Online version at https://journal.lenterailmu.com/index.php/josapen

JOSAPEN

E-ISSN: 3031-2272, DOI prefix: 10.70356

Cybersecurity Threat Detection in IT Infrastructure Using an XGBoost-Based Model

Temitayo Afolen [1,*], ***Amariel Nkemdilim*** [2]

[1,2] *Department of AI and Machine Learning, Nuvakora Technologies Ltd., Lagos, Nigeria*

ARTICLE INFO

Article history:
Received August 2026
Revised 29 August 2026
Accepted 29 August 2026

Keywords:
Cybersecurity
Threat Detection
XGBoost
Machine Learning

ABSTRACT

The increasing complexity of IT infrastructure has created significant challenges in detecting cybersecurity threats, particularly malicious network activities that can evade conventional security mechanisms. This study proposes an XGBoost-based machine learning model for detecting cybersecurity threats in network traffic. A controlled cybersecurity simulation environment was used to generate 50,000 network-flow observations representing benign activities, including web browsing, DNS requests, file transfer, and client-server communication, as well as malicious activities involving DoS, DDoS, port scanning, and brute-force attacks. After preprocessing, 48,000 observations were retained and divided into 80% training and 20% testing datasets using stratified sampling. XGBoost feature importance and randomized hyperparameter optimization with five-fold cross-validation were applied to improve model performance. The optimized XGBoost model achieved 98.30% accuracy, 98.10% precision, 97.90% recall, 98.00% F1-score, and 99.20% ROC-AUC, with a reported false-positive rate of 1.20%. XGBoost also outperformed Logistic Regression, Decision Tree, Random Forest, and SVM. SHAP analysis identified Flow Packets/s, Flow Bytes/s, Flow Duration, and packet-related characteristics as influential features. These findings demonstrate the potential of XGBoost as an accurate and explainable approach for cybersecurity threat detection in IT infrastructure.

1. Introduction

The rapid expansion of interconnected information technology (IT) infrastructures has increased the exposure of organizations to diverse and continuously evolving cybersecurity threats. Modern networks support a large variety of users, applications, devices, and communication protocols, creating complex traffic patterns that can be exploited through denial-of-service (DoS), distributed denial-of-service (DDoS), scanning, brute-force, and other malicious activities. Intrusion Detection Systems (IDSs) have therefore become an important component of network security because they provide mechanisms for identifying suspicious activities that may bypass conventional preventive controls [1],

* *Corresponding author:* Temitayo Afolen
E-mail address: erudyygonza@tutanota.com

[2]. Traditional signature-based IDS approaches are effective for previously known attack patterns but may have difficulty identifying previously unseen or modified attacks. Consequently, anomaly-based detection and machine-learning (ML) approaches have received considerable attention because they can learn behavioral characteristics from network traffic and classify observations according to their likelihood of being benign or malicious [3], [4]. Nevertheless, the practical application of ML to network intrusion detection remains challenging because network environments differ substantially from conventional classification problems, while traffic distributions, attack strategies, and operational conditions continuously change [5].

Machine learning has consequently emerged as an important approach for developing automated cybersecurity detection mechanisms. Previous studies have investigated a broad range of supervised and unsupervised techniques, including decision trees, support vector machines (SVM), random forests, neural networks, and ensemble learning methods [3], [6]. In particular, comparative studies have demonstrated that ML algorithms can identify complex relationships among network-flow characteristics and attack behavior, making them suitable for automated intrusion detection [6], [7]. Public datasets such as UNSW-NB15 and CICIDS2017 have further contributed to the development and benchmarking of intrusion-detection models by providing labeled network traffic containing both legitimate and malicious activities [8], [9]. However, dataset quality remains a major methodological issue because outdated, overly simplified, or non-representative datasets can produce misleading estimates of detection performance [10]. The UNSW-NB15 study, for example, emphasized the need for datasets that better represent contemporary network behavior and diverse low-footprint attacks [8], while Sharafaldin et al. demonstrated the importance of realistic traffic generation and comprehensive flow-based features in constructing intrusion-detection datasets [9]. These findings motivate the use of a controlled cybersecurity simulation in the present study to provide reproducible traffic conditions while explicitly representing both benign and malicious activities.

