Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 6, 2026

MACHINE LEARNING-BASED NETWORK TRAFFIC ANALYSIS FOR IDENTIFYING CYBER ATTACKS USING FLOW-LEVEL FEATURES

[1]Hafsa Anwar
[1]Faculty of Computing, Riphah International University, Islamabad
[1]khafsakhan1212@gmail.com

DOI:-https://doi.org/10.5281/zenodo.20613153

Keywords
Network traffic analysis; cyber attack detection; intrusion detection system; machine learning; CICIDS2017; Random Forest; XGBoost

Article History
Received: 20 May, 2026
Accepted: 07 June, 2026
Published: 09 June, 2026

Corresponding Author: *

Abstract
Background: Due to the growing volume and heterogeneity of traffic generated by modern digital services, cyber attacks are increasingly affecting enterprise, academic, cloud, and government networks. The traditional signature-based intrusion detection systems are capable of detecting known attacks but it is not effective when there is a change in attacks or new malicious behaviours emerge. Purpose: This research article presents a machine learning-based network traffic analysis framework in identifying cyber attacks by use of flow-level features. This paper is concerned with binary classification where each network flow is classified as benign or malicious. Procedure: The proposed framework is based on the CICIDS2017 intrusion detection dataset that contains labelled benign and attack traffic, packet captures and flow-based CSV files. The methodology consists of the data cleaning, label encoding, feature selection, train-test splitting, supervised model training and performance evaluation. The choice of the Logistic Regression, Decision Tree, Random Forest, and XGBoost are made to offer the baseline and the ensemble-based classification performance. Evaluation: Accuracy, precision, recall, F1-score, false positive rate and confusion matrix is used to evaluate the models. These measures are chosen since accuracy in itself can be deceptive in unbalanced datasets of intrusion detection. Contribution: The article has contributed to a structured research design, mathematical formulation, and experimentation procedure that can be direct implemented in Python to identify cyber attacks. It also points out practical concerns, including imbalance in classes, false alarms, biased dataset and the discrepancy between the benchmark performance and the real deployment of the network.

I. Introduction

Due to the interconnection of business processes, public services, education systems, industrial control settings, and financial platforms through local networks and internet-based infrastructure, cybersecurity has become a key requirement of modern organizations. This enhanced connectivity enhances communication and operational efficiency, but it also increases the attack surface. Weak authentication, vulnerable services, misconfigured servers, insecure web applications, and open network ports can be used by malicious users. This makes organizations need mechanisms that are able to monitor traffic and detect malicious behaviour prior to causing severe harm.

Network traffic analysis refers to the procedure of analyzing communication patterns between devices, applications, protocols and services. Traffic analysis is applied in the field of cybersecurity to detect abnormal communication, suspicious port activity, unusual packet rates, denial-of-service behaviour, scanning behaviour and brute-force attempts. It may be a packet-based analysis, where the individual packet header and payloads are analysed, or a flow-based analysis, where groups of packets sharing common properties are aggregated into statistical features. Flow-level traffic analysis is appealing to intrusion detection as it reduces the amount of data and retains helpful behavioural indicators flow duration, packet counts, byte counts, inter-arrival times and flag statistics.

The classical intrusion detection systems are usually based on signature-based rules. A signature-based system compares the traffic it observes to known attack patterns and raises an alert when a match is found. This method works well with familiar attacks, but cannot be extended. Attackers often alter payloads, swap command sequences, alter packet timing, or encrypt channels to evade static signatures. Consequently, signature-based detection might not be effective when dealing with new, unknown or altered attack patterns. Detecting based on anomalies is trying to overcome this shortcoming by learning normal traffic behaviour and identifying deviations. Machine learning aids anomaly-based and hybrid detection since it is able to learn patterns to classify future instances of traffic.

