Comparative Performance Analysis of Both Traditional and Ensemble Machine Learning for Intrusion Detection System

John OjoAjayi[1], Adesina Simon Sodiya[2], Mustapha AminuBagiwa[3], Ismaili Idris Sinan[4]
[1,4]*Department of Cyber Security, National Open University Nigeria (ACETEL), Abuja, Nigeria.*
[2]*Department of Computer Science, Federal University of Agriculture, Abeokuta, Nigeria*
[3]*Department of Computer Science, Ahmadu Bello University, Zaria, Nigeria*

***Abstract*—The increasing complexity of cyberattacks makes it more challenging to use traditional machine learning (ML) techniques to reliably identify intrusions. Failure to accurately detect intrusions could lead to a decline in the confidence placed in security services. This paper presents both ensemble machine learning methods, including Random Forest (RF), Light Gradient Boosting Machine (LGBM), and Extreme Gradient Boosting (XGBoost), as well as traditional machine learning techniques, including K-Nearest Neighbors (KNN), Naive Bayes (NB), and Decision Tree (DT), using the UNSW-NB15 dataset from the Australian Centre for Cyber Security's Cyber Range Lab to classify data as either intrusion or normal. The findings of this study show that the accuracy of NB is 51.38%, KNN is 89.75%, DT is 89.57%, RF is 91.51%, LGBM is 91.34%, and XGBoost is 91.57%. These results indicate that the XGBoost classifier performed the best in detecting intrusions within the scope of this study, outperforming the other five models considered. This research demonstrates that the XGBoost ensemble strategy exhibited exceptional performance, surpassing both traditional machine learning algorithms and other ensemble methods.**

***Index Terms*—decision tree, ids, k-nearest neighbour, lgbm, machine learning, naïve bayes, random forest, and XGBoost.**

I. INTRODUCTION

As daily activities became more dependent on information sharing technologies, degrees of entry to these systems and unobstructed access to user activities became critical (Zhiqiang el at, 2021). The increasing sophistication of cyberattacks in recent years emphasizes the necessity for cybersecurity reinforcement. As a result, more businesses are placing a premium on cyber security and making concerted attempts to secure their networks. Taking a casual approach to cyber security may prove to be a disaster rather than waiting for a security breach to occur before taking action. Effective IDSs are needed to ensure the confidentiality, availability, and integrity of government, individual, and organization infrastructures as the threat landscape broadens and hackers shift their focus to remote workers' systems (Ben, 2016).

In this context, the need for IDS is more important than ever in protecting company networks and endpoint equipment against complex threats. The increased threats of cyber incidents to organizations and the rise in the quantity of connected devices within enterprises are the two key causes driving the adoption of IDS/IPS from 2019 to 2025 (Venkat, 2021). With the use of ML algorithms, new attacks may be automatically identified and classified as they happen. The system can continually learn and adjust to changing attack strategies by utilizing ML algorithms, which improves its capacity to recognize new and previously unidentified threats. By identifying abnormalities in real-time, this strategy also provides a proactive defensive mechanism, enabling prompt reaction and the mitigation of future security breaches (Abdulrahaman and Alhassan 2018).

As stated by the Office for National Statistics (ONS) of the United Kingdom (UK), 96 percent of homes in Great Britain had an internet connection from January to February 2020, equating to around 46.6 million people every day, with 76 percent of adults using

internet banking, which may likely face the risk of intrusion on a daily basis (Milica, 2021).

By 2025, cybercrime is expected to damage the world's economy up to $10.5 trillion. This shows how important IDSs are to aid in defending against cyber threats or assaults (Ikusika, 2022). An intrusion detection system (IDS) is software or a physical electronic device that keeps an eye on system and network activity, detects threats, and notifies users when it finds them on a network (Farahnakian and Heikkonen, 2018).

This paper is organized into six sections: section I is an introduction to the research, section II provides the related work of IDS, section III is the methodology of the study, section IV is the results and discussion discovered from the research, section V is the conclusion of the study.

II. RELATED WORK

In this section, a literature review of past research on machine learning in IDS is analyzed.

