International Journal of
Advanced Science Computing and Engineering

Journal Homepage www.ijasce.org/index.php/ijasce

Feature Selection for Machine Learning-Based Multiclass Cyberattack Classification

Asmau Usman [a], Ibrahim Anka Salihu [b], Abdulrahman Aminu Ghali [c,*], Bilkisu Larai Muhammad Bello [b], Aminu Aminu Mu'azu [d], Amina Sambo Magaji [e]

[a] *Department of Computer Science, Faculty of Computing, Nile University of Nigeria, Nigeria*
[b] *Department of Software Engineering, Faculty of Computing, Nile University of Nigeria, Nigeria*
[c] *Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar, Malaysia*
[d] *Department of Computer Science, Faculty of Natural and Applied Sciences, Umaru Musa Yar'adua University Katsina, Nigeria*
[e] *Digital Literacy and Capacity Development Department, National Information Technology Development Agency, Nigeria*

Corresponding author: [*]aminu@utar.edu.my

***Abstract*—Cyberattack classification supports timely threat identification and network security monitoring. However, model effectiveness depends not only on the selected algorithm but also on whether the available features contain sufficient information to distinguish attack classes. This study evaluates the effect of feature selection on multiclass cyberattack classification and assesses the discriminative capability of a synthetic cybersecurity dataset. The dataset contains 40,000 records from three nearly balanced classes: distributed denial-of-service, malware, and intrusion. Preprocessing included removing high-cardinality and potentially post-detection attributes, handling missing security indicators, one-hot encoding categorical variables, and applying a stratified 80:20 training/testing split. We evaluated Random Forest, Logistic Regression, and XGBoost using all usable features and the ten highest-ranked features identified through Random Forest importance. We measured performance using accuracy, macro precision, macro recall, and macro F1-score, and included a stratified dummy classifier as a baseline. Logistic Regression using all features achieved the highest accuracy of 33.95%, while XGBoost using the Top-10 subset achieved the highest macro F1-score of 33.69%. The dummy baseline obtained 33.90% accuracy, showing that none of the models produced a practically meaningful improvement over chance-level classification. Feature reduction slightly improved Random Forest and XGBoost but reduced Logistic Regression performance. These findings indicate that the dataset features have limited discriminative capability for separating the three attack categories. Further research should validate the analysis using real-world traffic, alternative feature-selection methods, and cross-dataset experiments before interpreting model complexity and reported accuracy.**

***Keywords*— Cyberattack classification; feature selection; machine learning; cybersecurity; random forest; XGBoost.**

Manuscript received 15 Apr. 2026; revised 29 Jun. 2026; accepted 25 Jul. 2026. Date of publication 31 Aug. 2026.

I. INTRODUCTION

The increasing complexity of network attacks has created a need for intrusion detection systems that can efficiently identify malicious activities. Traditional approaches often depend on predefined rules or known attack signatures, which may limit their ability to detect previously unseen threats. Machine learning provides an alternative by learning patterns from network data and distinguishing between legitimate and malicious activities. Previous studies have demonstrated its application in network intrusion detection and cyberattack classification [1], [2].

Various machine learning and deep learning models have been applied to multiclass attack classification. Bacevicius and Paulauskaitė-Tarasevičienė compared several classical algorithms using raw, imbalanced intrusion detection data and found that the CART decision tree performed best [3]. Meanwhile, Kamal and Mashaly proposed Autoencoder-CNN and Transformer-DNN models for binary and multiclass intrusion detection [4]. These studies show that both conventional and advanced models can support cyberattack classification. However, performance depends on the dataset, preprocessing procedures, and input features.