This paper is a research done on machine learning-based network traffic analysis, to identify cyber attacks. The dataset used in the proposed study is the CICIDS2017 dataset since it offers labelled network traffic and various attack scenarios. According to the Canadian Institute of Cybersecurity, CICIDS2017 is a dataset of intrusion detection evaluation, which contains benign and malicious traffic, packet captures, labelled flows, and machine-learning-ready CSV files [1], [2]. The dataset is popular in network intrusion detection studies because it includes a variety of types of attack traffic, and more realistic traffic profiles than older datasets like KDD99.

A. Research Problem

The research issue that this paper aims to address is that it is challenging to detect cyber attacks in high network traffic. The process of manual inspection is not feasible since enterprise networks produce millions of packets and flows. Rule-based systems can be helpful, yet this system needs continual updates and can fail when the attacks are altered. Consequently, an analysis of network flow characteristics and the classification of traffic as benign or malicious with a balanced and reliable performance is required.

An effective model of cyber attack detection should minimize the number of false negatives and false positives. False negative is dangerous as malicious traffic will be treated as benign and will go undetected. False positive is also expensive since the benign traffic is reported as malicious, which add an extra burden to the workload of security analysts. This study thus compares models based on various measures rather than basing on accuracy alone.

B. Research Gap

Most intrusion detection literature claims that a model can be very accurate on benchmark datasets, but a very accurate model does not necessarily imply that a model will be useful in practice. In the case of an imbalanced dataset, a classifier can achieve a high accuracy by predicting the majority class more often. Additionally, other studies fail to state their

mathematical formulation, feature processing or experimental pipeline, making it challenging to reproduce.

The other gap is that flow level benchmark data sets might have dataset specific patterns that do not necessarily reflect actual enterprise traffic. Dube asserts that the condensed version of traffic-flow of CICIDS2017 is popular in supervised machine learning but its practical sufficiency needs to be scrutinized [5]. This implies that research must not only attempt to achieve maximum benchmark performance; it must also address preprocessing, validation, class imbalance, and deployment limitations.

C. Research Objectives

1. To determine the flow-level network traffic properties in detecting cyber attacks.
2. To develop a supervised machine learning system to classify normal and suspicious traffic.
3. Compare Logistic Regression, Decision Tree, Random Forest and XGBoost to intrusion detection.
4. To assess the performance of detection in terms of accuracy, precision, recall, F1-score, false positive rate and confusion matrix.
5. To determine effective constraints on the application of benchmark intrusion detection datasets in real world cyber attack detection.

D. Research Questions

1. What flow level traffic characteristics are helpful in detecting cyber attacks?
2. To what extent can controlled machine learning classifying benign and malicious traffic?
3. What is the most balanced machine learning model that can be used to detect cyber attacks?
4. What can we say about the practical reliability of intrusion detection models when we consider both false positives and false negatives?

II. Literature Review

Intrusion detection based on machine learning has received a lot of importance due to the dynamism and harderness in detecting cyber attacks using fixed rules alone. The early work on intrusion detection algorithms was supported by older data used, such as DARPA98, KDD99, and NSL-KDD, which has been criticized due to age of data used, lack of diversity, and unrealistic attack patterns. More recent datasets like CICIDS2017 and UNSW-NB15 offer more detailed profiles of traffic and more applicable attack scenarios.

To overcome the shortcomings of older intrusion detection datasets, Sharafaldin, Lashkari, and Ghorbani proposed CICIDS2017 [1]. The dataset was made to cover both benign traffic, as well as several categories of attacks which included brute-force, DoS, DDoS, botnet, infiltration, port scanning, and web attacks. The official CIC source offers PCAP files and CSV files, which makes it possible to analyze them both at the packet-level and the flow-level [2]. This is why CICIDS2017 is appropriate in a supervised machine learning since the records of the traffic are labelled.

CICFlowMeter is significant in the flow-based intrusion detection since it identifies the characteristics of the bidirectional traffic-flow of packet captures. According to the Canadian Institute of Cybersecurity, CICFlowMeter can create two-way flows and calculate over 80 statistical network traffic measures, including duration, number of packets, number of bytes and packet length statistics [3]. These features that were extracted enable machine learning models to learn patterns based on summarized traffic behaviour in lieu of full packet payloads.