The commercial invention of IDS technology started in the 1990s via its host-based Stalker IDS series solutions, and the first company to offer IDS tools for sale was Haystack Labs. IDS technology has since evolved to include network-based solutions that monitor and analyze traffic patterns for potential security threats. This technology has become a critical component in cybersecurity strategies for businesses of all sizes (Rajasekaran and Nirmala, 2015).

A data analytics technique called ML trains computers to learn from experience in the same way that both humans and other creatures do. Leverage on ML improves detection accuracy, handles complex data, and reduces false alarms, ultimately leading to more efficient and effective security measures (Ashesh, 2021). Supervised ML is within the purview of both AI and ML, it stands out due to how it trains computers to properly categorize data or forecast results by utilizing labeled datasets (IBM Cloud Education, 2020).

IDS researchers have utilized the majority of commercially available supervised learning methods. In essence, a supervised classifier's primary job is to identify malicious attacks traffic from normal ones in a set of network flows that have already been labeled (normal or attack) (Quang-Vinh, 2019). In some cases of ML, the classification model is only a collection of vectors of input, even without corresponding target requirements. Identifying clusters of related instances within the data is a procedure referred to as "clustering," or figuring out how the information is dispersed. Spatially, a method referred to as "density estimation could be the objective of such unsupervised learning issues (Sanatan, 2017).

Ashiku and Cihan (2021) opined that the broad use of computer systems interlinked with interoperability has become an essential need for enhancing our day-to-day operations. This interlinked system allows vulnerabilities that are far from the reach of humans to be exploited; for that reason, cyber-security methodologies are necessary to assure secured communication on a network.

Mazinia el at (2021) discuses about the use artificial bee colony (ABC) and AdaBoost algorithms to develop a hybrid solution for anomalous network-based IDS (A-NIDS) with a low false positive rate (FPR) and high detection rate (DR). The results show that this method significantly outperforms previous IDS on the same dataset, increasing accuracy and detection rate compared to traditional approaches. However, the researcher does not discuss the results using standardized metrics. The A-NIDS method is crucial for secure networks due to the vast amount of data on the network.

The challenges in network security due to the expansion of network size and data, leading to the emergence of new attacks are highlighted in (Ahmad, el at, 2020). They emphasize the importance of identifying hackers who may cause network attacks. IDS software is used to protect against network intrusions by analyzing traffic across networks to ensure availability, confidentiality, and integrity. The study uses the KDD Cup'99 and NSL-KDD datasets, which yielded 60% of the suggested methodologies. However, the effectiveness of these techniques is limited due to their age. The study suggests that AI-based NIDS should be evaluated using the latest dataset for optimal detection accuracy.

Akshay el at, (2022) propose "ImmuneNet," a deep learning hybrid system designed to detect and defend healthcare data against recent intrusion attacks. The system uses feature engineering, oversampling, and

hyper-parameter optimization strategies to achieve high accuracy and performance. The study shows that ImmuneNet outperforms other ML techniques using the CIC Bell DNS 2021 dataset, achieving ROC-AUC values of 99.2 percent and accuracy, precision, and recall of 99.19 percent. However, more examples from malware, spam, and phishing domains are needed for improved algorithm evaluation and natural class balance.

Hajisalem and Babaie (2018) highlight the vulnerability of computer networks to information theft due to increasing internet usage. They propose a hybrid categorization strategy combining ABC and AFS algorithms, using fuzzy C-means clustering and correlation-based feature selection techniques to separate the training dataset and remove extraneous features. If-Then rules are constructed using the CART technique to differentiate between normal and unhealthy situations. The hybrid approach performed better in performance metrics, with a 0.01 percent false positive rate and a 99 percent detection rate. However, the overhead was comparable to rival options in terms of time and computational complexity.

III. METHODOLOGY

A. Description of the dataset

The study used the UNSW-NB15 dataset from the Australian Centre for Cyber Security's Cyber Range Lab, which contains more than 2 million instances and 49 attributes, including two decisional attributes (normal 0 and attacks 1) in one class. 6 nominal values, 3 binary values, 10 float values, 28 integer values, and 2 timestamp values make up the attributes, with some of the properties in the dataset.

Table I shows the attack classification and the number of occurrences in the dataset. The dataset used for this research contains nine types of attacks: fuzzers, exploits, shell code, worms, DoS, backdoors, analysis, reconnaissance, generic, and normal (Moustafa and Slay, 2016).

