Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 57–66

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Detection of Network Attacks using Machine Learning and Deep Learning Models

Dhanya K. A.[a], Sulakshan Vajipayajula[b], Kartik Srinivasan[c], Anjali Tibrewal[c], T. Senthil Kumar[d], T. Gireesh Kumar[d]

[a] *TIFAC-CORE in Cyber Security, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.*
[b] *STSM Architect, IBM Security, IBM.*
[c] *Sr. Architect, IBM Security. IBM India Pvt Ltd*
[d] *Department of Computer Science and Engineering, Amrita School of Computing, Coimbatore, Amrita Vishwa Vidyapeetham, India.*

Abstract

Anomaly-based network intrusion detection systems are highly significant in detecting network attacks. Robust machine learning and deep learning models for identifying network intrusion and attack types are proposed in this paper. Proposed models have experimented with the UNSW-NB15 dataset of 49 features for nine different attack samples. Decision Tree classifier produced the best accuracy of 99.05% compared to ensemble models - Random forest(98.96%), Adaboost(97.87%), and XGBoost(98.08%). K-Nearest Neighbour classifier trained for various values of K and best performance obtained for K=7 with the accuracy of 95.58%. A Deep Learning model with two dense layers with ReLU activation and a third dense layer with a Sigmoid activation function is designed for binary classification and produced good accuracy of 98.44% with ADAM optimizer, 80:20 Train-Test Split Ratio. Network attack exploits are detected with an accuracy 95% by XGBoost, Fuzzers attack with 90% accuracy by Random Forest, Generic attacks with 99% accuracy by Random Forest, and Reconnaissance attacks with 79% by Decision Trees. All features are relevant and strong in network attack detection, which eliminates the requirement of feature selection.

Keywords: Network security; Random Forest; Deep Multi-Layer Perceptron; Adaboost; XGBoost

* Corresponding author.Tel.: +91-894-332-0379
E-mail address: t_gireeshkumar@cb.amrita.edu

10.1016/j.procs.2022.12.401

1. Introduction

The rapid increase in network data is due to the Internet of Things(IoT), cloud-based services, and colossal networking devices [31]. Along with network data, attacks are increasing exponentially, becoming a significant threat to network security. Network security can be improved by adding more security devices but cannot ensure complete protection. It needs to address not only present threats but also future threats. The existing Network Intrusion Detection System (NIDS) provides a layered security defense to network at the system, network, application, and transmission levels [12]. Layered security guarantees that further layers will stop an attacker who defeats one layer of defense. Significant challenges of recent NIDS are inadequate accuracy, dynamic behavior of network traffic, low-frequency network attack, adaptability to software-defined networks, the vast volume of stored and transmitted data, and various network access devices.

Most of the existing NIDS are signature-based or anomaly-based detection systems. Signature-based NIDS addresses only the list of known threats, and their indicators of compromise [13]. It has a high processing speed and great accuracy for known attacks. Still, it fails to identify the zero-day attack and unnecessarily raises alerts regardless of the outcome, like Window worm trying to attack the Linux system. It is not practical for internal attacks and depends on the operating system, version, and applications [16]. Anomaly-based NIDS can detect a new suspicious behavior deviating from normalized behavior. Anomaly-based NIDS is good in detecting zero-day attacks. Still, the increased likelihood of false positives results in additional time and resources to investigate all the alerts to potential threats [31].

Machine Learning based NIDS [18] can learn classification models from training data. Training with vast and diverse network data samples makes the model robust to classify attacks into possible categories. Deep learning models also play a vital role in NIDS by learning attack behavior from network features. It also eliminates the requirement for feature correlation, selection, and representation [29]. The deep learning model efficiently learns the hidden network behavior and efficiently identifies the attack with fewer false alarms [32], [30].Unfortunately, attackers are using sophisticated techniques to exploit vulnerabilities of computing resources. On the other hand, the number of compromised computing infrastructure are increasing exponentially. A robust NIDS using machine learning and deep learning techniques with high accuracy and F-Measure is proposed in this paper. The significant contribution of the proposed work is summarised as follows.

