2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG) | 979-8-3503-5815-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/SEB4SDG60871.2024.10629939

Comparison of Random Forest, K-Nearest Neighbor, and Support Vector Machine Classifiers for Intrusion Detection System

Emmanuel Chinanu Uwazie
Department of Computer Science
Nasarawa State University Keffi
Nasarawa, Nigeria
uwazieemmanuel@yahoo.com

Afolayan A. Obiniyi
Department of Computer Science
Ahmadu Bello University Zaria
Kaduna, Nigeria
aaobiniyi@gmail.com

Morufu Olalere
Department of Computer Science
Federal University of Technology Minna
Niger, Nigeria
lerejide@gmail.com

Perpetua N. Achi
Department of Computer Science
Nasarawa State University Keffi
Nasarawa, Nigeria
achiperpetua@gmail.com

Abstract— **Earlier classification investigations identified Random Forest (RF), k-Nearest Neighbor (kNN), and Support Vector Machine (SVM) as leading non-parametric classifiers capable of achieving high accuracies. Nonetheless, there has been limited research comparing the performances of these classifiers across various standard intrusion detection datasets. In this research, the performances of the RF, KNN, and SVM classifiers are examined and compared when they are used for intrusion detection on various standard intrusion detection datasets. For each of the algorithms, parameter tunings produced various accuracies on each of the datasets. The parameter with the highest accuracy for each algorithm is compared with its counterpart in other algorithms on the same dataset. The experimental results show that the KNN Intrusion Detection System outperformed other approaches on all the datasets, with accuracies of 0.999928556, 0.996921593 and 0.971580496 on NSL-KDD, CICIDS2017 and CICIDS2018 datasets, respectively. Consequently, the obtained results of KNN present better performances in terms of precision, recall and f1-score on the various network traffic classes when compared to the other algorithms. As shown in this research, the high performances of these machine learning algorithms show that they can be deployed in the field for intrusion detection.**

Keywords—NSL-KDD; CICIDS2017, CICIDS2018, Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), classification algorithms, Intrusion Detection System.

I. Introduction

With the growing dependence on the Internet, there has been a rising apprehension regarding cyber breaches and attacks in recent years. An Intrusion Detection System (IDS) serves as a means to identify such attacks [1]. IDS is categorized into two detection methods: anomaly-based detection and Signature-based (or Misuse detection) detection. The former monitors system behavior to detect and flag abnormalities in activities, albeit susceptible to false positives [2]. Signature-based IDS relies on predefined patterns, known as fingerprints or signatures of previously successful attacks, stored within the system. These signatures are utilized to identify intrusions when new packets enter the system. However, this method's drawback lies in its dependency on existing knowledge, rendering it ineffective against new attack types. Consequently, constant updates to the IDS knowledge base are necessary [3]. Despite the efficacy of current Intrusion Detection methods, there is a growing imperative to enhance existing approaches or introduce novel ones [4].

Therefore, it is imperative to design IDSs that are robust, efficient and accurate. Many classical machine learning techniques have been applied in IDS [5]. These approaches include: K-Nearest-Neighbor (KNN), Random Forest (RF), Support Vector Machines (SVM) and Naive Bayes (NB) [6]. In order to contribute to the improvement of current IDS, this work aims to evaluate the performance of three existing conventional Machine Learning (ML) algorithms and how these could be utilized in detecting evolving intrusion attacks. To gain further insight into where each algorithm provides the best performance, the algorithms have been trained and tested on several benchmark datasets and their performance have been evaluated against a number of standard metrics.

The rest of the paper has been organized as follows: Section II discusses the datasets used for the study. An overview of the steps performed during the preparation of the datasets is provided in section III. In Section IV, an overview of the ML techniques adapted for intrusion detection systems is provided. The description of the evaluation metrics is given in Section V and the comparison of different ML algorithms is given in Section VI. Finally, Section VII concludes the paper.