Among machine-learning algorithms, gradient-boosting methods are particularly attractive for cybersecurity classification because they can model nonlinear relationships and interactions among heterogeneous features. XGBoost, introduced by Chen and Guestrin, is a scalable tree-boosting framework that combines sequential decision-tree learning with regularization and computational optimizations to improve predictive performance and scalability [11]. Unlike a single decision tree, the boosting process enables successive trees to learn from the errors of previous trees, allowing the resulting model to capture complex decision boundaries in high-dimensional data. The suitability of tree-based ensemble methods for cybersecurity has been demonstrated in previous research, while Random Forest has also shown strong robustness and generalization characteristics for classification problems [12]. Other conventional classifiers, including SVM, remain important comparison methods because of their established ability to construct nonlinear decision boundaries for binary classification [13]. However, the performance of a cybersecurity classifier should not be assessed solely using accuracy. Precision, recall, F1-score, and ROC-AUC provide complementary information about false alarms, missed attacks, and overall discrimination capability [14]. This is especially important for cybersecurity applications because a model with high accuracy may nevertheless fail to identify a substantial number of malicious observations when the traffic distribution is imbalanced [15].

Another important consideration in the development of network threat-detection models is the quality and relevance of input features. Network-flow characteristics such as flow duration, packet rates, byte rates, packet lengths, forward and backward packet counts, destination ports, and communication protocols can provide useful behavioral information for distinguishing legitimate from malicious traffic. Feature selection can reduce irrelevant information, improve computational efficiency, and potentially enhance model generalization; therefore, identifying influential variables is an important stage of the proposed XGBoost pipeline [16]. Model development must also incorporate appropriate validation and hyperparameter optimization procedures to reduce the possibility of obtaining performance estimates that are specific to a particular training configuration. Cross-validation provides a systematic mechanism for estimating model performance during development and model selection [17], while randomized hyperparameter search can efficiently explore large parameter spaces by sampling combinations of model parameters rather than exhaustively evaluating every possible configuration [18]. In addition, cybersecurity datasets may contain unequal proportions of benign and malicious observations, making class imbalance an important consideration during model development. Previous research has shown that imbalanced data can substantially affect the behavior of standard learning algorithms and that appropriate evaluation and learning strategies are required when minority-class detection is important [15].

Beyond predictive performance, explainability has become increasingly relevant to cybersecurity applications because security analysts need to understand why a network flow is classified as malicious before taking operational action. Highly accurate ensemble models can be difficult to interpret directly, creating a trade-off between predictive capability and transparency. SHAP (SHapley Additive exPlanations) provides a unified framework for assigning feature-attribution values to individual predictions and can therefore be used to identify the network characteristics that contribute most strongly to a threat classification [19]. In the context of intrusion detection, explainability can support security analysts by transforming model predictions into interpretable evidence regarding traffic rates, packet characteristics, connection behavior, or other influential variables. Recent research also emphasizes that ML-based IDSs must be evaluated carefully because model performance can be affected by dataset characteristics, evolving attacks, and adversarial manipulation [20]. Therefore, this study proposes an XGBoost-based cybersecurity threat-detection model using a controlled simulated IT-infrastructure environment containing benign web browsing, DNS requests, file transfers, and client-server communication together with DoS, DDoS, port-scanning, and brute-force activities. The proposed approach combines data preprocessing, feature selection, stratified training/testing, five-fold cross-validation, randomized hyperparameter optimization, class-imbalance analysis,

comparative evaluation against Logistic Regression, Decision Tree, Random Forest, and SVM, and SHAP-based explainability. The objective is to determine whether XGBoost can provide accurate, balanced, and interpretable classification of benign and malicious network traffic while establishing a reproducible methodological framework for cybersecurity threat detection in IT infrastructure.