1. We evaluated the significance of various classical and ensemble machine learning models in identifying sophisticated network attacks.
2. The lazy learner, K-Nearest Neighbor models are trained for multiple values of K, and results are compared and evaluated its effectiveness in identifying network attacks.
3. Deep multi-layer Perceptron architecture is proposed to improve the classification performance of network intrusion detection systems, and results are compared with machine learning models.

The remaining paper is organized as follows. Section II includes background and literature. The architecture of the attack detection model is proposed in Section III. The results and discussion are included in Section IV. Finally, Section V concludes the research work.

2. Literature Review

2.1. Network attacks

Attacks on computer networks are devastating and can affect the functioning of the entire system by reading, damaging, and stealing the data [11]. Attacks are preceded by pre-intrusion activities like port scanning and IP Spoofing. The primary functions of NIDS are packet sniffing, identifying attack signatures, identifying attacks, and reporting attack details. Attacks are identified by capturing features from source and destination IP addresses, ports, protocol details, header details, etc. Based on the nature of attacks, attacks can be classified as passive and active [12]. The passive attack may be system-based or network-based, where the attacker silently monitors the network and try to learn confidential data. Passive attacks are challenging to monitor. Active attackers break all security measures and get into the networks by exploiting security loopholes, masquerading as a trusted system, or stealing passwords.

2.1.1. Fuzzers Attack

Fuzzers attack inputs a massive amount of random data to the system to make it fail and find bugs [27]. It can identify software and system vulnerabilities and loopholes in networks and operating systems.

2.1.2. Analysis

Penetrates the web application with port scanning, spam emails, and web scripts [22]. Machine learning models can identify port scanning by defeating IP Spoofing, altering port scan frequency, and changing the sequence in which ports are scanned. Spam emails are dangerous as they spread malicious code, run phishing scams and make money. Machine learning models use content-based email filtering, which identifies some keywords that can produce high variance between spam and legitimate emails [6]. Malicious HTML code penetrations have many consequences, like disclosure of cookies, thereby altering the victim's page content.

2.1.3. Backdoor

Backdoor attacks compromise security mechanisms and access computer and their data [22]. This attack targets the privacy and availability of computing resources to users [25].

2.1.4. DoS

DoS attacks make network resources unavailable to the user by suspending service [22]. Verisign reports a massive increase in frequency and complexity of DoS attacks which demand strong NIDs using machine learning and deep learning models.

2.1.5. Exploit

The attacker exploits the vulnerability of software or operating system, takes control of computer resources or network data, and results in system crashes or malfunctions. Zero-day exploits take advantage of software vulnerability about which vendors are unaware.

2.1.6. Generic

Generic attacks work against block ciphers without considering the internal structure of block ciphers [22]. Since the length of the key and blocks are limited, all block ciphers are under the threat of generic attacks. Generic attacks are detected by choosing appropriate external parameters. Different generic attacks on block ciphers are exhaustive key search, dictionary attack, rainbow table attack, etc. [7].

2.1.7. Reconnaissance

Reconnaissance attacks gather all possible information about the target system before launching the actual attack, and it acts as the preparation tool for the actual attack. The three main types of reconnaissance attacks are social, public, and software reconnaissance. During this attack, information is gathered by packet sniffing, port scanning, sweeping the ping, and queries regarding internet information [28]

2.1.8. Shellcode

Shellcode is a small piece of code used as the payload in the exploitation of software vulnerability. It runs a command interpreter that interactively enters commands to be executed on the vulnerable systems and reads back the output [3]. Shellcode attacks can be detected using run-time heuristics representing machine-level operations.

2.1.9. Worm

Worms replicate and spread to other computing resources by exploiting their security failures. Early warning and less reaction time for counteractions are two expected features of the worm detection system. It considers payload content and format, packet headers, network traffic, and monitoring host behavior for worm detection [19]

2.2. Signature based detection

Almutairi et al., proposed a four-component NIDS consisting of an Intrusion Detection System, frequent signature database, updating agent, and complimentary signature database [1]. IDS extracts signature from network packets, compare them with signature databases, and trigger an alert if a match occurs. This four-component system ensures early and accurate detection of attacks with fewer false positives. Attacks with infrequent signatures are also caught with the signatures kept in the complementary database. False alarm minimization is the main issue to be addressed in the signature-based detection and can be solved using signature enhancement, state-full signatures, and vulnerability signatures [14].