II. Description of Data

A. NSL-KDD

In this study, the NSL-KDD dataset was used. It was developed after removing additional and duplicate records from the University of New Brunswick KDD'99 training and testing data to get a clean and refined dataset [7]. There are four categories of attacks in NSL-KDD, which are further classified as 37, of which 27 are used for the testing dataset, and 23 are used for the training dataset for further experimentation.

All the attacks classified under the four categories are: DOS, Probe, R2L and U2R. NSL-KDD contains 41 Features.

B. CICIDS2017

The Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset of 2017 is a widely used dataset [8]. The dataset is a labelled dataset for IDS research. CICIDS2017 contains a total of 2,830,743 records, including 2,273,097 records for normal (benign) traffic and 557,646 records for abnormal (malign) traffic data representing a realistic network environment. It is a high-dimensional dataset, which contains 84 feature columns and 1 label column. The dataset contains features such as flow statistics, packet headers, and payload data. Network traffic include Brute Force, DDoS, and Web Attacks Benign, FTP-Patator, WebAttack, Dos-Hulk, SSH-Patator, Infltration, PortScan, Dos-slowlories, Heartbleed, DDos, Dos-slowhttptest, Dos-GoldenEye and Bot. The feature set of CICIDS2017 dataset include IAT, Forward/backward traffic packets, Flags and Flow rate.

C. CICIDS2018

An evolution of CICIDS2017, the CICIDS2018 dataset extends the variety of attacks, enhances the diversity of traffic, and refines features. CICIDS2018 is a recent intrusion detection dataset that is big data, available to the public and covers a wide range of modern attack types [9]. The whole dataset contains about 16,000,000 instances. The dataset includes various features extracted from users' network events. It encompasses a broader range of attack scenarios and network traffic variations. The attack classes include Brute Force, DoS, Heartbleed, Web Attack, Infiltration, Botnet, PortScan and DDoS attacks. The network traffic distributions of CICIDS2018 dataset are: Benign, Bot, Brute force, DDoS, DoS, Infiltration and Web attacks.

III. DATA PREPARATTION

Before the datasets were analysed, it was first prepared by going through feature selection and normalization processes.

A. Feature Selection

Feature selection is a crucial step in enhancing the efficiency of machine learning models. Each dataset provides a plethora of attributes, some of which might not contribute significantly to the detection process. Prior to model training, we perform feature selection to identify the most relevant features from the datasets while reducing dimensionality. This ensures that the models focus on the most informative attributes, leading to classification efficiency.

B. Normalization

The datasets come with pre-extracted features that undergo preprocessing and normalization to ensure uniformity leading to improved classification accuracy.

IV. MACHINE LEARNING TECHNIQUES

ML techniques play a crucial role in Intrusion Detection Systems (IDS) due to their effectiveness in distinguishing between abnormal and normal network traffic [10]. Classical machine learning models such as KNN, RF, SVM, etc., have been widely used in IDS.

We employ a supervised classification approach to train and evaluate the models. All three techniques, KNN, RF, and SVM, are used for classification tasks in intrusion detection. Each network architecture is trained on the datasets, using a split of training and testing instances. Each architecture's structure and characteristics influence their approach in learning to differentiate between normal and attack traffic patterns, with the ultimate goal of accurately classifying network traffic into different categories: normal or various attack types.

A. K-Nearest Neighbor (KNN)

The K-nearest neighbors (KNN) algorithm serves purposes in both classification and regression tasks. It operates on the assumption that items with similarities tend to be proximate to each other. In classification, KNN employs a majority voting mechanism among the k-nearest neighbors, while in regression, it computes the mean of the k nearest data points as the output. To classify a new instance, KNN calculates the distance between the item to be classified and all other training data items. The majority class among the k neighbors is then assigned as the classification result. The value of k represents the number of nearest neighbors considered for classification. Typically, various values are experimented with to ascertain the optimal k value, often favoring odd numbers based on conventional wisdom.

B. Random Forest (RF)

Random Forest is a supervised machine learning method applicable to both classification and regression tasks. It operates on the principle of ensemble learning, wherein multiple decision trees are grown instead of a single one to address complex problems and enhance model performance. Each tree in the Random Forest is trained on a randomly chosen subset of the original dataset. The predictions from all trees within the forest are combined, and the majority vote determines the final classification decision. Increasing the number of trees in the forest enhances accuracy and mitigates overfitting issues.