Rodriguez et al. tested machine learning algorithms to detect intrusion based on traffic flow using CICIDS2017 [4]. Their work demonstrates that flow-based classification could be effective in terms of appropriate preprocessing and model selection. The use of tree-based models can be beneficial in this area as non-linear relationships between features can be captured. The effectiveness of these models is however dependent on the quality of features, distribution of classes and the method of evaluation.

Random Forest is also commonly employed in intrusion detection since it uses a series of decision trees and mitigates overfitting. Breiman proposed Random Forests as a collection of tree predictors each of which relies on a random vector sampled independently [7]. This can be applied in network traffic classification since various features can imply

various types of attack. An example is that a high packet rate can be an indication of flooding and repeated connection attempts could be indications of brute force or scanning behaviour.

Another powerful ensemble technique is XGBoost. Chen and Guestrin introduced XGBoost as a scalable tree boosting platform which is designed to provide efficient and regularized gradient boosting [8]. XGBoost is applicable to intrusion detection as it can learn a complex boundary of decision-making and minimize the errors in a sequential manner. Nevertheless, it also needs to be carefully tuned in terms of hyperparameters, and being highly successful on a benchmark dataset should not be directly interpreted as readiness to deploy it in the real world [11].

UNSW-NB15 can also be applied in intrusion detection studies. According to the official UNSW source, it was created using IXIA PerfectStorm tool in a cyber range lab and includes normal traffic and nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms [6]. Whilst the current study relies on CICIDS2017 as the primary dataset, UNSW-NB15 is helpful in future validation since it provides an alternative distribution of attacks and a collection environment.

One of the main weaknesses in the literature is the inclination to focus on accuracy too much. Accuracy is helpful but during an imbalanced class, it may be misleading. Recall, precision, false positive rate, and F1-score offer a more comprehensive perspective of operational utility in detecting cyber attacks. A model that accurately detects the majority of attacks, but produces a high number of false alarms, might not be feasible [12]. Equally, a model that has a low false alarm rate and a low attack detection rate can put the network at a high-risk level [13, 14].

Table I: ***Summary of Related Literature***

Study / Source	Dataset / Method	Key Contribution	Limitation / Relevance to This Study
Sharafaldin et al. [1]	CICIDS2017 dataset	Introduced a modern IDS dataset with benign and attack traffic for evaluation.	Requires careful preprocessing and class imbalance treatment.
CIC official dataset page [2]	PCAP and CSV files	Provides labelled traffic files and supports machine learning experimentation.	Benchmark data may not fully represent every real network environment.
CICFlowMeter [3]	Bidirectional flow generator	Extracts more than 80 statistical features from packet captures.	Flow summaries may lose packet-level details and payload context.
Rodriguez et al. [4]	CICIDS2017 with ML classifiers	Demonstrated effectiveness of traffic-flow-based machine learning models.	Performance depends on selected features, preprocessing, and validation method.
Dube [5]	Critical analysis of CICIDS2017	Warned that summarized flow data may have limited practical adequacy.	Supports the need for cautious interpretation of benchmark results.
UNSW-NB15 [6]	Cyber range traffic dataset	Provides an alternative benchmark with nine attack families.	Useful for future validation but not the main dataset of this article.
Breiman [7]	Random Forest algorithm	Established ensemble tree learning with reduced overfitting.	Needs enough diverse data and careful parameter selection.
Chen and Guestrin [8]	XGBoost algorithm	Presented scalable and regularized gradient boosting.	Requires tuning and can overfit if configuration is poor.

III. Proposed Methodology

In this study, a supervised machine learning approach will be adopted for cyber-attacks detection using network traffic data. For supervised learning each record has input features and a known label. The model identifies the relationship between the traffic features and the target class, thus enabling the model to classify new network flows as benign or malicious. The methodology proposed is structured along the research pipeline to select datasets, preprocess datasets, select features, train models, test models, and evaluate the performance of the trained model.