As such, feature selection matters because cybersecurity datasets can include irrelevant, redundant, or weakly

informative attributes. Maseno and Wang proposed a hybrid wrapper feature selection method that combines the genetic algorithm and an optimized extreme learning machine for intrusion detection [5]. Kocyigit et al. [6] used a genetic algorithm to find relevant features for phishing URL detection. Both methods showed that selecting informative features can reduce model complexity while improving classification performance. Despite the strong performance in many studies, model reliability remains a concern. Cantone et al. showed that intrusion detection models performed very well in training and testing on the same dataset, but the results were not generalizable to other networks [7]. Verkerken et al. [8] also found that models degraded when the network environment changed. In such cases, high accuracy may reflect the data's specific characteristics. We therefore evaluate here the impact of feature selection on multiclass cyberattack classification using Logistic Regression, Random Forest, and XGBoost. We compare the models using all usable features and a smaller set of highly ranked features to see whether feature reduction can maintain classification performance while reducing model complexity.

II. Materials and Methods

This section describes the dataset, data preprocessing procedures, feature selection approach, machine learning models, and evaluation metrics used in this study. The experimental process includes preparing the cybersecurity data, identifying the most relevant features, training the selected classification models, and comparing performance using the complete and reduced feature sets.

A. Dataset Description

This study used the Cyber Security Attacks dataset published by Incribo on Kaggle [9]. The dataset is synthetic and contains 40,000 records represented by 25 attributes. It includes network traffic information, packet characteristics, device details, security indicators, attack signatures, and response-related information. The target attribute, Attack Type, consists of three classes: DDoS, malware, and intrusion.

The dataset contains both categorical and numerical attributes. Numerical attributes include source port, destination port, packet length, and anomaly score, while categorical attributes include packet type, protocol, network segment, traffic type, and log source. Several attributes also contain missing values or high-cardinality information. Therefore, preprocessing was required before feature selection and model training.

B. Data Preprocessing

Data preprocessing includes preprocessing operations such as encoding, normalization, data cleaning, and dimensionality reduction can affect intrusion detection performance, model complexity, and the reliability of the resulting analysis [10], [11]. Attributes with very high cardinality, including User Information, Payload Data, Source IP Address, Timestamp, Destination IP Address, Device Information, Geo-location Data, and Proxy Information, were excluded because most of their values were unique or unsuitable for direct model training.

Action Taken, Attack Signature, and Severity Level were also excluded because they may represent information available only after an attack has been identified. Their inclusion could allow the models to rely on post-detection information rather than patterns observable during attack classification.

We treated missing values in Alerts/Warnings, Malware Indicators, IDS/IPS Alerts, and Firewall Logs as the absence of the corresponding security indicator. Categorical attributes were transformed using one-hot encoding, while numerical attributes were retained in numerical form. We standardized numerical features for Logistic Regression.

We encoded the target variable into three classes: DDoS, malware, and intrusion. We divided the processed dataset into training and testing subsets using an 80:20 ratio with stratified sampling. We fitted all preprocessing operations, including encoding and standardization, only on the training data before applying them to the test data. This procedure prevented information from the test set from leaking into model development and producing overly optimistic performance estimates [12].

C. Feature Selection

Feature selection identified the most informative input variables and assessed whether a reduced feature set could maintain classification performance. Network intrusion datasets may contain redundant, incomplete, or irrelevant features that increase computational complexity without necessarily improving prediction performance [13].

We used Random Forest feature importance to rank the processed input features. Random Forest supports feature selection by assigning each feature a relevance score based on its contribution to reducing node impurity across the ensemble's decision trees [14]. We then arranged the resulting scores in descending order to identify the most influential features.

We then applied a Top-K strategy by selecting the ten highest-ranked features. Top-K feature selection ranks feature by importance and retains only the K most discriminative variables for model training [15]. We evaluated two feature configurations: all usable features remaining after preprocessing and the Top-10 feature subset. We ranked features using only the training data to prevent test-set information from influencing selection. After one-hot encoding, each categorical value became a separate binary feature; therefore, categories such as HTTP traffic, UDP protocol, and IDS/IPS alerts appeared as separate features in the ranking. We used the same training and testing partitions for both configurations.