C. Support Vector Machine Algorithm (SVM)

SVM, a supervised machine learning algorithm, is applicable to both classification and regression tasks. Its primary function is to classify data by establishing a hyperplane or a line that separates two classes within a dataset. To determine the optimal line for data separation, SVM computes the distances between points belonging to different classes and identifies the points closest to the line, known as support vectors. The hyperplane is then selected to maximize the margin between itself and the support vectors.

V. PERFORMANCE EVALUATION METRICS

A. Confusion Matrix

The matrix presented in TABLE I comprises four potential outcomes, illustrating the performance of the algorithm in classification tasks. It details the counts of correctly classified and incorrectly classified records. Given the datasets' lack of complete balance, relying solely on metrics such as accuracy can lead to misleading conclusions [11]. Confusion matrix consists of four features, namely: false Positive (FP), false Negative (FN), True Positive (TP) and True Negative (TN) values.

- True Positives (TP): Positive Samples that were already classified as positive.
- False Positives (FP): Negative Samples that were already identified as positive.
- True Negatives (TN): Negative Samples that were already detected as negative.
- False Negatives (FN): Positive Samples that have been counted as negative [12].

TABLE I. CONFUSION MATRIX

		Predicted traffic	
		Attack traffic	Normal traffic
Actual traffic	Normal traffic	TN	FP
	Attack traffic	FN	TP

Four quality metrics, namely Accuracy, Precision, Recall, and F1-Score, were employed to assess the performance of ML techniques. Samples corresponding to the predicted class are designated as positive (represented by ‘1’), whereas other samples are considered negative (represented by '0’). The formulas for all performance measures are provided in Equations (1-4).

1) Accuracy:

Accuracy, denoted by equation [13] or referred to as the correct rate, represents the proportion of correctly classified anomalies and normal cases to the total number of instances. The intrusion detection system's accuracy is determined through the utilization of a confusion matrix, as depicted in Equation 1:

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

2) Precision (Pre.):

Precision is computed as the proportion of true positives relative to the sum of true positives and false positives [14]. Its calculation involves utilizing a confusion matrix, illustrated by Equation 2:

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

3) Recall (Rec.):

Recall, also known as sensitivity, is determined as the fraction of true positives divided by the total of true positives and false negatives [15]. Its computation involves the use of a confusion matrix, outlined in Equation 3:

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

4) F-1 Score (f1.):

The F1-score balances the precision and recall values, suggesting how similar the predicted classes are to the actual classes [16]. Compared to accuracy, the F1-Score is more suitable for evaluating the detection performance of imbalanced datasets. Its calculation is as shown in Equation 4:

$$F1 - score = \frac{(2+TP)}{(2*TP+FP+FN)} \quad (4)$$

VI. COMPARISON

The three algorithms, KNN, RF, and SVM algorithm were used to predict classes on the NSL-KDD dataset.

For RF, 50 n_estimators has the highest accuracy of 0.999872989 as shown in Fig. 1.

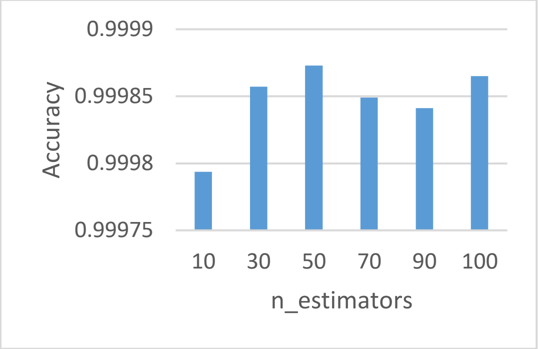

Fig. 1. Different n_estimators of RF on NSL-KDD.

The Pre., Rec., f1. and support values for RF with 50 n_estimators on NSL-KDD dataset is shown in Table II.