A. Dataset Selection

The CICIDS2017 dataset will be used for the study, which includes benign network traffic and various kinds of cyber-attacks. This dataset can be used for machine learning tasks as it is available in csv format, and it contains flow-level traffic records. The study will employ a binary classification technique; class 0 represents benign traffic and class 1 represents malicious traffic.

The reason why binary classification is chosen is the first goal of an intrusion detection system is to determine if traffic is normal or harmful. Once a reliable binary model is developed, it can be further expanded to multi-class classification to detect specific attack types like DDoS, PortScan, Botnet, Brute Force, Web Attack and Infiltration.

B. Data Preprocessing

The data will be preprocessed prior to training the machine learning models. This step is crucial as the raw network traffic data could be duplicated, missing, infinite, and labeled inconsistently. To prevent a biased result, duplicate records will be deleted. Values that are missing, null, NaN, and infinite will be either deleted or replaced, so that the selected algorithms can be correctly processed.

The classes then will be translated to numbers. All traffic will be coded as 1 for malicious traffic and 0 for benign traffic. Numerical features will also be scaled by an appropriate scaling technique like Min-Max scaling or standardization. This will help to ensure that features represented by higher numbers don't take over the learning process.

Class balancing techniques, like class weighting or SMOTE can be used if the dataset is heavily imbalanced, with more benign than malicious instances. This will help minimize the chance of the model being skewed towards the highest majority class. Lastly, the cleaned data set will be split into training and testing sets, with 70% of the data allocated to the training set and 30% to the testing set.

C. Feature Selection

The most relevant traffic characteristics to cyber-attack detection will be identified through feature selection. There are a large number of flow-level features in the CICIDS2017 dataset; not all of them are equally important for classification. As a result, feature selection will be useful in eliminating irrelevant and redundant features, decreasing the computational complexity, and enhancing the model generalization.

Important features can be selected by using correlation filtering, information gain or Random Forest feature importance. Useful traffic features are such as flow duration, the number of total forward packets, the number of total backward packets, flow bytes per second, mean packet length, standard deviation of packet length, flow inter-arrival time, forward packet length, backward packet length, and the number of TCP flags. The following features are important because cyber-attacks can affect the volume, timing, frequency, and behaviour of network communications.

D. Machine Learning Models

Four supervised machine learning models will be used in this study: Logistic Regression, Decision Tree, Random Forest, and XGBoost.

Logistic Regression will be used as the baseline model. Easy to understand, easily explained and convenient for comparison with more sophisticated machine learning models. It assists in deciding if a network flow is more likely to be good or bad.

Decision Tree will be used since it is an algorithm that develops decision paths based on rules and can find non-linear relationships within the data set. It has a deepness and splitting conditions that must be controlled, however, to minimize overfitting.

Random Forest will be used since it uses multiple decision trees and its final decision will be based on the maximum voting. This provides better stability and reliability than a single decision tree, particularly under complex network traffic conditions.

We will use XGBoost as it is a very powerful boosting algorithm which is built incrementally and weak learners are improved with time. It also contains regularization to mitigate overfitting and enhance performance with structured traffic data.

Model Training and Testing

Once the data is preprocessed and features selected, the selected models will be trained with the training set. In this learning phase the models will be trained with patterns of the chosen traffic features and the associated labels. After training the models will be evaluated on the test data which is not seen by the models.

The significance of this testing stage is that it demonstrates the ability of the models to correctly classify new network traffic not provided during model training. The outcome will be used to see if the suggested method is applicable to data it was not trained on before.

F. Evaluation Metrics

Accuracies, precisions, recalls, F1 score, false positive rate, and confusion matrix will be used to evaluate the models. However, accuracy will not be used alone as there can be imbalanced intrusion detection datasets.