Table 1 The Attack Classification and Quantity of Occurrences in The Dataset

S/N	ATTACK CLASSIFICATION	NUMBER OF OCCURRENCES
1	Fuzzers	24246
2	Reconnaissance	12228
3	Shell code	1288
4	Analysis	2677
5	Backdoors	2329
6	Dos	16353
7	Exploits	44525
8	Generic	215481
9	Normal	2218761
10	Reconnaissance	1759
11	Shell code	223
12	Worms	174
	Total	2540044

B. Experiment setup and Implementation

The study used various Python distribution platforms, including Anaconda Individual Edition, Jupyter Notebook, NumPy, Pandas, and Matplotlib, to conduct experiments. Python programming was used to achieve project objectives, as it is known for its simplicity, short coding times, and data processing capabilities (Worsley, 2024). The datasets used for testing and training were imported into the programming environment, cleaned to detect missing values, and filtered to include only intrusion-categorized data. The top ten features were selected using recursive feature elimination (RFE) to improve the model's performance and reduce overfitting, as shown in Table II and Fig.4. During the model-building phase, six base classifiers (XGBoost, LGBM, RF, NB, KNN, and DT) were trained on a fragment of the transformed dataset. Prediction evaluations were created on unseen data using 3-fold cross-validation to generate predictive models. The models were then compared based on metrics such as accuracy, precision, recall, and F1 score to determine the best-performing classifier. Finally, the selected model was fine-tuned using hyperparameter optimization techniques to further enhance its predictive capabilities.

Table 2 Top Features Importance

	Feature	Importance
0	sbytes	0.159533
1	smean	0.107565
2	ct_srv_dst	0.086754
3	dbytes	0.072566
4	service	0.062393
5	ct_srv_src	0.054479
6	sttl	0.054422
7	dmean	0.050324
8	sload	0.043304
9	dload	0.041363
10	dur	0.041285
11	rate	0.030530
12	ct_dst_sport_ltm	0.030504
13	sjit	0.029785
14	sinpkt	0.028145
15	djit	0.023840
16	dinpkt	0.023546
17	synack	0.022961
18	tcprtt	0.022449

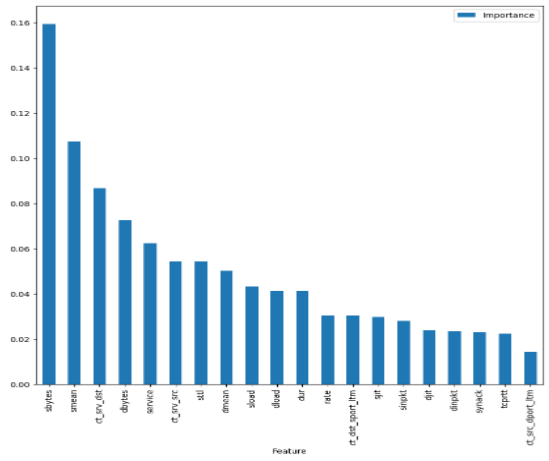

Fig.. 1: Feature selection using RFE.

Fig. 1 displays the qualities ranked in order of significance, with the most significant ones at the top. This ranking allowed to focus on the most influential traits when making predictions or conducting further analysis. Additionally, by using RFE, it ensures that the model is optimized for accuracy and efficiency. Using an RF classifier, the ten most important features were selected for the study out of 49 attributes.

C. Architecture of the IDS

Fig. 2 shows the architecture of the IDS. It has three stages: preprocessing, building the model, and model evaluation. Each stage is a part of the process and a comparison analysis to find the best model out of the six ML techniques that were used. The preprocessing module is in charge of normalization, feature selection, and data cleaning. It ensures that the incoming data is in a format that the model-building module can use. The model-building module trains and builds ML algorithms based on the preprocession of data using six distinct ML algorithms (Ajayi el at, 2022) Finally, to decide which method produces the best results for IDS, the model assessment evaluates the models' efficiency as measured by several standard metrics, including accuracy, precision, and recall. During this, six base classifiers (XGBoost, LGBM, RF, NB, KNN, and DT) were trained on a fragment of the transformed dataset. Each of the classifiers was evaluated on unseen data using 3-fold cross-validation, and the best ML classifier was determined through the performance evaluation.