TABLE II. RESULTS OF RF ON NSL_KDD

Class	Pre.	Rec.	f1.	support
1	1.00	1.00	1.00	11656
2	1.00	1.00	1.00	45927
3	1.00	0.96	0.98	52
4	1.00	1.00	1.00	995
5	1.00	1.00	1.00	67343

For KNN, 1 n_neighbors has the highest accuracy of 0.999928556 as shown in Fig. 2.

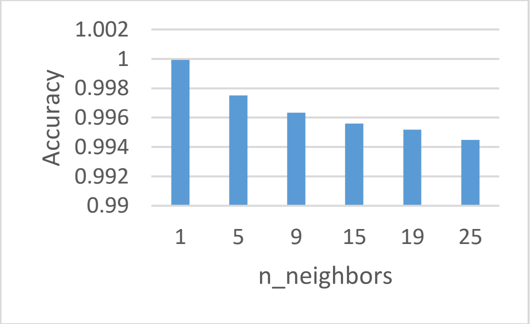

Fig. 2. Different n_neighbors of KNN on NSL-KDD.

The Pre., Rec., f1. and support values for KNN with 1 n_neighbors on NSL-KDD dataset is shown in Table III.

TABLE III. RESULTS OF KNN ON NSL_KDD

Class	Pre.	Rec.	f1.	support
1	1.00	1.00	1.00	11656
2	1.00	1.00	1.00	45927

3	0.98	1.00	0.99	52
4	1.00	1.00	1.00	995
5	1.00	1.00	1.00	67343

For SVM, rbf Kernel has the highest accuracy of 0.995546665 as shown in Fig. 3.

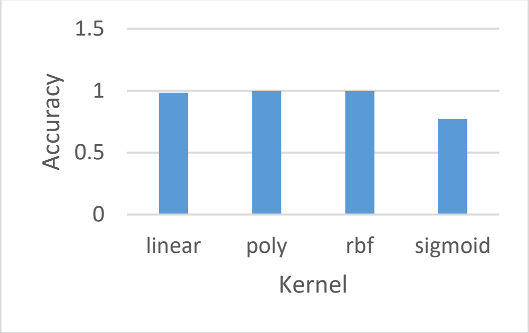

Fig. 3. Different kernels of SVM on NSL-KDD.

`The Pre., Rec., f1. and support values for SVM with rbf Kernel on NSL-KDD dataset is shown in Table IV.

TABLE IV. RESULTS OF SVM ON NSL_KDD

Class	Pre.	Rec.	f1.	support
1	0.98	0.99	0.99	11656
2	1.00	1.00	1.00	45927
3	0.97	0.71	0.82	52
4	0.94	0.82	0.88	995
5	1.00	1.00	1.00	67343

The three algorithms, RF, KNN and SVM algorithm were also used to predict classes on the CICIDS2017 dataset.
For RF, 90 n_estimators has the highest accuracy of 0.995944348 as shown in Fig. 4.

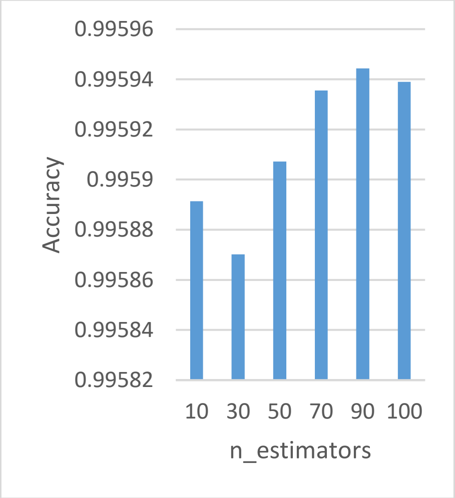

Fig. 4. Different n_estimators of RF on CICIDS2017.

The Pre., Rec., f1. and support values for RF with 50 n_estimators on CICIDS2017 dataset is shown in Table V.

