JIRBDAI

Journal of Institutional Research, Big Data Analytics and Innovation

Comparative Analysis of Selected Machine Learning Techniques for HTTP-Based Distributed Denial of Service (DDOS) Attack Detection in Web Applications

Binyamin A. Ajayi[1], Audu Elizabeth[2], Patrick K. Bauti[3*]

[1,2]Department of Computer Science, Faculty of Natural and Applied Sciences, Nasarawa State University, Keffi, Nigeria.
[3]Department of Computer Science, Federal Polytechnic, Nasarawa, Nasarawa State
Corresponding Author: bautipatrick@gmail.com

ABSTRACT

HTTP-based Distributed Denial of Service (DDoS) attacks threaten the availability of web applications by generating large volumes of requests that may resemble legitimate user activity. Because application-layer requests can appear valid, detection systems must distinguish malicious traffic from normal traffic while controlling false alarms. This study comparatively evaluated five supervised machine-learning classifiers—Random Forest, J48 Decision Tree, Logistic Regression, Naïve Bayes and Multilayer Perceptron (MLP)—for HTTP DDoS detection using the CICDDoS2019 dataset and the WEKA data-mining environment. The working dataset contained 55,508 traffic records and 78 attributes, including the class attribute. The records were pre-processed, non-predictive identifiers and a potentially leakage-related label field were excluded, and numeric attributes were normalized. The reported experiment allocated approximately 65% of the records to training and 35% to testing. Accuracy, precision, recall/true-positive rate and confusion-matrix counts were used to compare the classifiers. Random Forest produced the highest reported accuracy (99.9125%), with 3 false positives and 14 false negatives. J48 followed with 99.8816% accuracy, 7 false positives and 16 false negatives. Logistic Regression and MLP recorded 99.4853% and 99.4647% accuracy respectively, while Naïve Bayes recorded 95.9440% and the highest false-positive count (693). The results indicate that Random Forest and J48 were the strongest classifiers in this experiment, although J48 may be attractive where interpretability is important. The findings are limited to the dataset, preprocessing and experimental configuration used; they do not establish equivalent performance on all live networks. The study recommends independent validation, transparent reporting of data-splitting procedures, and testing under realistic traffic conditions before operational deployment.

KEYWORDS

HTTP DDoS; Machine Learning; Intrusion Detection; CICDDoS2019; Random Forest; J48 Decision Tree; WEKA; Web Application Security

I. INTRODUCTION

Web applications have become essential to commerce, education, public administration, communication and the delivery of digital services. Their availability is therefore an important operational and security requirement. As increasingly depend on internet-facing services, the systems that support them are exposed to a wide range of attacks. Distributed Denial of Service (DDoS) attacks are particularly disruptive because they attempt to exhaust the resources needed to serve legitimate users. An attack can degrade response times, interrupt access or make a service unavailable altogether, resulting in financial cost, reduced trust and interruption of business processes (Ugbolu, Adegboyega & Osewa, 2023). DDoS attacks use traffic originating

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

from multiple devices or network locations. In an application-layer attack, the adversary directs large numbers of requests towards a web service. HTTP requests may be syntactically valid and may resemble ordinary browsing behaviour, making it difficult to distinguish an attack from a sudden but legitimate increase in demand. Unlike a simple volume threshold, an effective detector needs to consider the pattern and characteristics of traffic, and it must do so with sufficient speed to support mitigation (Alshamrani, A., et al. 2025).

Conventional signature-based detection depends on previously patterns or rules. This can be effective against known attacks but is less reliable when attack characteristics change or a previously unseen pattern appears. Anomaly-based methods attempt to identify deviations from a baseline, but legitimate traffic fluctuations can produce false alerts. These limitations have motivated the use of supervised machine learning, in which a classifier learns associations between traffic features and labelled examples of benign and malicious activity. Machine-learning approaches differ in their assumptions, complexity, interpretability and computational requirements. Logistic Regression provides a comparatively simple probabilistic decision boundary. Naïve Bayes uses Bayes' theorem and assumes conditional independence between features. Random Forest aggregates multiple decision trees and can represent complex feature interactions. J48, an implementation of the C4.5 decision-tree approach available in WEKA, produces a set of decision rules that can be inspected. Multilayer Perceptron (MLP) is a neural-network classifier capable of modelling non-linear relationships, although its training may require more computation and parameter tuning (Thalapala, 2025).

