International Journal Multidisciplinary
Homepage : https://journal.antispublisher.com/index.php/IJMI
Email : admin@antispublisher.com

e-ISSN : 3031-9870
IJMI, Vol. 3, No. 3, August 2026
Page 15-27

Cyber Security Machine Learning Model Assessment: Perspectives from the UNSW-NB15 Dataset

Hazem Salim Abdullah*[1], Harith Abdulghani Ibrahim[2], Mohamed Waleed Jehad[3], Ahmed khaldoon abdulateef[4], Muhammed Saleh[5]

[1,2,3,4] Directorate of Municipalities, Nineveh Governorate, Mosul, IRAQ

[5] University Technology Malaysia (UTM), JB, Malaysia

DOI : https://doi.org10.61796/ijmi.v3i3.507

Sections Info

Article history:
Submitted: March 15, 2026
Final Revised: April 20, 2026
Accepted: May 25, 2026
Published: June 17, 2026

Keywords:
Intrusion Detection System (IDS)
Machine Learning (ML)
Network Intrusion Detection System (NIDS)

ABSTRACT

Objective: *Detection of cyberattacks still remains as one if the challenges.* ***Method:*** *The paper compares the performance of five machine learning classifiers namely Decision Tree (DT), XGBoost (XGB), Gaussian Naive Bayes (GNB), Random Forest (RF) and Logistic Regression (LR)) with respect to classifying network traffic as normal or malicious. It leverages the UNSW-NB15 dataset, which incorporates a wide variety of modern attack types and employs extensive data preprocessing, feature evaluation, and data discussion stratified splitting to ensure robust results.* ***Results:*** *The results show that the top performing model, XGBoost, has an accuracy of 93.62 % and AUC at 0.99 which means A great performance. Random forest and decision tree follow with accuracies of 93.60% and 92.93%, although less cumbersome fashions such as logistic regression and Gaussian Nav Bayes appear less accurate due to their limitations in dealing with complex communities in non-subnetworks traffic structure for real-time records tested.* ***Novelty:*** *This underlines the promise of advanced machine expertise, especially ensemble and tree-based perfection strategies, to improve automated cyber risk detection. Future efforts will focus on feature optimization and exploring streaming data.*

INTRODUCTION

The increase in internet-enabled devices along with the dependency on digital infrastructure has made the implementation of strong network security mechanisms very critical. One such network security mechanism includes IDS, which helps identify and prevent intrusion in the system and any type of attack on the system. IDS are responsible for monitoring traffic, analyzing traffic, and detecting any suspicious activity. However, most of the existing methods used in IDS include rule-based techniques and are mainly based on signatures. In order to overcome these limitations, the recent years have seen ML approaches receiving considerable attention. It includes communication data, including social networking activity, voice communication and email as well as user data, comprehensive logs of web applications & files for advanced analytics that show which users are active on what channels. Thus net forensic analytics of botnets can be carried out in realistic fashion if simulated data that capture configuration information of assets and business process data is enhanced while allowing effective risk assessments. The reliability of the Bot-IoT dataset is confirmed applying statistical and machine learning methods [1] highlighting the computing requirement of quality data sets for training and testing intrusion detection models. The rapid growth of virtual networks poses a threat to network-connected devices that has caused enormous concern among organizations that rely on this infrastructure [2] . Intrusion detection systems (IDS), which are designed to flag visitors in the community and stumble upon unauthorized access or malicious interests, serve as important protection. However, a full IDS based on traditional rules may also lack the

adaptability necessary to identify dynamic changes and emerging cyber threats, suggesting that more responsive and rational intrusion detection may want to combine system engineering techniques [3]. XGBoost, LightGBM, LOF, and the DRL model highlight the diversity of datasets in particular contexts [4]. The study compares the strengths and limitations of intrusion detection structures (IDS) classification algorithms [5]. Given the fact that network attack methods have become advanced and even bypassed Intrusion Detection Systems (IDS), this study compares the performance of several classifiers with the aim of selecting the best machine learning algorithm for IDS applications. The study notes how IDS systems face problems due to the use of IDS evasion techniques, which tamper with anomaly data and packets. Traditional IDS system weaknesses are examined alongside the development of machine learning and deep neural networks from their primitive state [6]. This overview proposes an (IDS) framework that reduces the issue of unmarried points of failure by leveraging collaborative efforts between users and cloud service providers to operate (IDS). Machine learning (ML) strategies play an important role in spotting every known and unknown threat [7]. This experiment evaluates 22 machines to learn algorithms focused on optimizing and validating performance through -phase techniques. The general linear pattern classifier, with the random oversampling approach, showed the highest detection performance [8] . The examine aspirations to compare the various gadgets gain knowledge of models in detecting and classifying in daily life and attack network activities the use of the U.S.-NB15 dataset. It evaluates 5 classifiers: decision tree, random forest, logistic regression, Gaussian naive baseline, and based on overall performance standards with gradient boosting, training time, accuracy, precision, not to forget, F1 estimation, AUC, and training evaluation. Structure will include: Section 2, literature review; Section 3, methodology and features selection; performance assessment; Section 4, results and discussions; and finally Section 5, with conclusions and recommendations for future research.