TABLE V. RESULTS OF RF ON CICIDS2017

Class	Pre.	Rec.	f1.	support
1	1.00	1.00	1.00	454547
2	1.00	1.00	1.00	46005
3	0.99	1.00	1.00	31869
4	1.00	1.00	1.00	25586
5	1.00	0.99	0.99	2010
6	0.99	1.00	0.99	1574
7	1.00	0.51	0.68	1197
8	0.99	0.91	0.95	1125
9	0.99	0.48	0.65	1108
10	0.96	0.38	0.54	397
11	0.57	0.76	0.65	310
12	0.40	0.19	0.26	134
13	0.00	0.00	0.00	10
14	0.00	0.00	0.00	4
15	1.00	1.00	1.00	1

For KNN, 5 n_neighbors has the highest accuracy of 0.996921593 as shown in Fig. 5.

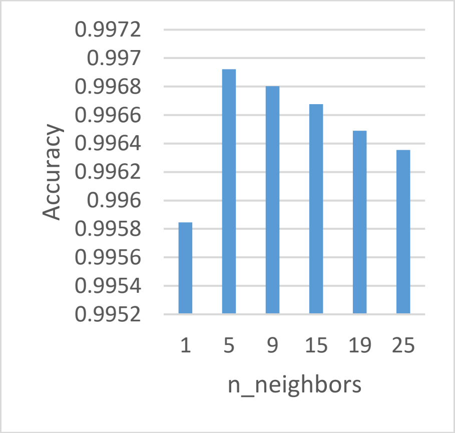

Fig. 5. Different n_neighbors of KNN on CICIDS2017.

The Pre., Rec., f1. and support values for KNN with 5 n_neighbors on CICIDS2017 dataset is shown in Table VI.

TABLE VI. RESULTS OF KNN ON CICIDS2017

Class	Pre.	Rec.	f1.	support
1	1.00	1.00	1.00	454640
2	1.00	1.00	1.00	45850
3	0.99	1.00	1.00	31662
4	1.00	1.00	1.00	25805
5	0.99	0.99	0.99	2094
6	1.00	0.99	0.99	1589
7	0.99	0.99	0.99	1180
8	0.99	0.99	0.99	1135
9	0.96	0.47	0.63	1056
10	0.83	0.67	0.74	402
11	0.54	0.75	0.63	310
12	0.41	0.33	0.36	135
13	1.00	0.30	0.46	10
14	0.00	0.00	0.00	4
15	1.00	1.00	1.00	5

For SVM, poly Kernel has the highest accuracy of 0. 0.984220385.

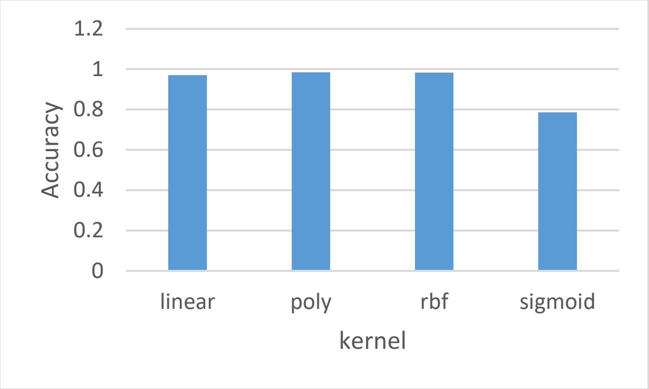

Fig. 6. Different kernels of SVM on CICIDS2017.

The three algorithms, RF, KNN and SVM algorithm were also used to predict classes on the CICIDS2018 dataset.

For RF, 70 n_estimators has the highest accuracy of 0.970445802 as shown in Fig. 6.

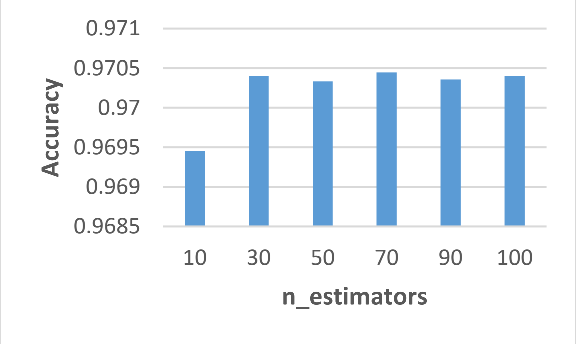

Fig. 7. Different n_estimators of RF on CICIDS2018.