2.3. Anomaly Based NIDS

Moustafa et al., performed statistical analysis of the observations and features using the Kolmogorov-Smirnov test, Multivariate skewness, and Multivariate kurtosis. Supervised feature correlation with Gain Ratio and unsupervised correlation with Pearson's correlation coefficient was also performed to measure the relevance between features. Finally, the UNSW-NB15 dataset complexities are evaluated with existing classifiers with metrics accuracy and false alarm rate. The decision tree classifier performed well with an accuracy of 85.56% and 15.78% false alarm rate [23]. Meftah et al., proposed anomaly-based NIDS with machine learning techniques. Random forest with 10-fold cross-validation to assign the index of feature significance in reducing impurity in the whole forest. The top features of UNSW-NB15 Dataset are ct_dst_src_ltm, ct_srv_dst, ct_dst_sport_ltm, ct_src_dport_ltm, ct_srv_src. Support vector machine with an accuracy of 82.11% outperformed Logistic Regression and Gradient Boost Machine in binary classification model for attack detection. For identifying the type of attack, the multi-classification model with Decision Tree C5.0, outperformed Naive Bayes and Support vector machine [20].

Peng et al., proposed Deep Neural Network(DNN)k with five hidden layers to identify attacks (Normal, DoS, Probe Categories, R2L, U2R) with NSL-KDD Dataset and compared the performance with Machine Learning models (Support Vector Machines, Random Forest, Linear Regression Models). DNN produced satisfactory results for identifying Normal, Dos, and Prob categories. SVM performed well in detecting Normal and four attacks. Random forest and linear regression also performed well in identifying network attacks [24].

The previous research on UNSW-NB15 dataset includes learning of machine learning and deep learning models on selected features, which decreases the performance of the model since the cardinality of the feature set is only 47 which is not all huge and the relevance of each feature is very significant. Regarding the deep neural network, works of literature are very limited and those works have addressed only a limited number of attacks. In this research four classical, three ensemble machine learning models, and deep multi-layer perceptron models are designed to identify network attacks.

3. Proposed Methodology

3.1. Problem Formulation

Let dataset $(A_1, A_2, ...A_9, N)$ consist of Analysis(A_1), DoS(A_2), Exploit(A_3), Fuzzers(A_4), Generic(A_5), Reconnaissance(A_6), Worms(A_7), Backdoor(A_8), Shellcode(A_9) and Normal(N) samples. Each samples S_i consisting of 49 Features $(f_1, f_2, ...F_{47}, T, C)$ where T is an attack type Label and A is Network Traffic Label. $T \in \{Analysis, DoS, Exploit, Fuzzers, Generic, Reconnaissance, Worms, Backdoor, Shellcode, Normal\}$ and $C \in \{Normal, Attack\}$. The problem is building a classification model that identifies any network sample S_j as an attack or normal. If the attack sample further classifies S_j to its attack type. A robust architecture for detecting network attacks is depicted in Fig 1.

3.2. Dataset

The proposed method has been experimented on UNSW-NB15 Dataset [22], which was created by the Australian Centre for Cyber Security, consisting of nine families of attacks and 49 features(5 Flow features, 13 basic features,

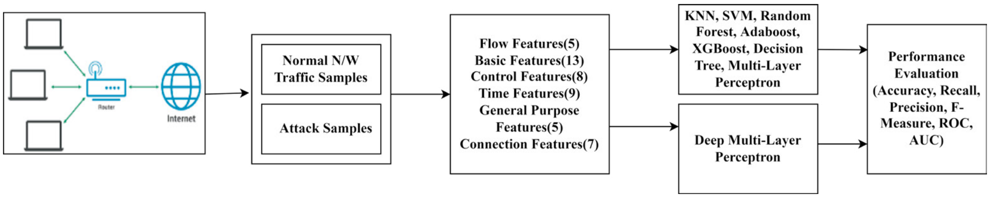
Router
Internet
Normal N/W Traffic Samples
Attack Samples
Flow Features(5)
Basic Features(13)
Control Features(8)
Time Features(9)
General Purpose Features(5)
Connection Features(7)
KNN, SVM, Random Forest, Adaboost, XGBoost, Decision Tree, Multi-Layer Perceptron
Deep Multi-Layer Perceptron
Performance Evaluation (Accuracy, Recall, Precision, F-Measure, ROC, AUC)

Fig. 1. Proposed architecture

eight control features, nine-time features, five additional generated general-purpose features, seven additional generated connection features, two labeled features). These features represent contemporary network traffic patterns collected from packet header, client to server, and server to client communications [21]. Dataset consist of 2540044 samples with Normal (2218761), Fuzzers(24246), Analysis(2677), Backdoors(2329), DoS(16353), Exploits(44525), Generic(215481), Reconnaissance(13987), Shellcode(1511) and Worms(174) network traffic.