Performance values reported in separate studies are difficult to compare when researchers use different datasets, class definitions, preprocessing methods, train-test splits and evaluation metrics. A controlled comparison of several classifiers on the same dataset can provide more useful evidence for model selection. The present study therefore compares five algorithms using HTTP-focused traffic from CICDDoS2019 and a common WEKA-based workflow. The objective is not simply to identify the model with the largest accuracy value, but also to consider false positives, false negatives, precision, recall and practical deployment trade-offs (Al-Khazraji, et al. 2024). DDoS attacks are commonly described as volumetric, protocol-based or application-layer attacks. Volumetric attacks seek to consume available bandwidth; protocol attacks exploit weaknesses or resource costs in network protocol handling; and application-layer attacks target the processing capacity of services such as HTTP servers. The categories can overlap in real incidents. Application-layer DDoS is challenging because a request may be valid at the protocol level while being malicious in its timing, rate, distribution or relationship to other requests (Singh, R., et al. 2025).

A typical distributed attack involves an attacker, a command-and-control mechanism or other coordination method, compromised devices (often described as bots), and a victim service. Once activated, the bots generate traffic towards the target. The resulting demand can exhaust bandwidth, CPU time, memory, connection slots or application resources. Modern connected devices and cloud-hosted infrastructure can increase the scale and diversity of traffic sources, complicating the task of filtering harmful requests without disrupting legitimate clients (Pavlovich, 2026). An intrusion detection system (IDS) monitors events or traffic and identifies activity that may violate security expectations. In a machine-learning IDS, traffic is represented through attributes such as packet length, flow duration, packet rates, inter-arrival times, protocol indicators and forward/backward flow statistics. A model uses these attributes to assign a class label. The resulting prediction can support an alert or be passed to a mitigation layer. Automated blocking should, however, be treated cautiously because a false positive can deny service to a genuine user (Rahman, 2025)

The study uses CICDDoS2019, a labelled benchmark dataset associated with the Canadian Institute for Cybersecurity at the University of New Brunswick. The dissertation filtered the

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

dataset for the HTTP-related classification task. Its final working dataset was reported as 55,508 records, consisting of 48,255 attack flows and 7,253 normal flows, and 78 attributes including the class field. This distribution is imbalanced: attack instances constitute approximately 86.9% and normal instances approximately 13.1%. Under such conditions, accuracy alone can conceal weaknesses in the minority class. Precision, recall, confusion-matrix counts and class-specific measures should therefore be interpreted alongside overall accuracy (Rathore, S., et al. 2026). The motivation for this comparison is practical as well as methodological. A model suitable for a high-capacity security platform may not be the best choice for a resource-constrained deployment. A highly interpretable classifier can be useful to administrators who need to understand why traffic was flagged, whereas an ensemble may provide a stronger overall classification result. A neural network may capture complex relationships but require more training effort. A comparison that reports both classification outcomes and practical limitations can support more defensible selection of a detection technique (Thalapala, 2025).

HTTP-based DDoS attacks can disrupt web applications by exhausting server and network resources with large volumes of requests that may resemble legitimate activity. If detection occurs late, the service may already have degraded or become unavailable. If the detector is too aggressive, legitimate users may be incorrectly flagged, generating unnecessary alerts or causing harmful blocking decisions (Xu, 2025). Traditional signature-based systems rely on known patterns and may not identify new or modified attacks. Anomaly-based systems can be sensitive to changing traffic conditions and may produce false positives. Although machine-learning methods have shown promise, no algorithm should be assumed to be best across all datasets and deployment conditions. Published results are also not always directly comparable because studies may use different traffic datasets, features, validation procedures and performance measures. The specific problem addressed in this study is the need for a consistent comparison of selected machine-learning classifiers for HTTP DDoS detection using the same prepared dataset and evaluation environment. The study compares Random Forest, J48 Decision Tree, Logistic Regression, Naïve Bayes and MLP using CICDDoS2019 traffic processed in WEKA. It examines not only accuracy but also precision, recall and false-positive/false-negative outcomes, so that the selection of a model can take account of both attack detection and the risk of misclassifying legitimate traffic. The study was guided by the following two objectives:

(a). To develop an experimental framework for detecting HTTP-based Distributed Denial of Service attacks using selected supervised machine-learning algorithms and the CICDDoS2019 dataset.