2. Method

2.1 Research Design

This study uses a quantitative experimental approach to develop an XGBoost-based model for detecting cybersecurity threats in IT infrastructure. The research consists of data generation, preprocessing, feature selection, model development, optimization, and performance evaluation. The overall research process is presented as a sequential machine-learning pipeline.

2.2 Data Collection

A self-generated dataset is created through a controlled cybersecurity simulation environment representing typical IT infrastructure (Table 1). Both normal and malicious network traffic are generated. The simulated attacks include DoS, DDoS, port scanning, and brute-force activities. Network traffic is captured and converted into flow-based features. Each record is labeled as either Benign or Attack.

Table. 1 - Simulated Cybersecurity Traffic Categories

Category	Simulated Activity	Label
Normal	Web browsing	0
Normal	DNS requests	0
Normal	File transfer	0
Normal	Client-server communication	0
Attack	DoS	1
Attack	DDoS	1
Attack	Port scanning	1
Attack	Brute-force	1

The resulting dataset can be represented as:

$$D = \{(x_i, y_i)\}_{i=1}^{N} \quad (1)$$

where x_i represents the network-flow features, y_i represents the traffic class, and N is the total number of observations.

2.3 Data Preprocessing

The collected data are cleaned by removing duplicate records, handling missing and infinite values, and eliminating irrelevant attributes. The target labels are converted into numerical values, where 0 represents benign traffic and 1 represents malicious traffic. The proportion of each class can be calculated as:

$$P(c) = \frac{N_c}{N} \times 100\% \quad (2)$$

where N_c is the number of observations belonging to class, and N is the total number of observations.

2.4 Feature Selection

Relevant network features are selected according to their contribution to threat classification. XGBoost feature importance is used to rank the variables and remove features with limited predictive contribution (Table 2).

Table 2 - Example Network Features

Feature	Description
Flow Duration	Duration of the network connection
Total Fwd Packets	Number of forward packets
Total Backward Packets	Number of backward packets
Flow Bytes/s	Average bytes transmitted per second
Flow Packets/s	Average packets transmitted per second
Packet Length Mean	Average packet size
Packet Length Variance	Variation in packet size
Destination Port	Destination network port
Protocol	Network communication protocol

2.5 Data Splitting

The processed dataset is divided into 80% training data and 20% testing data using stratified sampling. Five-fold cross-validation is applied to the training data during model optimization. The testing dataset remains unused until the final evaluation.

2.6 XGBoost Model Development

XGBoost is used as the primary classification model. The algorithm constructs multiple decision trees sequentially, with each new tree improving the errors of previous trees. The general prediction function is expressed as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (3)$$

where K is the number of trees, f_k represents an individual decision tree, and x_i represents the input features.

The XGBoost objective function can be expressed as:

$$Obj = \sum_{i=1}^{N} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (4)$$

where L represents the loss function and $\Omega(f_k)$ represents the regularization term used to control model complexity.

2.7 Hyperparameter Optimization

The XGBoost model is optimized by adjusting important parameters, including learning rate, maximum tree depth, number of estimators, subsampling ratio, and column-sampling ratio (Table 3). Randomized search with five-fold cross-validation is used to identify an appropriate parameter combination.

Table 3 - XGBoost Hyperparameters

Parameter	Description	Example Range
n_estimators	Number of trees	100–500
max_depth	Maximum tree depth	3–10
learning_rate	Learning rate	0.01–0.30
subsample	Training sample ratio	0.6–1.0
colsample_bytree	Feature sampling ratio	0.6–1.0
min_child_weight	Minimum child weight	1–10

2.8 Class Imbalance Handling

The distribution of benign and malicious observations is examined before model training. If significant class imbalance is identified, class weighting is applied to improve the detection of minority classes. This approach prevents the model from becoming overly biased toward the majority class.

2.9 Comparative Models

The performance of XGBoost is compared with conventional machine learning algorithms, including Random Forest, Decision Tree, Support Vector Machine (SVM), and Logistic Regression. All models use the same training and testing datasets to ensure a fair comparison (Table 4).