3.3. Classification Models

The binary classification models used to identify network attacks are Support Vector Machine(SVM), Adaboost, XGBoost, Random Forest, K-Nearest Neighbour (KNN), Decision Tree(DT), Multi-Layer Perceptron(MLP), and Deep Multi-Layer Perceptron(Deep MLP) [26]. SVM is a statistical learning classifier that uses multidimensional hyper-planes to separate network attack patterns from normal traffic patterns [33]. SVM with linear kernel supports an extensive feature set and fast training. KNN is a memory-based classifier that predicts the class label of an unknown sample based on the majority class label of K-nearest neighbours [5]. DT classifier is a multistage decision-making model that suits numerical and nominal data [2]. It produces fast decisions as it consists of a limited number of nested simple conditional statements. A Random Forest classifier is an ensemble classification model that trains a set of CART trees by bagging to make predictions. This model randomly selects features to set a decision on nodes and uses out-of-bag error estimation and final class decision by averaging individual tree class assignment probability [4]. Adaboost builds a robust predictive model by improving several weaker models. It is the best out-of-the-box classifier with no tweaking parameters and is less prone to over-fitting [9]. Extreme Gradient(XG) is an ensemble classifier that applies boosting to a weak classifier by the paralleled implementation. It also supports Tree pruning, handling missing data, avoiding over-fitting, and hardware optimization [10].

Deep neural networks are vital in mapping input features to class labels. Two models, Multi-Layer Perceptron (MLP) and Deep MLP, are implemented to predict class labels with a back propagation weight adjustment algorithm. The parameter values of MLP are quasi-newton lbfgs for the optimizer, hidden neuron strength 15, penalty regularisation term parameter alpha with value 1e-5, and random state value 1. The proposed Deep MLP consists of 2 dense layers with Relu activation followed by a dense layer with a sigmoid activation function. A dense layer with Relu activation generates an accurate mixture of inputs from features, and sigmoid activation predicts the target class based on the outputs from previous layers [17].

3.4. Evaluation Metrics

The metrics used to evaluate the proposed system are Accuracy, Precision, Recall, F1-Measure, Receiver Operating Characteristic Curve(ROC), and Area under ROC (AUC) [8]. Confusion matrix is 2*2 matrix consisting of True positive(TP), False negative(FN), False positive(FP) and True negative(TN) values. True Positive represents the number of attack samples classified as attacks. False Negative represents the number of attack samples miss-classified as normal. False Positives represent the count of miss-classified normal samples, and true negative represents counts of normal samples classified as normal. Accuracy denotes the percentage of samples that are correctly classified. Recall denotes the ratio of correctly classified attacks to the total number of attack samples. Precision denotes actual attack samples in classified attacks. Qualitative and quantitative metrics F1-Measure represents the harmonic mean of precision and recall.

4. Results and discussion

The proposed network intrusion detection system was implemented on Ubuntu 20.04.4 with the support of Intel core 11^{th} generation i7 processor, 16GB RAM, and 1TB HDD. The machine learning and deep learning models are implemented in python with Keras 2.3.1 and TensorFlow 2.2.0 libraries.

4.1. Performance of Machine Learning Models

The performance of various machine learning models in network attack detection are shown in Table 1 and Fig 2. Decision Tree produced the best accuracy of 99.05% and F1-Measure of 0.99 for identifying network attacks. The results produced by SVM with an accuracy of 95.17% and F1-Measure 0.94 are also not negligible. The KNN models are trained for different K neighbors (2,3,4,5,6,7,8,9), and the best accuracy was 95.58% produced for K=7. The features used for model training are very relevant and can make much variance among attack and regular network traffic. So simple decision tree performs better than ensemble learning like bagging in random forest and boosting in Adaboost and XGBoost. Multidimensional hyperplane model SVM performs with a recall of 0.93, which shows the model's inefficiency in identifying attacks (True Positives). The performance of KNN shown in Table 2 which depicts that the models are not strong in identifying attacks and are more prone to classify regular traffic as attacks (false positives).