RELATED WORK

Research has been conducted about the future of IDS (Intrusion Detection Systems) as well as current concerns with the use of the Internet of Things (IoT) To build an enhanced means for interconnecting devices to allow for improved data sharing and decision-making. However, due to the rapid growth of IoT devices, cyber-crime continues to rise, while also furthering research into Network Intrusion Detection (NID). Recent advancements in machine learning provide promise as a means for resolving these problems; nevertheless, the vast differences among the IoT-IDS systems creates difficulties for using these various features, and thus, requiring more effective methods for selecting the features to improve NIDS based on anomalies [9].In [10], a novel approach to detecting intrusions on a computer network attained an accuracy of 99.65% when utilizing the results from KDD-Cup'99. This research used deep learning, specifically AE-LSTM, which combines Long Short-Term Memory and autoencoder networks, to detect irregularities in network usage. Preprocessing included the use of standard scalar preprocessing to address the imbalanced nature of the data, thereby increasing the ability to identify illegal activities on a computer network. The results from the NSL-KDD tested with this method provided accuracies of 98.69% and 98.70% for two different types of attacks and 98.78% for distinguishing between malicious and regular traffic, respectively [11].The former included a filter-based feature selection algorithm (CFS-DE) to select key characteristics of the data and weighted stacking technique to learn how to effectively distinguish between benign and suspicious behaviors to improve intrusion detection performance. Precision was also tested with other datasets (CSE-CIC-

IDS2018) [12]. The researchers in [13] introduced an IDS framework using IoT datasets to enhance IDS performance. The model uses DL algorithms and meta-heuristic optimization techniques for feature mining and selection purposes. Furthermore, they have used a convolutional neural network (CNN) for feature extraction. Reptile Search Algorithm-based feature selection is used to mine useful features from CNN output, after which achieves excellent performance for multiple classifier metrics when assessed on various datasets. According to [14], software-defined networking (SDN) can be used to create more flexible networks but this innovative method provides new vulnerabilities such as complicated networks and internet fraud. For SDN, these researchers suggested an HFS-LGBM IDS that employs LightGBM (gradient-boosting decision tree) and hybrid feature selection algorithms for intrusion detection and classification and outperforms existing methods. In paper [15], a new intrusion detection system that detects five network threats were proposed. The five classifications of threats were divided into Exploit, DOS, Probe, Generic, and Normal. This new model was created with the UNSW-NB15 dataset and an integrated model-based on classification to detect if there is malicious activity in the network. Researchers in [16] pointed out Network Intrusion Detection Systems (NIDS) role because of existing internet security threats in IoT. This article claims the UNSW-NB15 dataset is more suited for NIDS evaluation. The researchers achieved better accuracy with the SVM method than with the others, with a score of 85.99% on binary classification and 75.77% on multi-classification. Study [17] evaluates the CNN-BLSTM version using NSL-KDD and UNSW-NB15 datasets, with an emphasis on resampling techniques to cope with magnitude imbalance to improve community intruder detection The results show that NSL-KDD using selective synthetic (ADASY) (ADASY). The classification accuracy increases for the dataset while minimal resampling is sufficient for the USA-NB15 dataset The findings highlight the need to optimize resampling techniques based on accurate dignity distributions to achieve top-quality deep learning results in cybersecurity.

METHODOLOGY

Materials and Procedures The dataset utilized to create the system, the suggested classification machine learning techniques, and the suggested system are all covered in this part.

3.1 Machine Learning Models

This step summarizes the toolkit for gaining knowledge of the models used for binary classifications of the UNSW-NB15 dataset, including decision tree (DT), random forest (RF), logistic regression (LR), Gaussian-naive Bayes (GNB), and Gradient-boosting (XGB). It's an evaluation.