The Pre., Rec., f1. and support values for RF with 70 n_estimators on CICIDS2018 dataset is shown in Table VII.

TABLE VII. RESULTS OF RF ON CICIDS2018

Class	Pre.	Rec.	f1.	support
0	0.97	1.00	0.99	30411
1	1.00	1.00	1.00	6334
2	0.98	0.88	0.93	3289
3	0.83	0.97	0.89	1860
4	1.00	1.00	1.00	1470
5	0.37	0.07	0.11	824
6	0.00	0.00	0.00	2

For KNN, 15 n_neighbors has the highest accuracy of 0.971580496 as shown in Fig. 7.

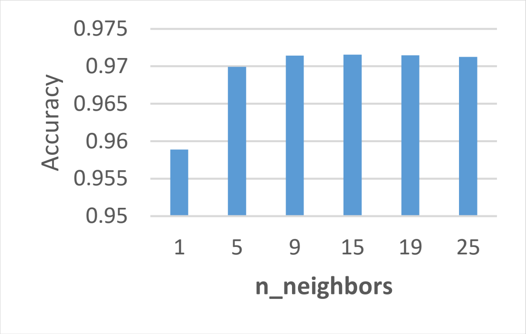

Fig. 8. Different n_neighbors of KNN on CICIDS2018.

The Pre., Rec., f1. and support values for KNN with 15 n_neighbors on CICIDS2018 dataset is shown in Table VIII.

TABLE VIII. RESULTS OF KNN ON CICIDS2018

Class	Pre.	Rec.	f1.	support
0	0.97	1.00	0.99	30583
1	1.00	1.00	1.00	6314
2	0.98	0.89	0.93	3179
3	0.84	0.97	0.90	1860
4	1.00	1.00	1.00	1445
5	0.51	0.03	0.06	810
6	0.00	0.00	0.00	4

For SVM, poly Kernel has the highest accuracy of 0.96324545.

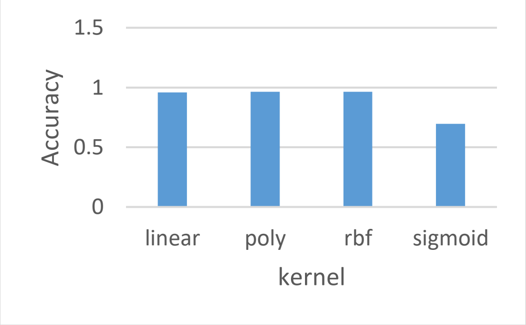

Fig. 9. Different kernels of SVM on CICIDS2018.

The Pre., Rec., f1. and support values for SVM with poly Kernel on CICIDS2018 dataset is shown in Table IX.

TABLE IX. RESULTS OF SVM ON CICIDS2018

Class	Pre.	Rec.	f1.	support
0	0.97	1.00	0.99	30574
1	1.00	1.00	1.00	6213
2	0.95	0.82	0.88	3220
3	0.76	0.94	0.84	1909
4	0.95	1.00	0.98	1464
5	0.00	0.00	0.00	799
6	1.00	0.33	0.50	6

VII. CONCLUSION

The implementation, evaluation, and comparison of machine learning algorithms for the classification of intrusion detection datasets were conducted. Three distinct datasets with varying sizes for training and testing samples were utilized. Across all datasets, the classification results demonstrated high accuracy levels, ranging approximately from 96% to 99%. In the NSL-KDD dataset, K-nearest neighbors (KNN) exhibited the highest performance with an accuracy of 0.999928556, followed closely by Random Forest (RF) with an accuracy of 0.999872989, while Support Vector Machine (SVM) displayed comparatively lower performance with an accuracy of 0.995546665. Similarly, for the NSL-KDD dataset, KNN demonstrated the highest performance with an accuracy of 0.996921593, followed by RF with an accuracy of 0.995944348, while SVM exhibited the lowest performance with an accuracy of 0.984220385. In the case of the CICIDS2018 dataset, KNN outperformed other algorithms with an accuracy of 0.971580496, followed closely by RF with an accuracy of 0.970445802, while SVM displayed the lowest performance with an accuracy of 0.96324545.