Table 1. Results: Network Attack detection Model

Model	Accuracy	Precision	Recall	F1-Measure
SVM	95.17	0.96	0.93	0.94
Decision Tree	99.05	0.99	0.99	0.99
Random Forest	98.96	0.99	0.99	0.99
Adaboost	97.87	0.98	0.97	0.98
XGBoost	98.08	0.98	0.97	0.98
MLP	97.47	0.98	0.96	0.97

Table 2. Results:KNN Results

Model	Accuracy	Precision	Recall	F1-Measure
2-NN	94.51	0.93	0.95	0.94
3-NN	95.47	0.95	0.95	0.95
4-NN	95.12	0.94	0.95	0.94
5-NN	95.56	0.95	0.95	0.95
6-NN	95.31	0.94	0.95	0.95
7-NN	95.58	0.95	0.94	0.95
8-NN	95.48	0.95	0.95	0.95
9-NN	95.57	0.95	0.94	0.95

4.2. Performance of Deep learning Models

Performance of the Deep Multi-Layer Perceptron shown in Table 3 is excellent with high accuracy of 98.44% and F1-Measure 0.98 for Adam optimizer and 80:20 Train-Test ratio. The performance of Adam is better than the Stochastic Gradient Descent optimizer. The Deep MLP model with adam optimizer produced the best result for the 80:20 Train-Test split as it represents natural system normal and attack network traffic ratio. The network anomaly detection system proposed by Moustafa et al., obtained an accuracy of 85.56% for Decision Tree and 81.34% for Artificial Neural Network [23]. But the proposed system produced an accuracy of 99.05% for the Decision Tree classifier and 97.47% for Multilayer Perceptron. The improved performance of the proposed system shows that all the features of the UNSW-NB15 dataset are very relevant and the application of feature selection in the dataset reduces the accuracy. Meftah et al. proposed NIDS with Random Forest feature selection and Recursive feature elimination [20]. After feature selection and elimination, classification with SVM produced an accuracy of 82.11%. The proposed system with SVM produced an accuracy of 95.17% without feature selection.