3.1.1 Decision trees

Decision Trees provide a flexible approach to using machine-learning algorithms for both classification and regression applications. Decision Trees graphically display decision-making processes (decisions made) based on the training data set (output of those decisions). The Decision Trees method is straightforward only requiring that data be of categorical (class) or numerical/continuous (measurements) type; the Decision Trees do not have a requirement on whether the data conforms to a specific type of distribution [18]. Decision Trees are capable of handling missing values, and they assist

in identifying which of the features are most important for their classification and will also capture the non-linear relationships between features. The ability of Decision Trees to depict/represent this information makes them very valuable in Intrusion Detection Systems (IDSs) for determining which features of the input data set (feature sets) are most critical for classification purposes.

3.1.2 Logistic Regression

Logistic regression is a supervised machine-mastering method for binary types that uses predictor variables and sigmoid functions to model the relationship between these predictors and binary end results The sigmoid feature transforms input values into probabilities from 0 to a 1, and wonderfully reflects color. This technique can be learned from the training records through optimization techniques such as gradient descent to limit the cost function, often cross-entropy loss [19].

3.1.3 Random Forest

Random Forest improves accuracy and prevents over-fitting using several decision trees for both classification and regression problems [20]. This method uses random selection of training data samples, splitting using random features, and averaging predictions to become resistant to biased data from training data sets. Moreover, this technique can deal with many input features and measure the importance of each feature in predicting.

3.1.4 XGBoost

XGBoost, created by Tianqi Chen and part of the Distributed Machine Learning Community, is an efficient tool for gradient-boosted decision trees that optimizes memory and hardware usage, improving performance and model tuning across different computing environments. Key features include gradient boosting, regularized boosting, stochastic boosting, reduced computation time, sparse-aware capabilities for missing values, parallel tree construction, and support for continued training on new data [21].

3.1.5 Gaussian naive Bayes

a probabilistic classifier that assumes feature independence and is based on Bayes' theorem. It is appropriate for real-time intrusion detection scenarios since it is effective and works well with tiny datasets. Because of its probabilistic nature, forecasts can be clearly explained.

3.2 Dataset Description and Characteristics

In this work, we discuss improving Intrusion Detection Systems (IDS) by analyzing the UNSW-NB15 dataset. This dataset was chosen due to its novelty and broad applications in developing general intrusion detection systems Compared to such as (KDD,NSL), as documented in the scientific literature. This differs from other similar datasets, such as AWID, which are primarily used for intrusion detection in wireless data traffic. Researchers at the Australian Centre for Cyber Security (ACCS) Laboratory at the University of New South Wales (UNSW) created the UNSW-NB15 dataset using the Perfect-Storm tool [31]. This dataset contains 100 GB of raw network traffic data, monitored by the TCP-Dump tool, and includes 2,540,044 real-world records. The dataset encompasses various types of network traffic, such as TCP, UDP, ICMP, and HTTP, as well as information about the traffic's source and destination and the time and duration of each packet [22, 23]. A total of 257,673 training and testing data were used, representing

10.14% of the total dataset. Of these, 93,000 were normal records and 164,673 were attack records as shown in Table 1, which details the Description category distribution. The ratio of attacks to normal records is shown in Figure 1 a,b.

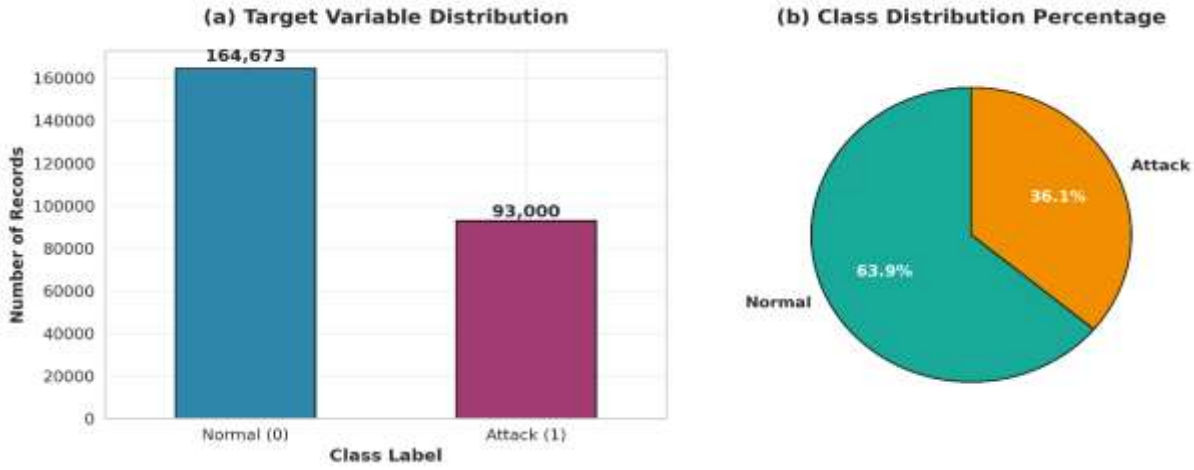

Figure 1. Distribution of the Attack or Normal record categories.