D. Machine Learning Models

We evaluated three supervised machine learning models: Random Forest, Logistic Regression, and XGBoost. We selected these models to represent bagging-based, linear, and boosting-based classification approaches. Similar studies have compared these models under a common intrusion-detection framework for binary and multiclass threat classification [16].

We used Logistic Regression as a linear baseline because it provides a simple, interpretable probabilistic model. Random Forest combines multiple decision trees and

determines the final class through aggregated voting. XGBoost builds decision trees sequentially, with each new learner attempting to reduce the errors produced by the preceding learners [17]. Tree-based ensemble models such as Random Forest and XGBoost have also shown strong performance in multiclass network attack classification [18]. All models were trained using the same training partition and evaluated using the same testing partition. We applied the same experimental procedure to both the full-feature and Top-10 feature configurations to ensure a consistent comparison.

E. Evaluation Metrics

We evaluated model performance using accuracy, macro precision, macro recall, and macro F1-score. These metrics are common in intrusion detection applications for assessing overall classification performance and class-specific detection performance [19]–[21]. Accuracy measures the proportion of correctly classified records across all test instances. Precision is the percentage of the predicted records in a class that are correctly classified, and recall is the fraction of the actual records in the class that are detected. The F1-score is the harmonic mean of precision and recall.

We used macro averaging for precision, recall, and F1-score. We calculated each metric separately for each attack class and then averaged them. Therefore, in this way, each class is not a major component of the evaluation, and multiclass performance is more balanced. Previous intrusion detection studies [19] emphasized that large data sets can mask poor detection performance for individual attack classes, so we wanted a more accurate comparison using macro-averaged and class-based metrics.

We also included a stratified dummy classifier for our baseline. The classifier generates predictions according to the class distribution, with no knowledge of the relationships between the input features and the target labels. We used the baseline to check whether the machine learning models achieved any significant improvement over chance-level classification. The confusion matrix for the selected-feature XGBoost model was also examined to see whether DDoS, malware, and intrusion classes overlap in predictions.

III. Results and Discussion

This section provides a comprehensive overview of the dataset characteristics, including the various attributes and their significance in the analysis. It assesses missing values and explores how gaps in the data could affect the overall analysis. It also evaluates classification performance, highlighting how effectively the model differentiates between cyber threats—specifically, Distributed Denial of Service (DDoS) attacks, malware, and intrusion attempts. The feature-ranking results are also discussed, showcasing which cybersecurity features are most influential in distinguishing these attack types. Finally, we compare results with a random baseline to assess the reliability and effectiveness of the available features in identifying and categorizing these security threats.

A. Dataset Characteristics

Table I shows the distribution of the three attack classes in the dataset, with a nearly even allocation among DDoS, malware, and intrusion records. Specifically, DDoS attacks accounted for 33.57% of the total dataset, closely followed by malware at 33.27%, and finally, intrusion incidents at 33.16%. This equitable distribution suggests that class imbalance is unlikely to significantly affect the classification outcomes, enhancing the reliability of the results.

TABLE I
DISTRIBUTION OF ATTACK CLASSES

Attack Type	Number of Records	Percentage
Intrusion	13,265	33.16%
Malware	13,307	33.27%
DDoS	13,428	33.57%
Overall	40,000	100.00%

An initial data-quality analysis identified missing values in five attributes, as shown in Table II. Each attribute contained approximately 50% missing entries. Alerts/Warnings recorded the highest missing percentage at 50.17%, while Proxy Information recorded the lowest at 49.63%. We treated missing entries in Malware Indicators, Alerts/Warnings, Firewall Logs, and IDS/IPS Alerts as the absence of the corresponding security indicator. Proxy Information was excluded because it had substantial missingness and high-cardinality values unsuitable for direct model training.