Precision will display the number of attacks predicted that were actually attacks. Recall will display the number of true attacks that are detected. The F1 score will give a balance score between precision and recall. The false positive rate will indicate the percentage of non-threat traffic that is incorrectly identified as a threat.

Recall is particularly crucial in detecting cyber-attacks, since a false negative occurs when a genuine attack is not detected. But precision is also significant as too many false alarms can lead to analyst skepticism and strain the security team's workload. So the best model would be chosen on the basis of a balance of performance, not accuracy.

G. Summary

The methodology presented is proposed to load the proposed concepts into a clear and systematic procedure to detect cyber attacks with supervised machine learning. The CICIDS2017 data will be preprocessed including cleaning, encoding, scaling, balancing, and training and testing partition. Appropriate traffic features will be chosen, and four machine learning models will be trained and tested. The final model to be selected will be the one having a high recall, good precision, reliable F1 score, and a low false positive rate. This renders the methodology useful in designing a real-world and measurable intrusion detection system.

Table II: ***Proposed Experimental Configuration***

Phase	Activity	Technique / Tool	Expected Output
Dataset	Acquire CICIDS2017 files	Official CIC source	Raw CSV traffic files
Cleaning	Remove null, infinite, and duplicate records	Pandas / NumPy	Clean dataset
Encoding	Convert labels into binary classes	Label mapping	Benign = 0, Attack = 1
Scaling	Normalize numeric features	Min-Max or Standard scaler	Scaled feature matrix
Selection	Select relevant flow attributes	Correlation / importance ranking	Reduced feature set
Splitting	Separate training and testing data	70:30 train-test split	Training and testing subsets
Training	Train four classifiers	Scikit-learn / XGBoost	Trained models
Testing	Predict labels on unseen data	Model prediction	Predicted labels

Phase	Activity	Technique / Tool	Expected Output
Evaluation	Measure model performance	Confusion matrix and metrics	Best model selection

IV. Mathematical Model of the Analytical Technique

The cyber attack detection task is formulated as a supervised binary classification problem. Each network flow is treated as one observation, and the objective is to learn a mapping from selected traffic features to a benign or malicious class label.

A. Dataset Representation

Let the complete dataset be represented as:

1. $$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$$

where n is the total number of network-flow records, $\mathbf{x}_i$ is the feature vector of the i^{th} network flow, and y_i is its class label. Each feature vector is defined as:

2. $$\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}, ..., x_{im}]^T \in \mathbb{R}^m$$

where m is the number of selected features. These features may include flow duration, packet length statistics, byte rate, packet rate, inter-arrival time, and TCP flag counts.

The binary class label is defined as:

3. $$y_i = \begin{cases} 0, & \text{if the traffic is benign} \\ 1, & \text{if the traffic is malicious} \end{cases}$$

The classifier learns a decision function:

4. $$f_\theta(\mathbf{x}_i) \rightarrow \hat{y}_i, \qquad \hat{y}_i \in \{0,1\}$$

where f_θ represents the trained machine learning model and $\hat{y}_i$ is the predicted class label.

B. Feature Normalization

Since network traffic features may have different scales, Min-Max normalization is applied before model training:

5. $$x'_{ij} = \frac{x_{ij} - x_j^{min}}{x_j^{max} - x_j^{min} + \epsilon}$$

where x_{ij} is the original value of feature j for record i, x_j^{min} and x_j^{max} are the minimum and maximum values of feature j, and ϵ is a very small constant used to avoid division by zero.