Table 1. Description category Distribution.

Type	Whole No. of Records	Training No. of Records	Testing No. of Records
Fuzzers	24,246	18,184	6,062
Analysis	2,677	2,000	677
Backdoors	2,329	1,746	583
DOS	16,353	12,264	4,089
Exploits	44,525	33,393	11,132
Generic	58,871	40,000	18,871
Reconnaissance	13,987	10,491	3,496
ShellCode	1,511	1,133	378
Worms	174	130	44
Attack Records	**164,673**	**119,341**	**45,332**
Normal Records	**93,000**	**56,000**	**37,000**
Total Records	**257,673**	**175,341**	**82,332**

3.2.1 Dataset Features

Table 2 outlines key features of the UNSW_NB15 dataset, which includes three categorical features: proto, service, and state. Additionally, the dataset includes a composite feature called "network byte" derived from the sbytes and dbytes attributes. Overall, the dataset has 49 attributes categorized into binary, nominal, and categorical types, encompassing both packet-based and flow-based features, and it identifies nine types of attacks alongside the normal class [23].

Table 2. Description of the attributes of UNSW-NB15

S.NO	Type of attributes	Name of attributes	Sequence No.
1	Flow	Script, Sport, Dstip, Dsport, Proto	1-5
2	Basic	State, Dur, Sbytes, Dbytes, Sttl, Dttl, Sloss, Dloss, Service, Sload, Dload, Spkts, Dpkts	6 - 18
3	Content	Swin, Dwin, Stepb, Dtcpb, Smeansz, Dmeansz, trans_depth, res_bdy_len	19-26
4	Time	Sjit, Djit, Stime, Ltime, Sintpkt, Dintpkt, Tcprtt, Synack, Ackdat	27-35
5	General Purpose	is_sm_ips_ports, ct_state_ttl, ct_flw_http_mthd, is_ftp_login, ct_ftp_cmd	36-40
6	Connection	ct_srv_src, ct_srv_dst, ct_dst_ltm, ct_src_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm	41-47
7	Labelled	attack_cat, Label	48-49

3.2.2 Preprocessing Data Framework

It is essential that datasets are properly prepared to help ensure reliable results from the UNSW-NB15 model and to mitigate potential biases in algorithms. Preprocessing involved removing unhelpful or non-predictive attributes to remove noise and avoid overfitting; converting all categorical variables to numerical variables without creating ordinal relationships; and standardizing all numeric attributes to provide equivalent weight to each attribute during training. Ensuring that the quality of the data eliminates the need to deal with an abundance of missing values greatly reduces the effort required to prepare the data; however, it is still important to be vigilant about possible problems related to the collection of data. In addition, the UNSW-NB15 dataset provides two different ways to classify (i.e., binary vs. multi-class) which provides researchers with a wide range of possibilities to test their research ideas, as well as the option to use either the preprocessed feature set or the raw packet capture data. Finally, dividing the dataset into a training set and a testing set also provides a mechanism to accurately validate the models developed to detect intrusions and fills some of the existing gaps in intrusion detection research.