TABLE II
ATTRIBUTES WITH MISSING VALUES

Attribute	Missing Records	Percentage
Proxy Information	19,851	49.63%
Firewall Logs	19,961	49.90%
Malware Indicators	20,000	50.00%
IDS/IPS Alerts	20,050	50.13%
Alerts/Warnings	20,067	50.17%

B. Classification Performance

Table III presents the performance of Logistic Regression, Random Forest, and XGBoost using two feature configurations. The first configuration used all usable features after preprocessing, while the second configuration used the ten highest-ranked features.

TABLE III
CLASSIFICATION PERFORMANCE

Feature Set	Model	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
All features	Logistic Regression	0.3395	0.3386	0.3392	0.3366
All features	Random Forest	0.3246	0.3247	0.3246	0.3245
All features	XGBoost	0.3363	0.3363	0.3363	0.3362
Top 10 features	Logistic Regression	0.3305	0.3293	0.33	0.3164
Top 10 features	Random Forest	0.3294	0.3294	0.3294	0.3293
Top 10 features	XGBoost	0.3371	0.337	0.337	0.3369

Logistic Regression using all usable features achieved the highest overall accuracy (33.95%). However, its macro F1-score was only 33.66%, indicating that the model did not consistently distinguish the three attack categories. XGBoost achieved 33.63% accuracy using all features. Its accuracy increased slightly to 33.71% when it used only the ten highest-ranked features. The corresponding macro F1-score also increased from 33.62% to 33.69%. This shows that

feature reduction produced a small improvement for XGBoost while reducing the number of input variables.

Random Forest achieved the lowest performance across all features, with an accuracy of 32.46%. When the feature set was reduced, it performed better, reaching 32.94% with fewer features. Although feature selection improved Random Forest and XGBoost performance, the gains were small and did not yield reliable attack classification. Logistic regression performed differently. Its accuracy decreased from 33.95% to 33.05% after feature reduction. This indicates that some lower-ranked features may still have provided little to no information to the linear classifier, even though their overall importance was relatively low.

C. Feature Importance

Table IV shows the 10 highest-ranked features by importance in the Random Forest feature ranking. Four of the numerical variables received considerably higher importance scores than the rest of the categorical variables.

TABLE IV
TOP TEN RANKED FEATURES

Rank	Feature	Importance
1	Source Port	0.1772
2	Destination Port	0.1762
3	Anomaly Scores	0.1758
4	Packet Length	0.1752
5	Traffic Type: HTTP	0.0157
6	Protocol: UDP	0.0154
7	Traffic Type: FTP	0.0153
8	Protocol: ICMP	0.0152
9	Protocol: TCP	0.0152
10	IDS/IPS Alert	0.0149

Source Port, Destination Port, Anomaly Scores, and Packet Length were the four most influential features. Each had an importance score of about 0.18, whereas the categorical protocol and traffic-type indicators had much lower importance. The large difference between the four numerical attributes and the other features indicates that our models focused mainly on port information, packet size, and anomaly scores. But the high importance of these features did not translate into good classification performance. An important feature can rank higher than other available features even if its absolute discriminative power is quite weak.

D. Comparison with the Random Baseline

We used a stratified dummy classifier to test whether machine learning models learned meaningful relationships between the available features and the attack labels. The dummy classifier predicted based on the class distribution and achieved about 33.90% accuracy. The highest machine learning accuracy was 33.95%, achieved by Logistic Regression with all features. This is only 0.05 percentage points better than the dummy baseline. XGBoost using the selected features achieved 33.71% accuracy, slightly below the dummy classifier.

The similarity between the machine learning results and the random baseline indicates that the input attributes contained very little discriminative information for differentiating DDoS, malware, and intrusion records. The models therefore performed about as well as random predictions in a balanced three-class problem.

The confusion matrix of the selected-feature XGBoost model shows substantial overlap among the three classes. No attack category was consistently distinguished from the others, and predictions were fairly evenly distributed across DDoS, malware, and intrusion labels.