C. Logistic Regression Model

In Logistic Regression, the probability that a network flow is malicious is estimated using the sigmoid function:

6. $$p_i = P(y_i = 1|\mathbf{x}_i) = \sigma(z_i) = \frac{1}{1+e^{-z_i}}$$

where:

7. $$z_i = \beta_0 + \sum_{j=1}^{m} \beta_j x_{ij}$$

Here, β_0 is the intercept and β_j represents the coefficient of feature j. The final prediction is made using a threshold τ, usually 0.5:

8. $$\hat{y}_i = \begin{cases} 1, & \text{if } p_i \geq \tau \\ 0, & \text{if } p_i < \tau \end{cases}$$

D. Random Forest Model

Random Forest predicts the final class by combining the outputs of multiple decision trees. If T decision trees are trained, the predicted class is obtained through majority voting:

9. $$\hat{y}_i = \text{mode}\{h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), ..., h_T(\mathbf{x}_i)\}$$

Equivalently, for binary classification, the model can be expressed as:

10. $$\hat{y}_i = \arg\max_{c \in \{0,1\}} \sum_{t=1}^{T} \mathbb{I}\,(h_t(\mathbf{x}_i) = c)$$

where h_t is the t^{th} decision tree and $\mathbb{I}(.)$ is an indicator function.

E. XGBoost Objective Function

XGBoost is an ensemble learning model that builds trees sequentially. Its prediction after K trees is expressed as:

11. $$\hat{y}_i^{(K)} = \sum_{k=1}^{K} f_k\,(\mathbf{x}_i)$$

The objective function minimized by XGBoost is:

12. $$Obj^{(K)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(K)}\right) + \sum_{k=1}^{K} \Omega\,(f_k)$$

where $l(y_i, \hat{y}_i)$ is the prediction loss and $\Omega(f_k)$ is the regularization term. The regularization term is commonly defined as:

13. $$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \parallel \mathbf{w}_k \parallel^2$$

where T_k is the number of leaves in tree k, $\mathbf{w}_k$ is the vector of leaf weights, and γ and λ are regularization parameters used to control model complexity and reduce overfitting.

F. Confusion Matrix and Evaluation Metrics

For cyber attack detection, the confusion matrix is defined using true positives, true negatives, false positives, and false negatives:

14. $$TP = \sum_{i=1}^{n} \mathbb{I}\,(y_i = 1 \wedge \hat{y}_i = 1)$$
15. $$TN = \sum_{i=1}^{n} \mathbb{I}\,(y_i = 0 \wedge \hat{y}_i = 0)$$
16. $$FP = \sum_{i=1}^{n} \mathbb{I}\,(y_i = 0 \wedge \hat{y}_i = 1)$$
17. $$FN = \sum_{i=1}^{n} \mathbb{I}\,(y_i = 1 \wedge \hat{y}_i = 0)$$

In this research, a true positive means attack traffic

correctly classified as attack, while a false negative means attack traffic incorrectly classified as benign. False negatives are especially serious because they represent missed cyber attacks.

The evaluation metrics are calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$FPR = \frac{FP}{FP + TN}$$

Accuracy measures overall correctness, precision measures how many predicted attacks are actually attacks, recall measures how many real attacks are successfully detected, and FPR measures how often benign traffic is incorrectly flagged as malicious. Therefore, recall and FPR are particularly important for evaluating the practical reliability of cyber attack detection systems.

V. Experimentation Design and Process Diagram

The experimentation design follows a controlled machine learning pipeline. The purpose is to ensure that model comparison is fair and reproducible. All models must be trained on the same training data and evaluated on the same test data. Preprocessing must be applied carefully to prevent data leakage. For example, scaling parameters should be learned from the training set and then applied to the test set, rather than being learned from the full dataset before splitting.

The process diagram below summarizes the complete experimental workflow from dataset acquisition to final model selection.