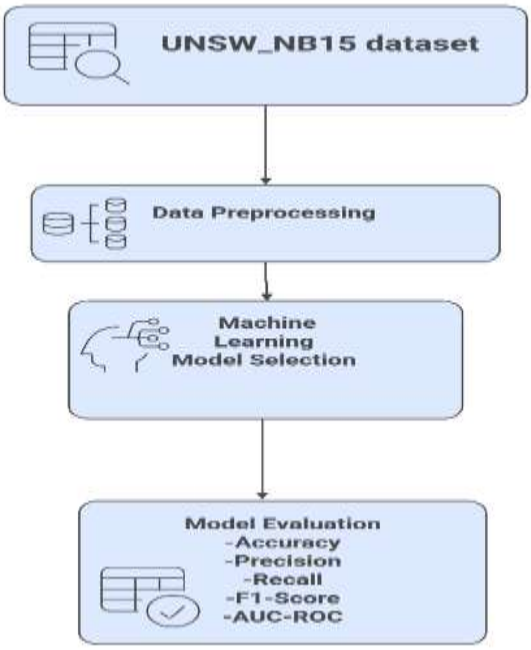

Figure 2. The proposed Experiment

The preprocessing methodology, illustrated in Figure 2 and Figure 3, begins with processing zero values before converting categorical data into numerical format via a label encoder. Subsequently, one-hot encoding is applied to eliminate value relationships from the label-encoded data. The processed dataset is then divided into 80% training and 20% testing. Models are constructed using classifiers such as XGB, GNB, RF, DT, and LR. Predictions are made using the test data's labels, followed by a comparison between actual and predicted results. Model performance is evaluated through metrics like accuracy, precision, recall, and F1 score. While challenges exist due to synthetic data and categorical imbalances, the dataset also presents opportunities for enhancing data balancing techniques and generalization methods.

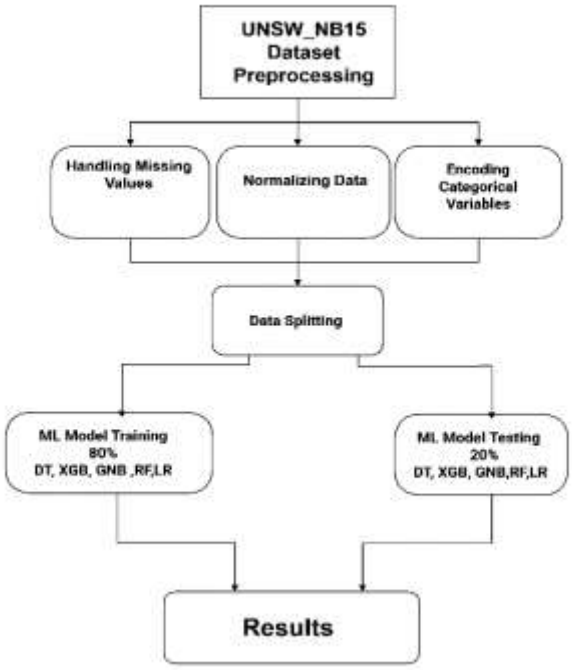

Figure 3. Flowchart of the proposed IDS.

3.3 Evaluation Metrics

- in figure 4 display Using a metrics such as accuracy, recall, precision, the F1-score, and the False Positive Rate from the confusion matrix, the study assesses the efficacy and efficiency of a classification model. Each performance metric is calculated as follows:
- Accuracy =((TP + TN))/((TP + FP + TN + FN)) [1]
- Recall = TP/(TP + FN) [2]
- Precision = TP / (TP + FP) [3]
- F1-Score = 2TP / (2TP + FP + FN) [4]
- FPR = FP / (FP + TN) [5]

		Predicted Condition	
Total Pop. = P+N		Pos. (PP)	Neg. (PN)
Actual Cond.	Pos. (P)	TP	FN
	Neg. (N)	FP	TN
Total Pop. = P+N		(P)	(N)

FIGURE 4: CONFUSION MATRIX

- where, TP is the number of true positive cases, TN is the number of true negative cases, FP is the number of false positive cases, and FN is the number of false negative cases.
- The Receiver Operating Characteristic (ROC) graph is used to determine the efficiency of machine learning algorithms in recognizing cyber security threats using the TPR and FPR at different classification cut-offs. The area under the curve (AUC) is a measure of the algorithm's efficiency, where a higher value approaching 1.0 denotes improved ability to distinguish malicious from harmless actions.

RESULT AND DISCUSSION

The UNSW-NB15 dataset was employed for evaluating various machine learning classifiers, including Decision Tree (DT), XGB, Gaussian Naive Bayes (GNB), Random Forest (RF), and Logistic Regression (LR). The experiments were conducted on an analytics machine equipped with an i5 CPU, Windows 11, and 16 GB of RAM, utilizing

Python for execution. The dataset was split into training and testing subsets in an 80:20 ratio post-preprocessing. Each classifier's performance was measured using multiple evaluation metrics, as documented in Table 3.