E. Discussion

The results demonstrate that feature selection alone could not overcome the weak relationship between the available attributes and the attack labels. Reducing the feature set slightly improved Random Forest and XGBoost, but the results remained close to the random baseline. Therefore, we observed no significant improvement. The results also show that model complexity does not necessarily result in better classification. XGBoost and Random Forest were not substantially better than Logistic Regression, even though they do capture nonlinear relationships. The low performance of all three models indicates that the main challenge in feature classification is feature-label separability, not algorithm selection.

The dataset contains relevant cybersecurity information, such as ports, packet length, protocol, traffic type, security alerts, and anomaly scores. However, the distribution of this information may not be consistent enough across the three attack types. As a result, the models could not identify consistent patterns across DDoS, malware, and intrusion data.

These insights underscore the importance of evaluating feature discrimination before treating high machine-learning performance as evidence of an effective intrusion detection system. A dataset might have balanced labels and multiple security-related features but still lack good relationships between its features and its target classes. Future evaluation should account for simple baselines, feature-label analysis, and class separability before developing complex models.

IV. CONCLUSION

In this study, we conducted a comprehensive evaluation of how feature selection affects multiclass cyberattack classification, using three prominent machine learning models: Logistic Regression, Random Forest, and XGBoost. Our analysis began by comparing these models' performance using two feature sets: the full set and a reduced set of the top ten features identified through the Random Forest feature importance ranking.

The results revealed that Logistic Regression achieved the highest accuracy of 33.95% when utilizing all available features. Conversely, XGBoost achieved the highest macro F1-score of 33.69% when using the top ten features. Interestingly, a baseline stratified dummy classifier recorded an accuracy rate of 33.90%, suggesting that none of the machine learning models demonstrated a significant improvement over random-chance classification.

Further investigation into the effects of feature reduction showed a slight performance improvement for both Random Forest and XGBoost, while Logistic Regression declined. This suggests that decreasing the number of inputs did not effectively address the inherent weaknesses in the relationships between the selected features and the corresponding target labels.

Additionally, the confusion matrix analysis highlighted notable overlap among records of DDoS attacks, malware

incidents, and intrusion attempts. These insights point to a concerning limitation in the discriminative power of the available features, making it difficult to clearly distinguish the three categories of cyberattacks.

Looking ahead, we recommend future research to validate these findings by employing real-world network traffic datasets. We also suggest exploring additional feature engineering and selection methods, as well as conducting cross-dataset experiments to uncover more reliable and generalizable patterns of cyberattack behavior.

REFERENCES

[1] A. S. Dina and D. Manivannan, "Intrusion detection based on Machine Learning techniques in computer networks," *Internet Things*, vol. 16, p. 100462, Dec. 2021, doi:10.1016/j.iot.2021.100462.

[2] S. V. Golande, S. Vaidya, A. Pardeshi, V. Katkade, and V. Pawar, "An efficient network intrusion detection and classification system using machine learning," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 267–272, Nov. 2024, doi:10.48175/ijarsct-22045.

[3] M. Bacevicius and A. Paulauskaite-Taraseviciene, "Machine learning algorithms for raw and unbalanced intrusion detection data in a multi-class classification problem," *Appl. Sci.*, vol. 13, no. 12, p. 7328, Jun. 2023, doi:10.3390/app13127328.

[4] H. Kamal and M. Mashaly, "Enhanced hybrid deep learning models-based anomaly detection method for two-stage binary and multi-class classification of attacks in intrusion detection systems," *Algorithms*, vol. 18, no. 2, p. 69, Jan. 2025, doi:10.3390/a18020069.

[5] M. Maseno and Z. Wang, "Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection," *J. Big Data*, vol. 11, no. 1, Feb. 2024, doi:10.1186/s40537-024-00887-9.

[6] E. Kocyigit, M. Korkmaz, O. K. Sahingoz, and B. Diri, "Enhanced feature selection using genetic algorithm for machine-learning-based phishing URL detection," *Appl. Sci.*, vol. 14, no. 14, p. 6081, Jul. 2024, doi:10.3390/app14146081.