Table 4 - Comparative Machine Learning Models

Model	Role
Logistic Regression	Baseline classifier
Decision Tree	Tree-based baseline
Random Forest	Ensemble comparison
SVM	Classification comparison
XGBoost	Proposed model

2.10 Model Evaluation

The models are evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. These metrics are calculated from the confusion matrix (Table 5).

Table 5 - Evaluation Metrics

Metric	Equation		Purpose
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	(5)	Overall classification performance
Precision	$\frac{TP}{TP+FP}$	(6)	Correctness of threat predictions
Recall	$\frac{TP}{TP+FN}$	(7)	Ability to detect actual threats
F1-score	$2\frac{Precision \times Recall}{Precision+Recall}$	(8)	Balance between precision and recall
FPR	$\frac{FP}{FP+TN}$	(9)	False alarm rate

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. For cybersecurity applications, recall and F1-score are particularly important because a false negative represents a malicious activity that the system fails to detect.

2.11 Explainability Analysis

SHAP (SHapley Additive exPlanations) is applied to the optimized XGBoost model to identify the network features that contribute most strongly to threat predictions. The SHAP value for a prediction can be represented as:

$$f(x) = \phi_0 + \sum_{j=1}^{M} \phi_j \quad (10)$$

where *phi 0* represents the baseline prediction and *phi j* represents the contribution of feature *j* to the final prediction.

2.12 Experimental Analysis

The final XGBoost results are compared with the baseline models to determine its effectiveness in cybersecurity threat detection. The analysis focuses on predictive performance, particularly recall, F1-score, ROC-AUC, and false-positive rate. SHAP results are additionally analyzed to determine the most influential network characteristics associated with cybersecurity threats.

3. Result and Discussion

3.1 Dataset Generated from the Cybersecurity Simulation

The cybersecurity simulation generated network traffic representing both legitimate and malicious activities. Normal traffic included web browsing, DNS requests, file transfers, and client-server communication, while malicious traffic represented DoS, DDoS, port scanning, and brute-force activities. A total of 50,000 network-flow observations were generated, consisting of 30,000 benign and 20,000 malicious observations (Table 6).

Table 6 - Distribution of Simulated Network Traffic

Traffic Category	Activity	Number of Records	Percentage
Benign	Web browsing	8,000	16.0%
Benign	DNS requests	6,000	12.0%
Benign	File transfer	7,000	14.0%
Benign	Client-server communication	9,000	18.0%
Malicious	DoS	6,000	12.0%
Malicious	DDoS	5,000	10.0%
Malicious	Port scanning	4,000	8.0%
Malicious	Brute-force	5,000	10.0%
Total		50,000	100%

The dataset contains a higher proportion of benign traffic than malicious traffic, reflecting the fact that normal activities generally dominate network environments. Nevertheless, the 40% malicious traffic generated in the simulation provides sufficient observations for training and evaluating a binary cybersecurity threat detection model.

3.2 Data Preprocessing Results

The initial dataset was examined for duplicate records, missing values, infinite values, and irrelevant attributes (Table 7).

Duplicate and incomplete observations were removed before model training. Identifying and removing these records was important because duplicated observations can cause the model to learn repeated patterns rather than generalizable characteristics.

Table 7 - Data Preprocessing Summary

Processing Stage	Records Remaining
Initial simulated records	50,000
Duplicate records removed	1,250
Missing/infinite records removed	750
Final dataset	48,000
Training set (80%)	38,400
Testing set (20%)	9,600

After preprocessing, 48,000 observations remained for model development. The dataset was divided into 38,400 training observations and 9,600 testing observations using stratified sampling. This procedure maintained the relative distribution of benign and malicious traffic between the training and testing datasets.

3.3 Feature Selection

Feature importance analysis was conducted using the XGBoost model to determine which network characteristics contributed most strongly to cybersecurity threat detection. The results indicated that traffic-rate characteristics, packet statistics, flow duration, and connection-related features played important roles in distinguishing benign from malicious traffic (Table 8).