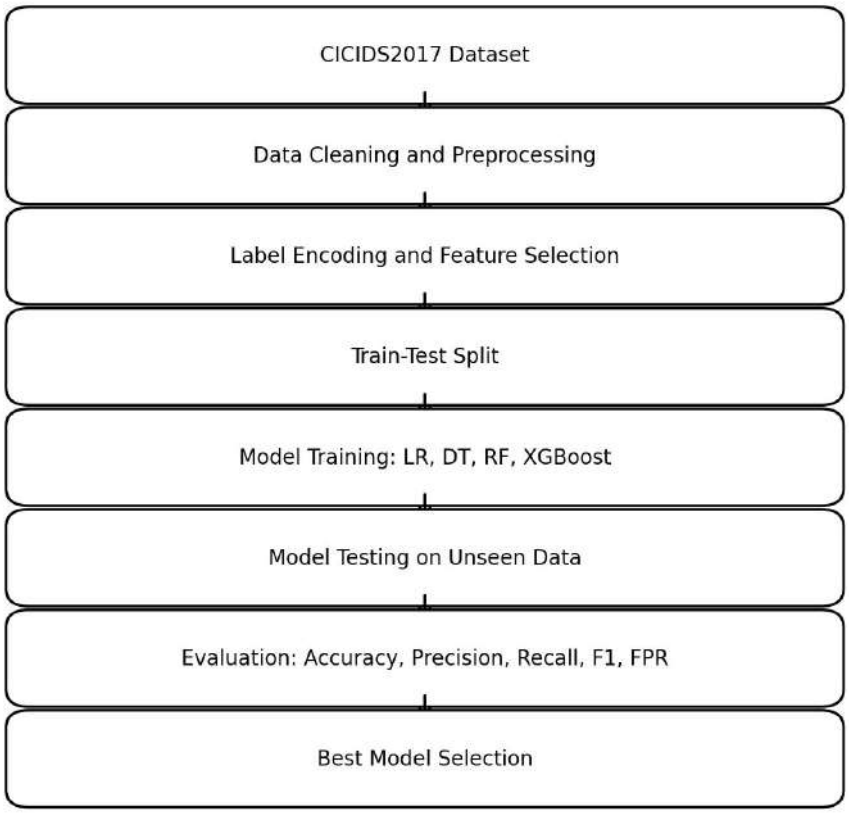

Figure 1: Proposed machine learning process for network traffic-based cyber attack detection

Table III: *Model Performance Reporting Template*

Model	Accuracy	Precision	Recall	F1-score	False Positive Rate
Logistic Regression	To be computed	To be computed	To be computed	To be computed	To be computed
Decision Tree	To be computed	To be computed	To be computed	To be computed	To be computed
Random Forest	To be computed	To be computed	To be computed	To be computed	To be computed
XGBoost	To be computed	To be computed	To be computed	To be computed	To be computed

VI. Expected Discussion

The anticipated discussion will involve a comparison of the four models that were identified using the reporting template in Table III. Logistic Regression is likely to give a helpful baseline but is less likely to work well when there is a very non-linear relationship between features and attack labels. The decision tree can be used to capture non-linear patterns, however, it can also overfit the training data unless pruning or depth control is used.

Random Forest will deliver more robust and consistent classification since it combines various trees. XGBoost can also achieve strong performance since boosting corrects errors made previously sequentially, and it also applies regularization. But the ultimate decision must not be relying solely on accuracy. When the model is highly accurate but has poor recall, the model might fail to detect many attacks. In the case where it has a high recall but low precision then it can result in an excessive number of false alarms.

There should also be an examination of what features contribute the most towards classification. Explaining whether the model is learning significant traffic behaviour or data-specific shortcuts can be achieved by feature importance. As an example, flow-duration features, packet-length features, and packet-rate features might be security-relevant, whereas identifiers, such as IP address or timestamp, might cause overfitting, since they capture the circumstances of datasets collection, but not the behaviour of an attack.

Class imbalance is another key topic of discussion. In the event benign traffic is vastly large in comparison to attack traffic, the model might be biased toward the benign group. That is why recall, precision, F1-score and false positive rate should be viewed as a combination. An effective model of intrusion detection must detect attacks with high reliability and false alarms should be minimal to allow security analysts to manage them effectively.

VII. Limitations and Threats to the Validity.

There are a number of limitations in this study. First, the CICIDS2017 is a benchmark dataset that is gathered in a controlled setting. The enterprise networks in the real world can have a greater number of diverse applications, encrypted traffic, user behaviours and background noise. Thus, a good benchmark performance does not necessarily imply a good real world deployment performance.