[7] M. Cantone, C. Marrocco, and A. Bria, "Machine learning in network intrusion detection: A cross-dataset generalization study," *IEEE Access*, vol. 12, pp. 144489–144508, 2024, doi:10.1109/ACCESS.2024.3472907.

[8] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *J. Netw. Syst. Manage.*, vol. 30, no. 1, Oct. 2021, doi:10.1007/s10922-021-09615-7.

[9] Incribo, "Cyber security attacks," Kaggle, 2024. [Online]. Available: https://www.kaggle.com/datasets/teamincribo/cyber-security-attacks.

[10] H. Güney, "Preprocessing impact analysis for machine learning-based network intrusion detection," *Sakarya Univ. J. Comput. Inf. Sci.*, vol. 6, no. 1, pp. 67–79, Apr. 2023, doi:10.35377/saucis...1223054.

[11] E. Jaw and X. Wang, "Feature selection and ensemble-based intrusion detection system: An efficient and comprehensive approach," *Symmetry*, vol. 13, no. 10, p. 1764, Sep. 2021, doi:10.3390/sym13101764.

[12] M. A. Bouke and A. Abdullah, "An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability," *Expert Syst. Appl.*, vol. 230, p. 120715, Nov. 2023, doi:10.1016/j.eswa.2023.120715.

[13] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: Enhancing intrusion detection with hybrid feature selection and stack ensemble learning," *J. Big Data*, vol. 11, no. 1, Sep. 2024, doi:10.1186/s40537-024-00994-7.

[14] M. A. Abdel-Rahman et al., "Feature importance guided autoencoder for dimensionality reduction in intrusion detection systems," *Sci. Rep.*, vol. 16, no. 1, Feb. 2026, doi:10.1038/s41598-026-36695-9.

[15] B. M. Kouassi, A. B. Ballo, K. J. Ayikpa, D. Mamadou, and M. Z. J. Coulibaly, "Top-K feature selection for IoT intrusion detection: Contributions of XGBoost, LightGBM, and random forest," *Future Internet*, vol. 17, no. 11, p. 529, Nov. 2025, doi:10.3390/fi17110529.

[16] A. Shaikhanova, O. Kuznetsov, A. Tokkuliyeva, K. Ayapbergenov, S. Olzhas, and T. Danir, "Security audit of IoT device networks: A reproducible machine learning framework for threat detection and performance benchmarking," *Sensors*, vol. 25, no. 24, p. 7519, Dec. 2025, doi:10.3390/s25247519.

[17] Z. Fan and Z. You, "Research on network intrusion detection based on XGBoost algorithm and multiple machine learning algorithms," *Theor. Nat. Sci.*, vol. 31, no. 1, pp. 161–166, Mar. 2024, doi:10.54254/2753-8818/31/20241171.

[18] H. F. Soon, A. Amir, H. Nishizaki, N. A. H. Zahri, L. M. Kamarudin, and S. N. Azemi, "Evaluating tree-based ensemble strategies for imbalanced network attack classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 1, 2024, doi:10.14569/IJACSA.2024.01501111.

[19] L. K. S. Kumar et al., "Anomaly-based intrusion detection on benchmark datasets for network security: A comprehensive evaluation," *Sci. Rep.*, vol. 16, no. 1, Mar. 2026, doi:10.1038/s41598-026-38317-w.

[20] M. Zakariah, S. A. AlQahtani, and M. S. Al-Rakhami, "Machine learning-based adaptive synthetic sampling technique for intrusion detection," *Appl. Sci.*, vol. 13, no. 11, p. 6504, May 2023, doi:10.3390/app13116504.

[21] R. Almuhanna and S. Dardouri, "A deep learning/machine learning approach for anomaly based network intrusion detection," *Front. Artif. Intell.*, vol. 8, Sep. 2025, doi:10.3389/frai.2025.1625891.