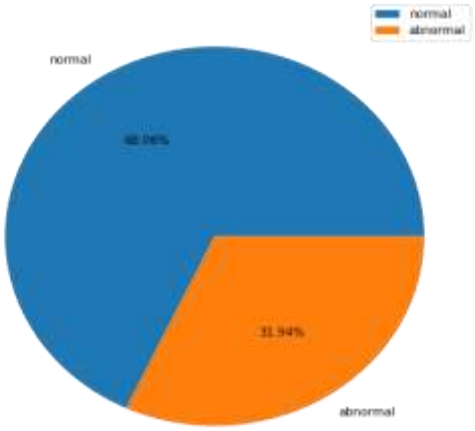

Fig. 4: Distribution of normal and abnormal data

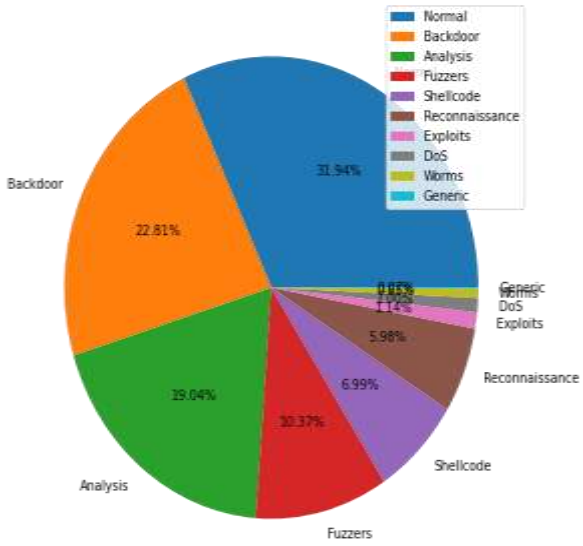

Fig. 5: Distribution of Multi-class labels

b. Implementation:

The UNSWNB-15 dataset was used. There are around 45 columns in this dataset. There are about 175K entries in the dataset. There are about 68% normal data points and 32% abnormal data points in the dataset. There are about 9 types of attack categories in the data set. First, we are going to apply normalization to our data set so that every value in the dataset is converted between 0 and 1. Normalization mainly helps in model processing and makes error detection easy. In our data set x=0 implies normal and x=1 implies abnormal. There are about 45 features in our dataset, if we use all these features in our dataset then definitely our accuracy will drop down to great extent, also we have used a correlation matrix on these features and found out which has less co-relation and used those features for our model. We have taken around 14 features after implementing a co-relation matrix out of 45 features to further train our model. All the data after the correlation matrix are stored in a bin.csv file. 20% will be test data and 80% will be train data. We have applied the below algorithms for our project. After performing the pre-processing techniques on the data set, we will be using different algorithms for NIDS to build the best model with high accuracy and good efficiency. Later we compare each with other models to select the best from them. The selected feature are " 'rate', 'sttl', 'dload', 'swin', 'dwin', 'dmean', 'ct_state_ttl', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm', 'proto_tcp', 'state_CON', 'state_FIN', 'state_INT' ".

We have performed the operation on classifiers/algorithms like "Logistic Regression, Random Forest, KNN, Decision tree, and also neural network algorithms like Multilayer perceptron, ANN with a different number of hidden layers". For neural network algorithms, we must train the models using various hidden layer counts. We have performed models with a different number of epochs like 2,10 and 100. As the number of epochs increases, we observed that there is a rise in the accuracy of the model with the different number of hidden layers.

V. RESULT

The result of the models we trained has given some good values. Additionally, we worked on the ANN model, which outperformed these machine learning methods in terms of accuracy. In comparison, the ANN model has the highest accuracy. We must train the models using various hidden layer counts. We have performed the models with a different number of epochs like 2,10 and 100. We found that the accuracy of the model with the variable number of hidden layers increases as the number of epochs increases. We got an accuracy of 93.6 at the beginning of the epoch of models with different hidden layers. Additionally, we have increased the number of hidden layers, which has led to a minor improvement in the model's accuracy. And we have strained this by increasing the number of epochs to train the model. In the beginning, we got an accuracy of 93% and after performing the multiple operations it increased to 94.5%. By this, we can conclude that the neural network model gave the best accuracy compared to others. And further, we can achieve more accuracy by performing the model with different algorithms in neural network/deep learning like CNN, RNN, etc.