Second, binary classification is firstly proposed in this article. Binary classification is applicable in deciding whether or not traffic is benign or malicious but not the type of attack. A security analyst might require further details like whether the assault is a DDoS, PortScan, botnet, brute force, or web-based exploitation.

Third, flow-based properties summarize the behaviour of packets and can lose the details contained in the payload. This can be advantageous in terms of privacy and effectiveness, but it can also diminish the

visibility of attacks to the application-layer. Fourthly, data leaks can be caused by lack of careful design of the experiment. The leakage of data would result in exaggerated performance and unreliable conclusions.

Lastly, the proposed models need to be tuned and validated. Hyperparameters like tree depth, number of trees, learning rate, and regularization strength can be of great importance. Hence, cross-validation and sensitivity analysis should be incorporated in the implementation in the future.

VIII. Future Work and conclusion.

The research article was a proposal of a machine learning-based structure of detecting cyber attacks based on network traffic analysis. The paper concentrated on flow-level characteristics of CICIDS2017 and developed the problem of cyber attack detection as a supervised binary classification task. The choice of the logistic regression, decision tree, random forest and XGBoost were selected to compare them with each other as they represent the baseline and advanced ensemble learning methods.

The research problem, research gap, objectives, questions, literature review, mathematical model and the design of the experiment were presented in the paper. The accuracy, precision, recall, F1-score, false positive rate, and confusion matrix are the proposed evaluation metrics since intrusion detection needs a balance in performance. A model that seems correct, but lacks attacks or generates too many false alarms might not be useful in practice.

The experiment in Python should be implemented in future work and the actual performance of the model reported. It can be further extended to the multi-class attack classification, cross-dataset validation using UNSW-NB15 or CSE-CIC-IDS2018, deep learning models, adversarial robustness testing, and live traffic validation. These extensions would enhance the feasible soundness of the recommended intrusion detection framework.

References

[1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018, pp. 108-116.

[2] Canadian Institute for Cybersecurity, "Intrusion detection evaluation dataset (CIC-IDS2017)," University of New Brunswick. Accessed: May 4, 2026. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[3] Canadian Institute for Cybersecurity, "CICFlowMeter," University of New Brunswick. Accessed: May 4, 2026. [Online]. Available: https://www.unb.ca/cic/research/applications.html

[4] M. Rodriguez, A. Alesanco, L. Mehavilla, and J. Garcia, "Evaluation of Machine Learning Techniques for Traffic Flow-Based Intrusion Detection," Sensors, vol. 22, no. 23, Art. no. 9326, 2022.

[5] R. Dube, "Faulty use of the CIC-IDS 2017 dataset in information security research," Journal of Computer Virology and Hacking Techniques, vol. 20, pp. 203-211, 2024.

[6] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in Proceedings of the Military Communications and Information Systems Conference, 2015.

[7] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[8] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.

[9] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153-1176, 2016.

[10] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, 2019.

[11] F. Amin, I. Said, and M. A. Butt, "AI-based cybersecurity solutions: Securing information and privacy in the evolving digital age," *Journal of Engineering and Computational Intelligence Review*, vol. 2, no. 2, pp. 95–111, 2024.

[12] U. Imtiaz and H. Elbedour, "Cybersecurity risk management in the digital era: The strategic value of ethical hacking," *Spectrum of Engineering Sciences*, pp. 1076–1086, 2025.

[13] S. M. H. Shah, F. Amin, and A. Khan, "Cyber-Resilient Mobile Edge Computing: A Deep Neural Approach for Secure and Efficient Task Offloading," *The Asian Bulletin of Big Data Management*, vol. 5, no. 1, pp. 200–215, 2025.

[14] U. Imtiaz, "Investigating the impact of phishing attacks on organizational cybersecurity posture," *Spectrum of Engineering Sciences*, pp. 1758–1780, 2025