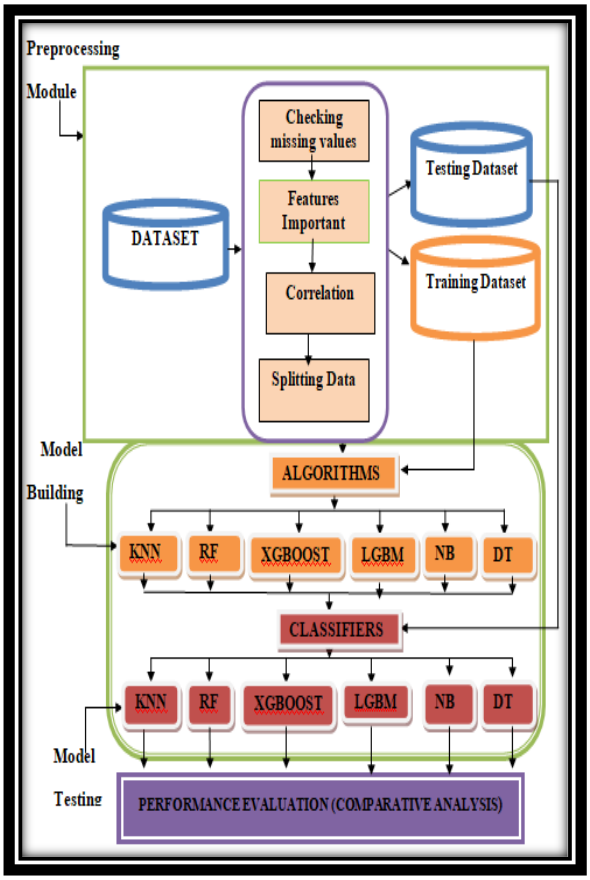

Fig. 2: Architecture of the IDS

IV. RESULT DISCUSSION

The six classifiers were used to determine the best three base classifiers out of the six classifiers through performance evaluation.

A. The classification reports trained classifiers.

Table III shows the classification report of the NB with an accuracy score of 0.5138 and a precision of normal 0.51, attack 0.89, recall normal 1.00, attack 0.03, and f1-score normal 0.67, attack 0.06, respectively.

NB Classification Report

Accuracy Score :
0.5137857142857143

Classification Report :

	precision	recall	f1-score	support
Normal	0.51	1.00	0.67	56000
Attack	0.89	0.03	0.06	56000
accuracy			0.51	112000
macro avg	0.70	0.51	0.37	112000
weighted avg	0.70	0.51	0.37	112000

Table IV shows the classification report of the KNN with an accuracy score of 0.8975 and a precision of attack 0.96, normal 0.85, recall attack 0.83 normal 0.97, and f1-score normal 0.90, attack 0.89, respectively.

KNN Classification Report

Accuracy Score :
0.8975357142857143

Classification Report :

	precision	recall	f1-score	support
Normal	0.85	0.97	0.90	56000
Attack	0.96	0.83	0.89	56000
accuracy			0.90	112000
macro avg	0.91	0.90	0.90	112000
weighted avg	0.91	0.90	0.90	112000

Table V shows the DT classification report with an accuracy score of 0.8957 and a precision of attack 0.96, normal 0.85 recall attack 0.83, normal 0.96 and f1-score normal 0.90, attack 0.89, respectively.

DT Classification Report

Accuracy Score :
0.8956875

Classification Report :

	precision	recall	f1-score	support
Normal	0.85	0.96	0.90	56000
Attack	0.96	0.83	0.89	56000
accuracy			0.90	112000
macro avg	0.90	0.90	0.90	112000
weighted avg	0.90	0.90	0.90	112000

Table VI shows the RF classification report with an accuracy score of 0.9151 and a precision of attack 0.97, normal 0.87, recall attack 0.85, normal 0.98 and f1-score normal 0.92, attack 0.91, respectively.

RF Classification Report

Accuracy Score :
0.9151160714285714

Classification Report :

	precision	recall	f1-score	support
Normal	0.87	0.98	0.92	56000
Attack	0.97	0.85	0.91	56000
accuracy			0.92	112000
macro avg	0.92	0.92	0.91	112000
weighted avg	0.92	0.92	0.91	112000

Table VII shows the XGBoost classification report with an accuracy score of 0.9157 and a precision of attack 0.98, normal 0.87, recall attack 0.86, normal 0.98 and f1-score normal 0.92, attack 0.91, respectively.