Table 3. Performance comparison of selected classifiers using UNSW-NB15

Model	Accuracy	Precision	Recall	F1-Score	Training Time (s)	ROC-AUC
XGB	93.62	97.15	92.73	94.89	4.89	0.9889
RF	93.60	97.21	92.66	94.88	107.23	0.9891
DT	92.93	97.19	91.59	94.31	3.99	0.9811
LR	88.53	91.98	89.90	90.93	296.38	0.9603
GNB	81.95	84.64	87.66	86.13	0.11	0.8910

The Figure 5, which describes the Accuracy of the selected machine learning model in a graphical representation of their precision. From the results, the XGB classifier outperforms the rest of the classifiers with an accuracy of 93.62%. See Figure 6 and figure 7 explain The area under curve was 0. 99, the training time was reasonable and good, and the classification effect was satisfactory. It has very good classification ability for different attack types and normal traffic Figure 8. Explain confusion matrix. In contrast to this, GNB had the worst accuracy out of the considered models. At the same time, Random Forest and Decision Tree gave also good results with 93.60% and 92.93% of accuracy respectively .The combination of this unique balance and technical adaptability along with advanced algorithmic optimizations creates XGBoost as the ideal practical solution for use in the detection of intrusions in a real-world setting when balancing among competing demands such as speed, accuracy, robustness, and cost of computation are required.

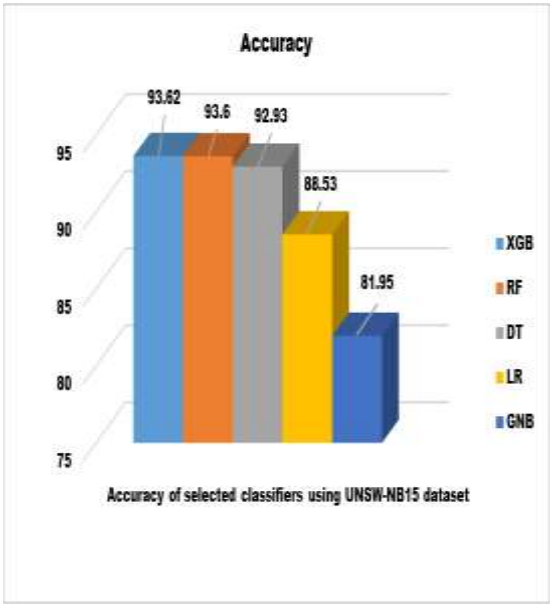

Figure 5. Accuracy of selected classifiers using UNSW-NB15 dataset

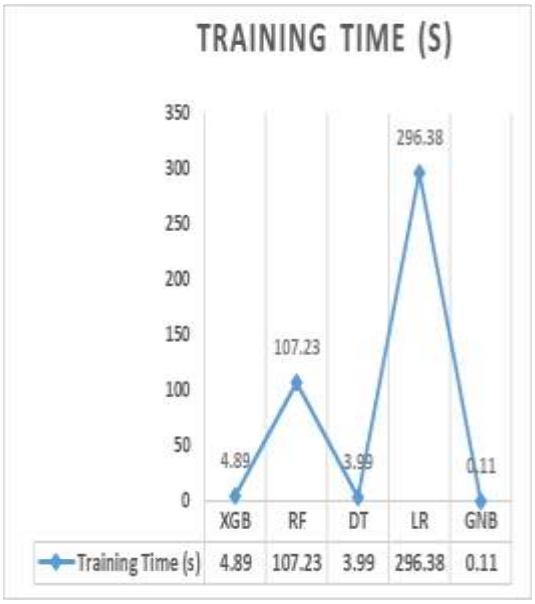

Figure 6. Training Time(s) of ML Model.

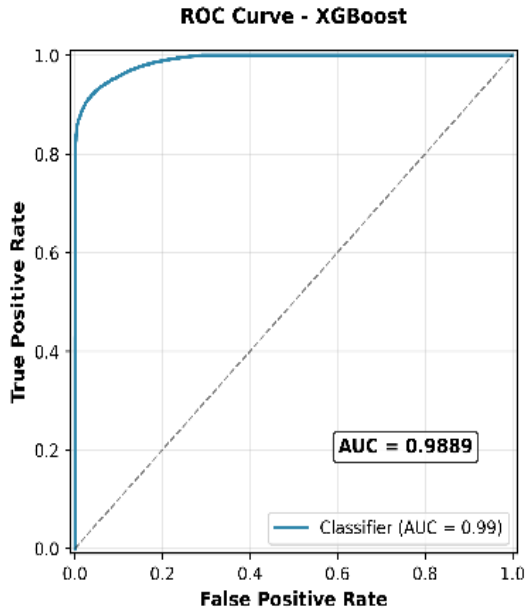

Figure 7. ROC Curve.