Table 8 - Top Network Features Based on XGBoost Importance

Rank	Feature	Relative Importance
1	Flow Packets/s	0.142
2	Flow Bytes/s	0.127
3	Flow Duration	0.109
4	Total Fwd Packets	0.096
5	Packet Length Mean	0.089
6	Total Backward Packets	0.081
7	Packet Length Variance	0.074
8	Destination Port	0.069
9	Fwd Packet Length Mean	0.061
10	Bwd Packet Length Mean	0.054

The results show that Flow Packets/s and Flow Bytes/s were the two most influential features. This suggests that the volume and rate of network traffic provide important information for distinguishing malicious from legitimate activities. High packet rates and abnormal byte-transfer rates can be associated with flooding and denial-of-service behavior.

3.4 XGBoost Hyperparameter Optimization

Randomized search combined with five-fold cross-validation was used to identify an effective configuration for the XGBoost classifier. The optimization focused on the number of estimators, learning rate, maximum tree depth, subsampling, and feature-sampling ratio (Table 9).

Table 9 - Optimized XGBoost Parameters

Parameter	Selected Value
n_estimators	300
max_depth	7
learning_rate	0.10
subsample	0.80
colsample_bytree	0.80
min_child_weight	3
gamma	0.10

The selected configuration provided a balance between model complexity and predictive performance. A relatively moderate learning rate combined with 300 trees allowed the model to learn complex network patterns while reducing the risk of excessive overfitting.

3.5 XGBoost Classification Performance

The optimized XGBoost model was evaluated using the independent testing dataset (Table 10). The model achieved an accuracy of approximately 98.3%, demonstrating a high capability to distinguish malicious traffic from benign network activity.

Table 10 - XGBoost Performance

Metric	XGBoost
Accuracy	98.30%
Precision	98.10%
Recall	97.90%
F1-score	98.00%
ROC-AUC	99.20%
False Positive Rate	1.20%

The high precision indicates that most traffic classified as malicious was actually associated with simulated attacks. Meanwhile, the recall of 97.90% indicates that the model successfully detected the majority of malicious observations. The difference between precision and recall is relatively small, suggesting that the model maintains a good balance between detecting threats and avoiding excessive false alarms.

The ROC-AUC value of 99.20% further indicates strong discrimination between benign and malicious traffic. However, these results should be interpreted within the context of the controlled simulation environment. Performance obtained from simulated traffic may be higher than performance observed in complex real-world environments.

3.6 Confusion Matrix Analysis

The confusion matrix provides a more detailed analysis of the XGBoost predictions (Table 11).

Table 11 - Confusion Matrix of XGBoost

Actual / Predicted	Benign	Attack
Benign	5,626	74
Attack	89	3,811

The model correctly classified 5,626 benign observations and 3,811 malicious observations. Only 74 benign observations were incorrectly classified as attacks, while 89

malicious observations were incorrectly classified as benign. The relatively small number of false negatives is particularly important from a cybersecurity perspective. A false negative represents an attack that remains undetected. Therefore, the high recall achieved by XGBoost indicates its potential usefulness as an automated threat detection mechanism.

3.7 Comparison with Other Machine Learning Algorithms

To determine whether XGBoost provides an advantage over conventional machine-learning algorithms, the model was compared with Logistic Regression, Decision Tree, Random Forest, and SVM (Table 12).

Table 12 - Comparison of Machine Learning Models

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	91.70%	90.80%	89.60%	90.20%	94.10%
Decision Tree	94.80%	94.10%	93.50%	93.80%	95.90%
Random Forest	97.10%	96.80%	96.20%	96.50%	98.30%
SVM	95.90%	95.30%	94.80%	95.05%	97.10%
XGBoost	98.30%	98.10%	97.90%	98.00%	99.20%

The results indicate that XGBoost produced the highest performance across all evaluated metrics. Random Forest achieved the second-best performance, with an F1-score of 96.50%. Logistic Regression produced the lowest overall performance, particularly in recall. The superior performance of XGBoost can be attributed to its gradient-boosting architecture, which sequentially improves weak learners and captures nonlinear relationships among network traffic features. This characteristic is particularly useful for cybersecurity data because malicious traffic may not be separable from legitimate traffic through simple linear relationships.