International Journal of Engineering Applied Sciences and Technology, 2022
Vol. 7, Issue 2, ISSN No. 2455-2143, Pages 312-318
Published Online June 2022 in IJEAST (http://www.ijeast.com)

Table 3: Model Performance Results

Classification Models						
Performance parameters	Logistic Regression	Random Forest	XG Boost	Light GBM	KNN	MLP
Accuracy in (%)	80	92.36	90.40	90.80	87.36	93.6
Recall in (%)	75	92	91	91	89	96
F1-Score in (%)	76	91.3	90	91	88	94
Precision in (%)	83	91	90	90	89	95
ROC in (%)	75	95	90	90	90	96

VI. CONCLUSION AND FUTURE WORK

Due to the similarities between the UNSW-NB15 data set and contemporary attacks on regular network traffic, the data set is regarded as complex. This suggests that this data set can be used to assess both established and novel NIDS algorithms in a trustworthy manner. This research concludes that the neural network model gave the best accuracy compared to others. And further, we can achieve more accuracy by performing the model with different algorithms in neural network/deep learning like CNN, RNN, etc.

Future work will involve using real-time network traffic data to apply the debated deep learning models. After the dataset has been generated and turned into connection records, additional features from "different logs, firewalls, alarms of each system, Syslog servers, routers, and switches" may be added to increase the dataset's depth. We consider the generation of actual data as future work and test this by applying the previously stated learning algorithms to it.

With the use of various algorithms and approaches, the results of our machine learning and deep learning techniques (algorithms) can be utilized to create models that are very accurate and efficient. Several articles have used the UNSW-NB15 dataset to construct an effective IDS forecasting model that will be used in future work. Current machine learning-based IDS will always rely on historical data and might not be successful against freshly developed attacks. The deep learning models can be employed for unexpected patterns and are dynamic. It aids in improving the model's efficiency and accuracy.

VII. REFERENCES

[1]. Coelho, Frederico; Braga, Antonio Padua; Verleysen, Michel. (2012). Cluster homogeneity as a semi-supervised principle for ´feature selection using mutual information. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning 2012 (pp. 507-512).

[2]. H, Gharaee ; H, Hosseinvand. (2016). "A new feature selection IDS based on genetic algorithm and SVM," 2016 8th International Symposium on Telecommunications (IST), 2016, pp. 139-144, doi: 10.1109/ISTEL.2016.7881798.

[3]. A, Gül ; E, Adalı.(2017). "A feature selection algorithm for IDS," 2017 International Conference on Computer Science and Engineering (UBMK), 2017, pp. 816-820, doi: 10.1109/UBMK.2017.8093538.

[4]. F, Zhang ; D, Wang.(2013). "An Effective Feature Selection Approach for Network Intrusion Detection," 2013 IEEE Eighth International Conference on Networking, Architecture and Storage, 2013, pp. 307-311, doi: 10.1109/NAS.2013.49.

[5]. N, Moustafa; J, Slay.(2015). "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1-6, doi: 10.1109/MilCIS.2015.7348942.

[6]. N, Gao; L, Gao; Q, Gao ; H, Wang.(2014). "An Intrusion Detection Model Based on Deep Belief Networks," 2014 Second International Conference on Advanced Cloud and Big Data, 2014, pp. 247-252, doi: 10.1109/CBD.2014.41.

[7]. B, Subba; S, Biswas ; S, Karmakar.(2016). "A Neural Network based system for Intrusion Detection and attack classification," 2016 Twenty Second National Conference on Communication (NCC), 2016, pp. 1-6, doi: 10.1109/NCC.2016.7561088.

[8]. Mukkamala, Srinivas; Sung, Andrew H ; Abraham, Ajith.(2005). Intrusion detection using an ensemble of intelligent paradigms,Journal of Network and Computer Applications,Volume 28, Issue 2,2005,Pages 167-182,ISSN 1084-8045.

[9]. J.-S, Xue; J.-Z, Sun ; X, Zhang.(2004). "Recurrent network in networkintrusion detection system," in Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, vol. 5. IEEE, 2004, pp. 2676–2679.

[10]. Wang,Zhanyi.(2015). "The applications of deep learning on traffic identification," BlackHat USA, 2015.

[11]. Staudemeyer, R C ; Omlin, C W.(2014). "Extracting salient features for network intrusion detection using

machine learning methods," South African computer journal, vol. 52, no. 1, pp. 82–96.

[12]. Staudemeyer, R C ; Omlin, C W.(2013). Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data. In: Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, pp. 218–224.

[13]. Staudemeyer, R C.(2015). "Applying long short-term memory recurrent neural networks to intrusion detection," South African Computer Journal, vol. 56, no. 1, pp. 136–154.