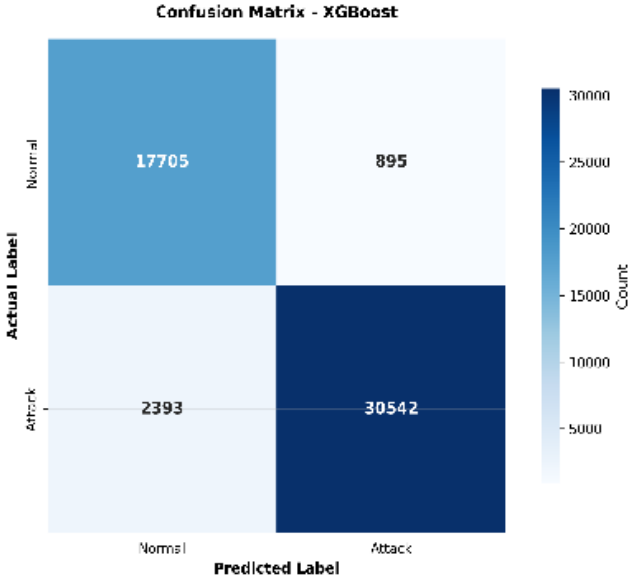

Figure 8. Confusion Matrix

Simpler models such as logistic regression and GNB fell short in accuracy due to their limited ability to capture nonlinear attack styles Overall, the overview underscores the effectiveness of machine learning methods for improving cybersecurity systems by allowing real-time computerized risk classification. For further investigation, Table 4 provides an evaluation of the classifications provided against the results obtained through different researchers and shows potential ways to improve botnet detection through the use of deep mastering.

Table 4. A Comparison between different ML models based on UNSW-NB15 **dataset**

Ref. No.	Model	Acc. (%)	Model Type
Proposed	**XGBoost**	**93.62**	**Machine Learning (ML) Models**
[24]	XGBoost	86.87	
[26]	XGBoost	87.5	
Proposed	**Random Forest**	**93.60**	
[26]	RF	87.2	
Proposed	**Decision Tree**	**92.93**	
[24]	DT	87	
[25]	DT	88.13	
[26]	DT	86.3	
Proposed	**Logistic Regression**	**88.53**	
[25]	LR	79.59	
[26]	LR	80.8	
Proposed	**Gaussian Naive Bayes**	**81.95**	
[24]	GNB	73	

CONCLUSION

Fundamental Finding: XGBoost posted an impressive accuracy of 93.62% along with an area under curve (AUC) of 0.99 making it the best performer and very viable for real-time protection. **Implication:** Ultimately, this study has demonstrated the promise of using ML for cyber-security purposes through machine learning systems. **Limitation:** Random Forest and Incremental Decision Trees also performed well but did so since they are complex algorithms; whereas, Logistic Regression and Bayesian classifiers did not perform as well due to being overly simplistic. **Future Research:** Additionally, more work needs to be done in the botnet detection area for the sake of deep learning approaches to be utilized more effectively as well as optimizing the model to give the best results possible.

REFERENCES

[1] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," IEEE Sensors Letters, vol. 3, no. 1, pp. 1–4, 2018.

[2] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.

[3] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges," Soft Computing, vol. 25, no. 15, pp. 9731–9763, 2021.

[4] J. Vitorino, R. Andrade, I. Praça, O. Sousa, and E. Maia, "A comparative analysis of machine learning techniques for IoT intrusion detection," in Springer, pp. 191–207, 2022, doi: 10.1007/978-3-031-08147-7_13.

[5] A. Heidari and M. A. J. Jamali, "Internet of Things intrusion detection systems: A comprehensive review and future directions," Cluster Computing, vol. 26, no. 6, pp. 3753–3780, 2023, doi: 10.1007/s10586-023-04126-3.

[6] O. V. Salmakayala, S. S. Ghidary, and C. Howard, "Review of IDS, ML and Deep Neural Network Technique in DDoS Attacks," International Journal of Computer Science, Engineering and Information Technology, vol. 12, no. 4, pp. 123–145, 2024.

[7] L. Diana, P. Dini, and D. Paolini, "Overview on Intrusion Detection Systems for Computers Networking Security," Computers, vol. 14, no. 3, p. 87, 2025, doi: 10.3390/computers14030087.

[8] R. Alasmari and A. A. Alhogail, "Protecting Smart-Home IoT Devices From MQTT Attacks: An Empirical Study of ML-Based IDS," IEEE Access, vol. 12, pp. 25993–26004, 2024.

[9] S. Walling and S. Lodh, "Network intrusion detection system for IoT security using machine learning and statistical based hybrid feature selection," Security and Privacy, vol. 7, no. 6, p. e429, 2024, doi: 10.1002/spy2.429.

[10] M. Imran, N. Haider, M. Shoaib, and I. Razzak, "An intelligent and efficient network intrusion detection system using deep learning," Computers and Electrical Engineering, vol. 99, p. 107764, 2022, doi: 10.1016/j.compeleceng.2022.107764.