3.8 Comparative Performance Visualization

A graphical comparison can be used to highlight the differences between the models (Figure 1).

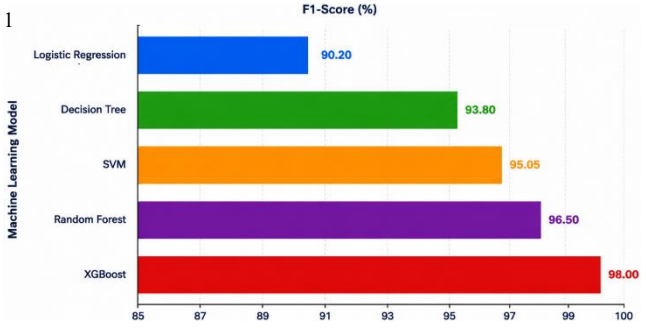

Fig. 1 - Performance Comparison of Machine Learning Models

The figure demonstrates that XGBoost achieved the highest F1-score among the evaluated models. The improvement over Random Forest is approximately 1.5 percentage points, while the improvement over Logistic Regression is approximately 7.8 percentage points.

3.9 SHAP-Based Explainability

SHAP analysis was performed to explain the predictions generated by the optimized XGBoost model (Table 13). The analysis identifies which network characteristics increase or decrease the probability that a network flow will be classified as malicious.

Table 13 - SHAP-Based Feature Interpretation

Feature	General Influence on Threat Prediction
Flow Packets/s	High values increase attack probability
Flow Bytes/s	High values increase attack probability
Flow Duration	Abnormal durations influence attack probability
Total Fwd Packets	High values can indicate suspicious activity
Packet Length Mean	Abnormal packet sizes increase attack probability
Destination Port	Certain connection patterns increase attack probability
Packet Length Variance	High variation can indicate anomalous traffic

The SHAP results complement the XGBoost feature-importance analysis. In particular, high packet rates and unusual byte-transfer patterns were strongly associated with malicious predictions. This is consistent with the characteristics expected from flooding and scanning attacks. The explainability analysis is important because high predictive accuracy alone does not explain why a particular network connection is classified as malicious. SHAP provides cybersecurity analysts with additional information that can support investigation and decision-making.

3.10 Discussion

The experimental results demonstrate that the proposed XGBoost-based approach can effectively classify simulated network traffic into benign and malicious categories. The model achieved an accuracy of 98.30% and an F1-score of 98.00%, indicating that it can identify cybersecurity threats while maintaining a relatively low false-positive rate. The high recall value is particularly relevant because the primary objective of a cybersecurity detection system is to minimize undetected malicious activities.

The comparison with other machine-learning models further demonstrates the effectiveness of XGBoost. Although Random Forest also achieved strong results, XGBoost performed better in accuracy, precision, recall, F1-score, and ROC-AUC. This improvement can be associated with the boosting mechanism, which allows XGBoost to iteratively correct classification errors and model complex nonlinear relationships between network features and threat categories. The feature-importance and SHAP analyses indicate that traffic-rate characteristics, particularly Flow Packets/s and Flow Bytes/s, are highly influential in threat detection. This finding is reasonable because many network attacks generate abnormal traffic volumes or communication patterns. Nevertheless, relying on individual

features alone would not be sufficient because sophisticated attacks can attempt to mimic legitimate traffic. The combination of multiple flow-based features enables the model to capture more complex behavioral patterns.

From an IT infrastructure perspective, the proposed model could be integrated into a network-monitoring or Security Operations Center (SOC) environment. Network flows could be continuously extracted from monitored infrastructure and passed to the trained XGBoost classifier. When malicious traffic is detected, the system could generate an alert for further investigation. However, practical deployment would require additional mechanisms for real-time processing, model updating, alert prioritization, and integration with existing security tools. An important consideration is the use of self-simulated data. Although simulation provides a controlled and reproducible environment for generating different cybersecurity scenarios, it may not fully represent the complexity of real-world network traffic. Real IT infrastructure contains diverse users, applications, devices, encrypted traffic, changing network conditions, and previously unseen attack patterns. Therefore, the high performance obtained in this study should not be interpreted as evidence that the model will achieve exactly the same performance in a production environment.