XGBOOST CLASSIFICATION REPORT

Accuracy Score :
0.9156964285714285

Classification Report :

	precision	recall	f1-score	support
Normal	0.87	0.98	0.92	56000
Attack	0.98	0.85	0.91	56000
accuracy			0.92	112000
macro avg	0.92	0.92	0.92	112000
weighted avg	0.92	0.92	0.92	112000

Table VIII shows the LGBM classification report with an accuracy score of 0.9137 and a precision of attack 0.998 normal 0.87, recall attack 0.85, normal 0.98 and f1-score normal 0.92, attack 0.91, respectively.

LGBM Classification Report

Accuracy Score :
0.9136875

Classification Report :

	precision	recall	f1-score	support
Normal	0.87	0.98	0.92	56000
Attack	0.98	0.85	0.91	56000
accuracy			0.91	112000
macro avg	0.92	0.91	0.91	112000
weighted avg	0.92	0.91	0.91	112000

B. The Receiver operating characteristics curves of the Trained Classifiers.

Each of the trained classifiers (NB, KNN, RF, XGBoost, LGBM, and DT) was tested cross validation and ROC curves obtained from each fold. The ROC-AUC performance of each classifier is unfolded and presented in this section.

1) The ROC-AUC curve obtained by Naïve Bayes

Fig.3 displays the performance of the NB AUC-ROC curve, which represents the plot of the TPR against the FPR. The AUC-ROC is 0.90.

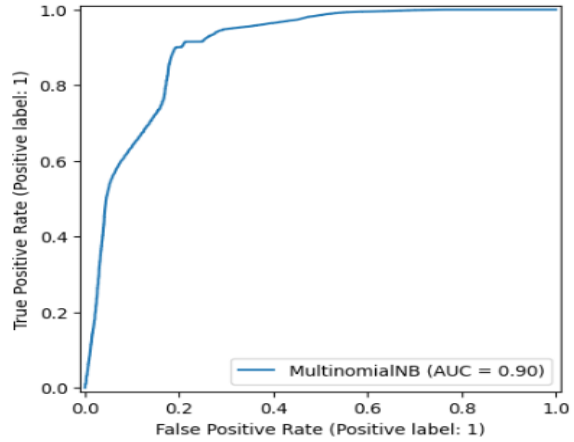

Fig.. 3: NB AUC- ROC Curve

2) The ROC-AUC curve obtained by KNN

Fig. 4 shows the performance of the KNN AUC-ROC curve, which represents the plot of the TPR against the FPR. The AUC-ROC is 0.96.

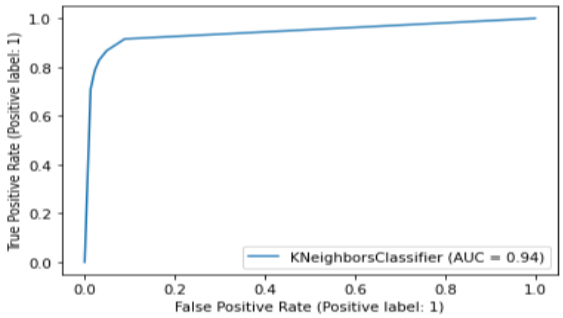

Fig.. 4: KNN AUC- ROC Curve.

3) The ROC-AUC curve obtained by DT.

Fig. 5 shows the performance of the DT AUC-ROC curve, which represents the plot of the TPR against the FPR. The AUC-ROC is 0.90.

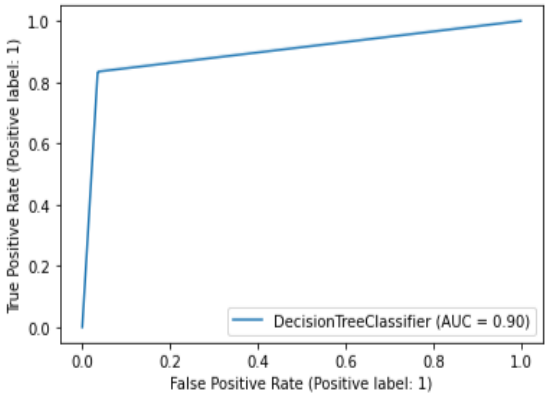

Fig.. 5: Decision Tree AUC- ROC Curve.