[11] M. Mahmoud, M. Kasem, A. Abdallah, and H. S. Kang, "Ae-LSTM: Autoencoder with LSTM-based intrusion detection in IoT," in Proc. International Telecommunications Conference (ITC-Egypt), 2022, pp. 1–6, doi: 10.1109/ITC-Egypt55520.2022.9855688.

[12] R. Zhao, Y. Mu, L. Zou, and X. Wen, "A hybrid intrusion detection system based on feature selection and weighted stacking classifier," IEEE Access, vol. 10, pp. 71414–71426, 2022, doi: 10.1109/ACCESS.2022.3186975.

[13] A. Dahou et al., "Intrusion detection system for IoT based on deep learning and modified reptile search algorithm," Computational Intelligence and Neuroscience, vol. 2022, Art. no. 6473507, 2022, doi: 10.1155/2022/6473507.

[14] G. Logeswari, S. Bose, and T. J. Anitha, "An intrusion detection system for SDN using machine learning," Intelligent Automation & Soft Computing, vol. 35, no. 1, pp. 867–880, 2023, doi: 10.32604/iasc.2023.026769.

[15] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule-based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset," Cluster Computing, vol. 23, no. 2, pp. 1397–1418, 2020.

[16] D. Jing and H.-B. Chen, "SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset," in Proc. IEEE 13th International Conference on ASIC (ASICON), Chongqing, China, 2019, pp. 1–4.

[17] N. Dangol, A. Eaman, E. Shakshuki, and E. Hassan, "Impact of resampling techniques in deep learning based intrusion detection: A comparative study on NSL-KDD and UNSW-NB15," Procedia Computer Science, vol. 272, pp. 84–91, 2025, doi: 10.1016/j.procs.2025.10.182.

[18] S. J. Lee, P. D. Yoo, A. T. Asyhari, Y. Jhi, L. Chermak, Y. Y. Chan, and K. Taha, "IMPACT: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction," IEEE Access, vol. 8, pp. 65520–65529, 2020.

[19] N. Bhusal, M. Gautam, and M. Benidris, "Detection of Cyber Attacks on Voltage Regulation in Distribution Systems Using Machine Learning," IEEE Access, vol. 9, pp. 40402–40416, 2021.

[20] G. Apruzzese, M. Andreolini, M. Colajanni, and M. Marchetti, "Hardening random forest cyber detectors against adversarial attacks," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 4, pp. 427–439, 2020, doi: 10.1109/TETCI.2020.2988920.

[21] J. Brownlee, "A gentle introduction to XGBoost for applied machine learning," Machine Learning Mastery. [Online]. Available: http://machinelearningmastery.com/gentle-introduction-xgboost-appliedmachine-learning/. [Accessed: Mar. 2, 2018].

[22] N. Moustafa, "The UNSW-NB15 Dataset," Research Data Australia. [Online]. Available: https://researchdata.edu.au/the-unsw-nb15-dataset/1957529. [Accessed: Nov. 9, 2023].
[23] "The UNSW-NB15 Dataset," UNSW Research. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset. [Accessed: Nov. 9, 2023].
[24] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: Enhancing transparency and interpretability," Frontiers in Computer Science, vol. 7, Art. no. 1520741, 2025, doi: 10.3389/fcomp.2025.1520741.
[25] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," Journal of Big Data, vol. 7, 2020.
[26] K. K. Pal, A. V. Eriksen, and N. Dinh, "XGBoost Feature Selection for Multi-Class and Binary Classification on UNSW-NB15 Dataset," in Proc. IEEE International Conference on Consumer Electronics (ICCE), 2025, pp. 1–6.

* **Hazem Salim Abdullah (Corresponding Author)**
Directorate of Municipalities, Nineveh Governorate, Mosul, IRAQ
Email: hazemsas368@gmail.com

Harith Abdulghani Ibrahim
Directorate of Municipalities, Nineveh Governorate, Mosul, IRAQ
Email: haimsa77@gmail.com

Mohamed Waleed Jehad
Directorate of Municipalities, Nineveh Governorate, Mosul, IRAQ
Email: mhwaji08@gmail.com

Ahmed khaldoon abdulateef
Directorate of Municipalities, Nineveh Governorate, Mosul, IRAQ
Email: ahmedarab5@yahoo.com

Muhammed Saleh
University Technology Malaysia (UTM), JB, Malaysia
Email: asmuhammed2@graduate.utm.my