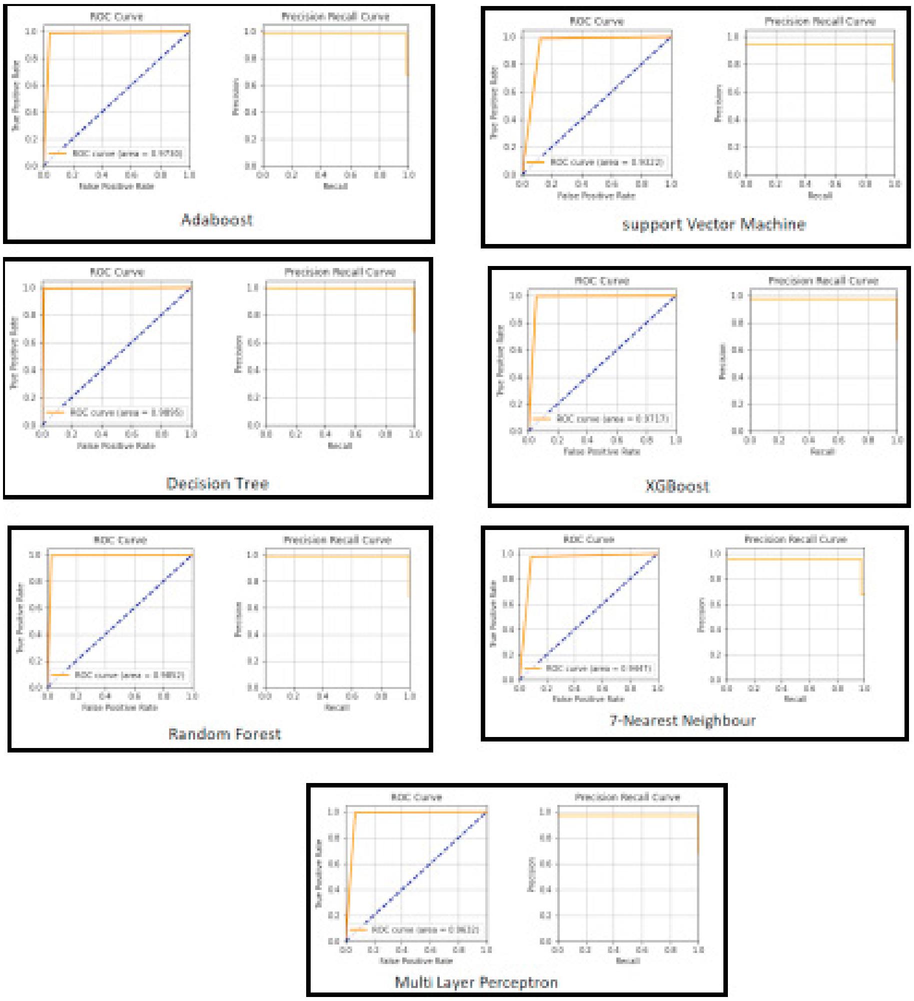

Fig. 2. ROC & Precision-Recall Curve(ML Models)

Table 3. Deep Multi Layer Perceptron Results

Optimizer	Train:Test	Accuracy	Precision	Recall	F1-Measure	AUC
SGD	90 : 10	98.19	0.98	0.98	0.98	0.976
SGD	80 : 20	97.68	0.98	0.97	0.97	0.971
SGD	70 : 30	97.99	0.98	0.98	0.98	0.975
SGD	60 : 40	97.82	0.98	0.97	0.97	0.972
ADAM	90 : 10	98.11	0.98	0.98	0.98	0.975
ADAM	80 : 20	98.44	0.98	0.98	0.98	0.981
ADAM	70 : 30	98.36	0.98	0.98	0.98	0.98
ADAM	60 : 40	98.35	0.98	0.98	0.98	0.979

4.3. *Performance of Attack Detection Models*

Performance of Machine learning models for identifying nine networks attacks are shown in Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11 and Table 12. Machine learning models are weak in identifying network attacks like analysis(Random Forest 23%), DoS(Random forest 35%), worms (XGBoost 46%), Backdoor (ADABoost 53%) and Shellcode (Random Forest 65%) with UNSW-NB15 features. XGBoost is the best model with 95% accuracy and 0.95 Recall for exploits detection. Random Forest is good in identifying Fuzzers attack with 90%

accuracy and generic attack with 99% accuracy. Performance of machine learning models for Reconnaissance attacks are satisfactory with 79% accuracy for Decision Tree. The performance of Random forest for identifying Fuzzers attack is better than the models proposed by [20] [15].

Table 4. Analysis Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.24	0.23	0.23	0.23
XGBoost	0.73	0.19	0.3	0.19
ADABoost	0.04	0.01	0.01	0.01
MLP	0.63	0.12	0.21	0.12
Decision Tree	0.5	0.12	0.19	0.12
7-NN	0.48	0.16	0.24	0.16

Table 5. DoS Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.37	0.35	0.36	0.35
XGBoost	0.41	0.03	0.05	0.03
ADABoost	0.09	0.02	0.04	0.02
MLP	0.39	0.09	0.14	0.09
Decision Tree	0.00	0.00	0.00	0.00
7-NN	0.29	0.29	0.29	0.29

Table 6. Exploits Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.74	0.79	0.76	0.79
XGBoost	0.61	0.95	0.75	0.95
ADABoost	0.56	0.35	0.43	0.35
MLP	0.61	0.89	0.72	0.89
Decision Tree	0.53	0.92	0.67	0.92
7-NN	0.62	0.73	0.67	0.73

Table 7. Fuzzers Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.9	0.9	0.9	0.90
XGBoost	0.93	0.89	0.91	0.89
ADABoost	0.68	0.6	0.64	0.60
MLP	0.86	0.84	0.85	0.84
Decision Tree	0.94	0.11	0.19	0.11
7-NN	0.76	0.81	0.78	0.82