4) *The ROC-AUC curve obtained by RF.*

Fig. 6 shows the performance of the RF AUC-ROC curve, which represents the plot of the TP rate against the FP rate. The AUC-ROC is 0.98.

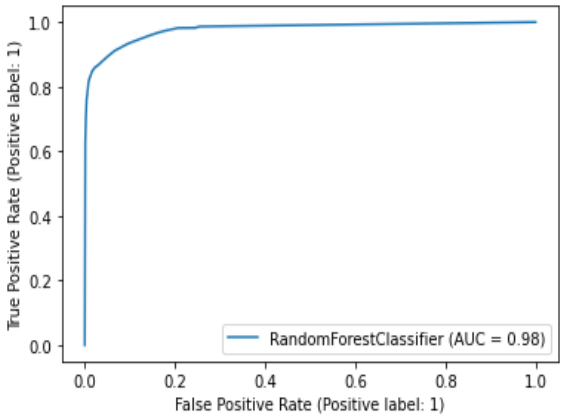

Fig.. 6: RF AUC- ROC Curve.

5) *The ROC-AUC curve obtained by XGBoost.*

Fig.7 shows the performance of the XGBoost AUC-ROC curve, which represents the plot of the TPR against the FPR. The AUC-ROC is 0.98.

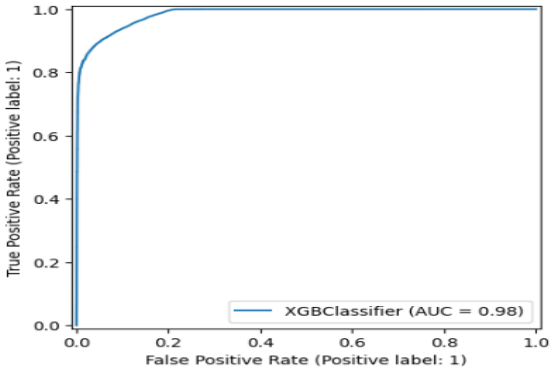

Fig.. 7: XGBoost AUC- ROC Curve.

6) *The ROC-AUC curve obtained by LGBM.*

Fig.8 shows the performance of the LGBM AUC-ROC curve, which represents the plot of the TPR against the FPR. The AUC-ROC is 0.98.

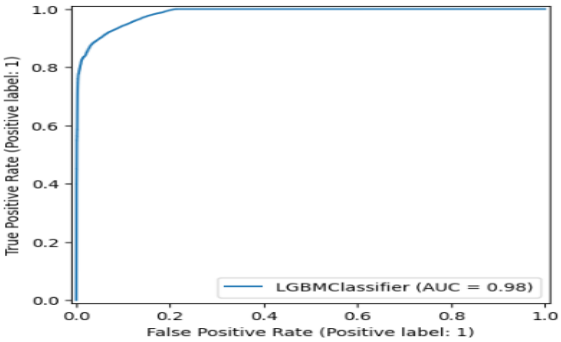

Fig.. 8: LGBM AUC- ROC Curve.

The Fig. 3, 4, 5, 6, 7, and 8 shows the AUC-ROC of each classifier (NB, KNN, RF, XGBoost, LGBM, and DT). It can be observed that the AUC-ROC of XGBoost, LGBM and RF (0.98) surpasses those of NB (0.90), KNN (0.94), and DT (0.90).

C. Performance results across the metrics

Table XI shows the statistical analysis of the results obtained in the experiment after evaluating the efficiency of the models as determined by the confusion matrix table and ROC-AUC metrics.

Performance results across the metrics

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE	ROC-AUC
NB	0.51	0.70	0.51	0.37	0.90
KNN	0.90	0.91	0.90	0.90	0.94
DT	0.90	0.90	0.90	0.90	0.90
RF	0.92	0.92	0.92	0.91	0.98
XGBoost	0.92	0.92	0.92	0.92	0.98
LGBM	0.91	0.92	0.91	0.91	0.98

Fig. 9 displays the results of the performance assessment of the basic models (ensemble and traditional classifiers) using standard metrics including accuracy, precision, recall, and F1 score. These metrics offer a thorough evaluation of the models' performance in classification tasks.