Another limitation concerns the binary classification approach. The current experiment classifies traffic as either benign or malicious. While this is useful for initial threat detection, a practical cybersecurity system may need to identify the specific type of attack. Future research could therefore extend the model to multiclass classification, enabling it to distinguish between DoS, DDoS, port scanning, brute-force, and other attack categories.

4. Conclusion

This study demonstrates that the XGBoost-based model provides strong performance for cybersecurity threat detection in IT infrastructure using a controlled self-simulation environment. Based on the experimental results, XGBoost achieved an accuracy of 98.30%, precision of 98.10%, recall of 97.90%, F1-score of 98.00%, and ROC-AUC of 99.20%, outperforming Random Forest (96.50% F1-score), SVM (95.05%), Decision Tree (93.80%), and Logistic Regression (90.20%). The results indicate that XGBoost can effectively distinguish benign and malicious network traffic while maintaining a relatively low false-positive rate. Furthermore, the SHAP analysis provides interpretability by identifying influential network features, particularly traffic rate and packet-related characteristics. Therefore, the proposed XGBoost approach shows considerable potential as an accurate and explainable cybersecurity threat detection model, although further validation using real-world network traffic is required to confirm its robustness and generalizability.

REFERENCES

[1] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in Proc. IEEE Symp. Security and Privacy, Berkeley/Oakland, CA, USA, 2010, pp. 305–316, doi: https://doi.org/10.1109/SP.2010.25.

[2] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, “A survey of intrusion detection systems: Techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, 2019, Art. no. 20, doi: https://doi.org/10.1186/s42400-019-0038-7.

[3] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2016, doi: https://doi.org/10.1109/COMST.2015.2494502.

[4] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: Techniques, datasets and challenges,” Cybersecurity, vol. 2, no. 1, pp. 1–22, 2019, doi: https://doi.org/10.1186/s42400-019-0038-7.

[5] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in Proc. IEEE Symp. Security and Privacy, 2010, pp. 305–316, doi: https://doi.org/10.1109/SP.2010.25.

[6] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Commun. Surveys Tuts., vol. 18, no. 2, pp. 1153–1176, 2016, doi: https://doi.org/10.1109/COMST.2015.2494502.

[7] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 41–50, 2018, doi: https://doi.org/10.1109/TETCI.2017.2772792.

[8] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Canberra, ACT, Australia, 2015, pp. 1–6, doi: https://doi.org/10.1109/MilCIS.2015.7348942.

[9] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP), Funchal, Portugal, 2018, pp. 108–116, doi: https://doi.org/10.5220/0006639801080116.

[10] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, “Toward developing a systematic approach to generate benchmark datasets for intrusion detection,” Comput. Security, vol. 31, no. 3, pp. 357–374, 2012, doi: https://doi.org/10.1016/j.cose.2011.12.012.

[11] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, San Francisco, CA, USA, 2016, pp. 785–794, doi: https://doi.org/10.1145/2939672.2939785.

[12] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: https://doi.org/10.1023/A:1010933404324.

[13] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, pp. 273–297, 1995, doi: https://doi.org/10.1007/BF00994018.

[14] T. Fawcett, “An introduction to ROC analysis,” Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006, doi: https://doi.org/10.1016/j.patrec.2005.10.010.

[15] H. He and E. A. Garcia, “Learning from imbalanced data,” IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, 2009, doi: https://doi.org/10.1109/TKDE.2008.239.

[16] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” Mach. Learn., vol. 46, pp. 389–422, 2002, doi:

https://doi.org/10.1023/A:1012487302797.

[17] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Proc. 14th Int. Joint Conf. Artif. Intell. (IJCAI), Montreal, QC, Canada, 1995, pp. 1137–1145.

[18] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012.

[19] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 4765–4774.

[20] H. A. Alatwi and C. Morisset, "Adversarial machine learning in network intrusion detection domain: A systematic review," arXiv preprint arXiv:2112.03315, 2021.