Table 8. Generic Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.99	0.98	0.99	0.98
XGBoost	1.00	0.98	0.99	0.98
ADABoost	0.03	0.00	0.00	0.00
MLP	1.00	0.98	0.99	0.98
Decision Tree	1.00	0.98	0.99	0.98
7-NN	1.00	0.98	0.99	0.98

Table 9. Reconnaissance attacks Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.79	0.76	0.78	0.76
XGBoost	0.90	0.75	0.82	0.75
ADABoost	0.54	0.45	0.49	0.45
MLP	0.73	0.71	0.72	0.71
Decision Tree	0.41	0.79	0.54	0.79
7-NN	0.66	0.51	0.58	0.51

Table 10. Worms Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.73	0.21	0.32	0.21
XGBoost	0.90	0.46	0.61	0.46
ADABoost	0.00	0.08	0.00	0.08
MLP	0.00	0.00	0.00	0.00
Decision Tree	0.00	0.00	0.00	0.00
7-NN	1.00	0.05	0.10	0.00

Table 11. Backdoor Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.23	0.22	0.23	0.22
XGBoost	0.73	0.23	0.35	0.23
ADABoost	0.04	0.53	0.07	0.53
MLP	0.00	0.00	0.00	0.00
Decision Tree	0.00	0.00	0.00	0.00
7-NN	0.51	0.04	0.07	0.04

Table 12. Shell code Attack Detection Results

Model	Precision	Recall	F1-Measure	Accuracy
Random Forest	0.69	0.65	0.67	0.65
XGBoost	0.66	0.49	0.56	0.49
ADABoost	0.71	0.29	0.41	0.29
MLP	0.70	0.20	0.31	0.20
Decision Tree	1.00	0.01	0.02	0.01
7-NN	0.51	0.14	0.22	0.14

5. Conclusion

This paper discusses machine learning and deep learning models for NIDS. Decision Tree produced the best performance with 99.05% compared to ensemble models Random Forest, Adaboost, and XGBoost. Features of the UNSW-NB15 dataset are very relevant and robust in identifying network attacks. The machine learning model KNN with 7 neighbors produced an accuracy of 95.58%. The deep learning model with ADAM optimizer and 80:20 Train-Test split produced an accuracy of 98.44%with high True Positives and False low negatives. Machine learning models are also effective in identifying attacks type like Exploits, Fuzzers, Generic attacks, and Reconnaissance. Features of UNSW-NB15 are not robust in producing variance for DoS, Worms, Backdoors, and Shellcode attacks. As a future direction, Deep learning models like one-dimensional Convolutional Neural networks (CNN) can be directly trained with strings present in network traffic pcap files to identify attacks. CNN eliminates the requirement of complex feature engineering by automatically extracting features from pcap files with its convolutional layers. One emerging method for network attacks and malware detection is using Visualization representation. The network pcap files binary can be represented in greyscale, RGB, or Markov image. These images are used to train two-dimensional CNN, which can detect network attacks with much covariance.

References

[1] Almutairi, A.H., Abdelmajeed, N.T., 2017. Innovative signature based intrusion detection system: Parallel processing and minimized database, in: 2017 International Conference on the Frontiers and Advances in Data Science (FADS), IEEE. pp. 114–119.

[2] Ammar, A., et al., 2015. A decision tree classifier for intrusion detection priority tagging. Journal of Computer and Communications 3, 52.

[3] Arce, I., 2004. The shellcode generation. IEEE security & privacy 2, 72–76.

[4] Belgiu, M., Drăguţ, L., 2016. Random forest in remote sensing: A review of applications and future directions. ISPRS journal of photogrammetry and remote sensing 114, 24–31.

[5] Cunningham, P., Delany, S.J., 2021. k-nearest neighbour classifiers-a tutorial. ACM Computing Surveys (CSUR) 54, 1–25.

[6] Dada, E.G., Bassi, J.S., Chiroma, H., Adetunmbi, A.O., Ajibuwa, O.E., et al., 2019. Machine learning for email spam filtering: review, approaches and open research problems. Heliyon 5, e01802.

[7] De Canniere, C., Biryukov, A., Preneel, B., 2006. An introduction to block cipher cryptanalysis. Proceedings of the IEEE 94, 346–356.