REFERENCES

[1] KA Tait, JS Khan, F Alqahtani, AA Shah, FA Khan, MU Rehman, W Boulila, and J Ahmad, Intrusion Detection using Machine Learning Techniques: An Experimental Comparison, in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*. IEEE, 2021, pp. 1-10.

[2] C. Chio and D. Freeman, Machine Learning and Security: Protecting Systems with Data and Algorithms. " O'Reilly Media, Inc.", 2018.

[3] M. A. khan, M. A. Khan, S. Latif, A. A. Shah, M. U. Rehman, W. Boulila, M. Driss, and J. Ahmad, "Voting classifier-based intrusion detection for iot networks," in 2021 2nd International Conference of Advance Computing and Informatics (ICACIN). Springer, 2021.

[4] Z. E. Huma, S. Latif, J. Ahmad, Z. Idrees, A. Ibrar, Z. Zou, F. Alqahtani, and F. Baothman, "A hybrid deep random neural network for cyberattack detection in the industrial internet of things," IEEE Access, vol. 9, pp. 55 595–55 605, 2021.

[5] Ahmed M. Mahfouz, Deepak Venugopal, and Sajjan G. Shiva. Comparative analysis of ml classifiers for network intrusion detection. In ICICT, 2019.

[6] S.M. Kasongo and Y. Sun, A Deep Long Short-Term Memory based classifier for Wireless Intrusion Detection System, ICT Express 6 (2020) 98–103.

[7] Z. K. Ibrahim and M. Y. Thanon, "Performance Comparison of Intrusion Detection System Using Three Different Machine Learning Algorithms," 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2021, pp. 1116-1124, doi: 10.1109/ICICT50816.2021.9358775.

[8] J. Qin, X. Han, C. Wang, Q. Hu, B. Jiang, C. Zhang, and Z. Lu, Network Traffic Classification Based on SD Sampling and Hierarchical Ensemble Learning, Security and Communication Networks, (2023) 1:1-16.

[9] J.L. Leevy, & T.M. Khoshgoftaar. A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data. Journal of Big data. (2020) 7, 104.

[10] Mbow, M., Koide, H., Sakurai, K., (2022). Handling class Imbalance problem in Intrusion Detection System based on deep learning, International Journal of Networking and Computing. 12(2): 467–492.

[11] Akshay Kumaar M, Samiayya D, Vincent PMDR, Srinivasan K, Chang C-Y and Ganesh H (2022) A Hybrid Framework for Intrusion Detection in Healthcare Systems Using Deep Learning. Front. Public Health 9:824898. doi: 10.3389/fpubh.2021.824898.

[12] H. Wang, Z. Cao, and B. Hong, "A network intrusion detection system based on convolutional neural network," J. Intell. Fuzzy Syst., vol. 38, no. 6, pp. 7623–7637,2020.

[13] Singh K, Mahajan A, Mansotra V. Using recursive feature elimination and fisher score with convolutional neural network for identifying port scan attempts. In: Smart Trends in Computing and Communications. Singapore: Springer (2022). p. 551–60. doi: 10.1007/978-981-16-4016-2_52

[14] Tandon A, Sharma R, Sodhiya S, Vincent PM. QR code based secure OTP distribution scheme for authentication in net-banking. Int J Eng Technol. (2013) 5:0975–4024.

[15] Tervoort T, De Oliveira MT, Pieters W, Van Gelder P, Olabarriaga SD, Marquering H. Solutions for mitigating cybersecurity risks caused by legacy software in medical devices: a scoping review. IEEE Access. (2020) 8:84352–61.

[16] Thamilarasu G, Odesile A, Hoang A. An intrusion detection system for internet of medical things. IEEE Access. (2020) 8:181560–76.