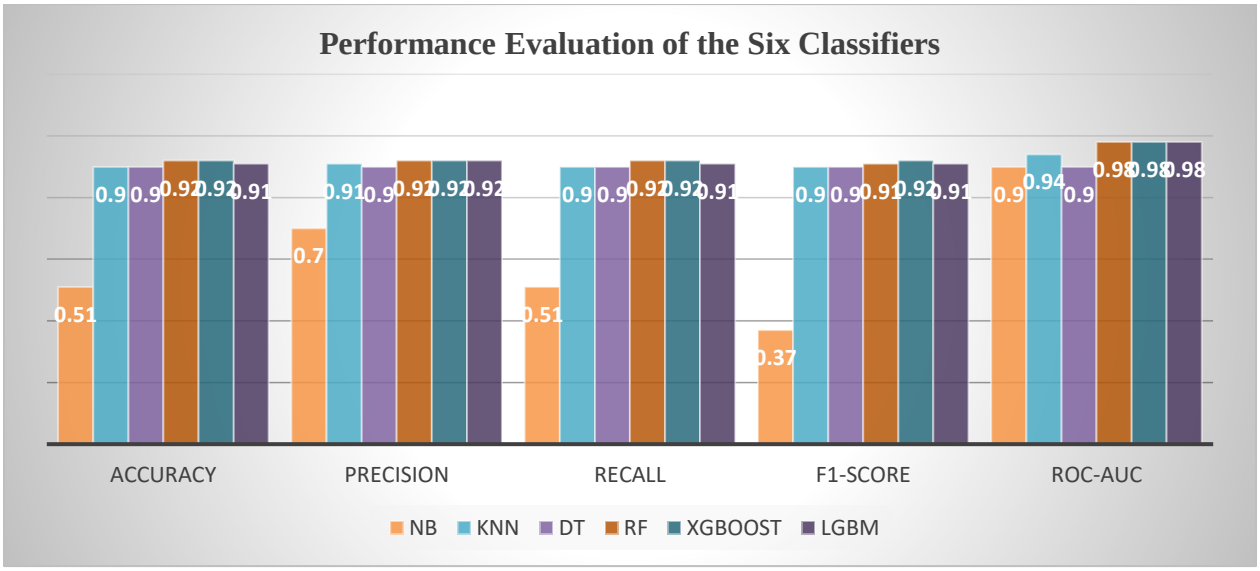

Fig. 9: Performance Evaluation of the base classifiers

The results obtained from each of the predictive classifiers in this research show the individual abilities of each machine learner in Table IX to detect intrusion. The results of the AUC-ROC curve are shown in Fig. 3, 4, 5, 6, 7, and 8 for each classifier: Naïve Bayes 0.90, K-Nearest Neighbour 0.94, Random Forest 0.98, XGBoost 0.98, LGBM 0.98, and Decision Tree 0.9. The results of the precision of each classifier show that Random Forest and XGBoost (0.98) have the highest performance compared to Naive Bayes (0.64), Decision Tree (0.85), K-Nearest Neighbour (0.85), and LGBM (0.87). The results of the recall of each classifier show that Random Forest, LGBM, and XGBoost (0.92) have the highest performance compared to Naive Bayes (0.51), Decision Tree (0.90), and K-Nearest Neighbour (0.90). The results of the F1-score of each classifier show that Random Forest, LGBM, and XGBoost (0.89) have the highest performance compared to Naive Bayes (0.28), Decision Tree (0.87), and K-Nearest Neighbour (0.86). The result of the accuracy of each classifier, XGBoost (0.8982), surpasses that of Naive Bayes (0.3399), KNN (0.8729), Random Forest (0.8966), LGBM (0.8950), and Decision Tree (0.8744).