[8] Dhanya, K., Dheesha, O., Gireesh Kumar, T., Vinod, P., 2020. Detection of obfuscated mobile malware with machine learning and deep learning models, in: Symposium on Machine Learning and Metaheuristics Algorithms, and Applications, Springer. pp. 221–231.

[9] Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences 55, 119–139.

[10] Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics , 1189–1232.

[11] Gandhi, M., Srivatsa, S., 2008. Detecting and preventing attacks using network intrusion detection systems. International Journal of Computer Science and Security 2, 49–60.

[12] Garuba, M., Liu, C., Fraites, D., 2008. Intrusion techniques: Comparative study of network intrusion detection systems, in: Fifth International Conference on Information Technology: New Generations (itng 2008), IEEE. pp. 592–598.

[13] Gascon, H., Orfila, A., Blasco, J., 2011. Analysis of update delays in signature-based network intrusion detection systems. Computers & Security 30, 613–624.

[14] Hubballi, N., Suryanarayanan, V., 2014. False alarm minimization techniques in signature-based intrusion detection systems: A survey. Computer Communications 49, 1–17.

[15] Jing, D., Chen, H.B., 2019. Svm based network intrusion detection for the unsw-nb15 dataset, in: 2019 IEEE 13th international conference on ASIC (ASICON), IEEE. pp. 1–4.

[16] Kumar, V., Sangwan, O.P., 2012. Signature based intrusion detection system using snort. International Journal of Computer Applications & Information Technology 1, 35–41.

[17] Lee, B., Amaresh, S., Green, C., Engels, D., 2018. Comparative study of deep learning models for network intrusion detection. SMU Data Science Review 1, 8.

[18] Lee, C.H., Su, Y.Y., Lin, Y.C., Lee, S.J., 2017. Machine learning based network intrusion detection, in: 2017 2nd IEEE International conference on computational intelligence and applications (ICCIA), IEEE. pp. 79–83.

[19] Li, P., Salour, M., Su, X., 2008. A survey of internet worm detection and containment. IEEE Communications Surveys & Tutorials 10, 20–35.

[20] Meftah, S., Rachidi, T., Assem, N., 2019. Network based intrusion detection using the unsw-nb15 dataset. International Journal of Computing and Digital Systems 8, 478–487.

[21] Moustafa, N., Slay, J., 2015a. The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems, in: 2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS), IEEE. pp. 25–31.

[22] Moustafa, N., Slay, J., 2015b. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 military communications and information systems conference (MilCIS), IEEE. pp. 1–6.

[23] Moustafa, N., Slay, J., 2016. The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Information Security Journal: A Global Perspective 25, 18–31.

[24] Peng, Y., Su, J., Shi, X., Zhao, B., 2019. Evaluating deep learning based network intrusion detection system in adversarial environment, in: 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), IEEE. pp. 61–66.

[25] Rieger, P., Nguyen, T.D., Miettinen, M., Sadeghi, A.R., 2022. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. arXiv preprint arXiv:2201.00763 .

[26] Sugunan, K., Gireesh Kumar, T., Dhanya, K., 2018. Static and dynamic analysis for android malware detection, in: Advances in Big Data and Cloud Computing. Springer, pp. 147–155.

[27] Thanh, H.N., Van Lang, T., 2020. Evaluating effectiveness of ensemble classifiers when detecting fuzzers attacks on the unsw-nb15 dataset. Journal of Computer Science and Cybernetics 36, 173–185.

[28] Uma, M., Padmavathi, G., 2013. A survey on various cyber attacks and their classification. Int. J. Netw. Secur. 15, 390–396.

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

[30] Vinayakumar, R., Soman, K., Poornachandran, P., 2017. Applying convolutional neural network for network intrusion detection, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE. pp. 1222–1228.

[31] Wang, W., Jian, S., Tan, Y., Wu, Q., Huang, C., 2022. Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions. Computers & Security 112, 102537.

[32] Yang, H., Cheng, L., Chuah, M.C., 2019. Deep-learning-based network intrusion detection for scada systems, in: 2019 IEEE Conference on Communications and Network Security (CNS), IEEE. pp. 1–7.

[33] Yang, Y., Li, J., Yang, Y., 2015. The research of the fast svm classifier method, in: 2015 12th international computer conference on wavelet active media technology and information processing (ICCWAMTIP), IEEE. pp. 121–124.