V. CONCLUSION

In this paper a review of various literatures on IDSs was carried out. Six induced classifiers KNN, RF, LGBM, XGBoost, NB, and DT are presented, for the purpose to detecting intrusions. The UNSW-NB15 datasets and standard metrics were employed to determine the classifiers' effectiveness. The outcomes demonstrated that every classifier had advantages and disadvantages, with some classifiers performing better in some situations than others. This paper gives insightful information on how to apply these classifiers for IDS and makes suggestions for further study in this area. With this method, the data was correctly classified as either anomalous or healthy. This was established, and it was discovered that the XGBoost classifier outperformed others five classifiers in terms of accuracy (90%). Furthermore, by employing this strategy, the users and the organization's total security measures against any cyber-attacks may be greatly improved. Further investigation into the use of this strategy in various network contexts and an assessment of its efficacy against changing cyber threats might be the main goals of future studies, which would be helpful for improving precautions against cyber security threats.

REFERENCES

[1] Ajayi, J. O., Adetunmbi, A. O., Olowookere, T. A., &Sodiya, A. S. (2022). Phishing detection: Performance evaluation of both ensemble and classical machine learning models. *International Journal of Information Security, Privacy and Digital Forensics Computer Society (NCS), 6*(2), 1–7.

[2] Ahmad, Z., Adnan, S. K., &CheahWai, S. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Wiley Survey Paper*, 1–29. https://doi.org/10.1002/ett.4150

[3] Ashesh, A. (2021, August 30). Top 6 machine learning techniques. *Analytic Steps*. Retrieved September 12, 2022, from https://www.analyticssteps.com/blogs/top-6-machine-learning-techniques

[4] Ashiku, L., &Cihan, D. (2021). Network intrusion detection system using deep learning. *Elsevier Science Direct Procedia, 185*, 239–247. https://doi.org/10.1016/j.procs.2021.05.026

[5] Ben, G. (2016). *National cyber security strategy 2016–2021* (pp. 1–80). UK.

[6] Farahnakian, F., &Heikkonen, J. (2018). Anomaly-based intrusion detection using deep neural networks. *International Journal of Digital Content Technology and Its Applications, 12*(3), 70–81. https://doi.org/10.23919/ICACT.2018.8323687

[7] Hajisalem, V., &Babaie, S. (2018). A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. *Elsevier Computer Networks, 136*, 37–50. https://doi.org/10.1016/j.comnet.2018.02.028

[8] IBM Cloud Education. (2020, August 19). Supervised learning. *IBM Cloud Learn Hub*. Retrieved September 12, 2022, from https://www.ibm.com/cloud/learn/supervised-learning

[9] Ikusika, B. (2022, July 15). *SSRN*. Retrieved January 29, 2024, from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4165204

[10] Mazinia, M., Babak, S., &Iraj, M. (2019). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and Adaboost algorithms. *King Saud University-Computer and Information Sciences, 31*, 1–13. https://doi.org/10.1016/j.jksuci.2017.12.006

[11] Milica, S. (2021, June 21). 60+ amazing internet usage statistics UK edition. *Cybercrew*. Retrieved October 31, 2021, from https://cybercrew.uk/blog/internet-usage-statistics-uk/

[12] Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset. *Information Security Journal: A Global Perspective*, *25*(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

[13] Quang-Vinh, D. (2019). Studying machine learning techniques for intrusion detection systems. *HAL Open Science*, 02306521, 1–17. https://hal.archives-ouvertes.fr/hal-02306521

[14] Rajasekaran, K., & Nirmala, K. (2012). Classification and importance of intrusion. *International Journal of Computer Science and Information Security (IJCSIS), 10*(8), 1–5.

[15] Sanatan, M. (2017, May 19). Unsupervised learning and data clustering. *Towards Data Science*. Retrieved September 18, 2022, from https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a

[16] Venkat, R. (2021, April 1). Home: Information and communications technology—Intrusion detection system/intrusion prevention system market. *Industry Arc*. Retrieved October 26, 2021, from https://www.wicz.com/story/43590708/intrusion-detection-system-market-expected-to-grow-at-a-cagr-of-12-during-the-forecast-period-20192025

[17] Worsley, S. (2024, July 30). All about Python: The most versatile programming language. *DataCamp*. Retrieved October 20, 2024, from https://www.datacamp.com/blog/all-about-python-the-most-versatile-programming-language

[18] Zhiqiang, L., Bo, X., Bo, C., Xiaomei, H., & Mehdi, D. (2021). Intrusion detection systems in cloud computing: A comprehensive and deep literature review. *Wiley Online*, *34*(4), 1–15. https://doi.org/10.1002/cpe.6216