(b). To compare the performance of Random Forest, J48 Decision Tree, Logistic Regression, Naïve Bayes and Multilayer Perceptron using accuracy, precision, recall and confusion-matrix results, and identify the most suitable model under the reported experimental conditions.

II. RELATED WORKS

Machine learning has been increasingly applied to network intrusion and DDoS detection because it can identify statistical relationships in traffic that are difficult to express as fixed rules. The literature reviewed in the source dissertation describes comparisons of classical classifiers, tree-based methods, ensemble learning and neural-network models. A recurring finding is that results depend on the dataset and experimental design; an algorithm that performs

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

well in one setting may not retain the same ranking in another (Ugbolu, Adegboyega, & Osewa, 2023). Logistic Regression is a supervised classifier that estimates the probability of a class using a logistic function. Its strengths include relatively straightforward training, compact representation and interpretability through coefficients. It can provide a useful baseline, particularly when relationships between predictors and the class are reasonably well represented by the model. Its limitations include difficulty representing complex non-linear relationships without feature transformations or interactions (Rahman, 2025).

Naïve Bayes estimates class probabilities using Bayes' theorem and a conditional-independence assumption. It is often computationally efficient and can be useful where rapid classification is important. However, network-flow features may be correlated; where that assumption is strongly violated, probability estimates and classification performance can be affected. In the present experiment, Naïve Bayes provides a lightweight baseline against which the other methods can be assessed (Li, X., et al. 2024). Random Forest combines predictions from multiple decision trees trained using resampled data and randomized feature selection. The ensemble can model non-linear relationships and interactions, and aggregation can reduce the variance associated with an individual tree. Its disadvantages may include a larger model and reduced transparency compared with a single decision tree. The dissertation's literature review reports that several earlier studies found strong Random Forest performance in DDoS classification, including work using IoT and CICDDoS2019-related data. Such findings provide motivation for testing the method, but they cannot substitute for results from the present experiment (Mohammed, & Aziz, 2025).

J48 implements the C4.5 decision-tree approach in WEKA. It recursively selects attributes for splitting and can use pruning to reduce overfitting. Its main advantage is that the resulting tree can be inspected and translated into decision rules. This is valuable in security operations, where analysts may need to understand or audit the conditions associated with an alert. A decision tree can nevertheless be sensitive to changes in data and may overfit if not appropriately configured.

MLP is a feed-forward neural network with an input layer, one or more hidden layers and an output layer. Through weighted connections and non-linear activation functions, it can learn complex boundaries. Training generally involves iterative optimization and backpropagation. Performance may depend on parameter settings, feature scaling and training duration. The dissertation reports that the initial MLP training run took approximately 11 hours and that parameters were subsequently adjusted before the final result was recorded. This experience underscores the need to report configurations and training costs alongside classification scores (Janivasya, Suryawanshi & Sharma, 2024).

The dissertation's empirical review includes studies that reported strong results for J48, Random Forest and optimized MLP, while also noting limitations such as narrow attack focus, incomplete metrics, limited model sets and computational expense. Because the cited literature contains heterogeneous datasets and settings, its reported percentages should be treated as context rather than direct benchmarks for this study. The contribution of the present comparison is the evaluation of five distinct classifier families under a common dataset-preparation and reporting workflow (Janivasya, et al. 2024). The theoretical basis described in the source dissertation includes Statistical Learning Theory and Pattern Recognition Theory. Statistical learning provides a basis for considering how a model learns from labelled examples and how well it generalizes to unseen instances. Pattern recognition concerns the extraction of regularities from observations and the assignment of observations to categories. Together, these perspectives support the task of learning traffic patterns and classifying flows as benign or malicious. They also reinforce the importance of an independent test set: strong performance on training records alone would not establish reliable detection of unseen traffic (Gibson, & Ifenthaler, 2024).

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

III. METHODOLOGY

This section presents the methodology adopted to achieve the research objectives and answer the research questions. It describes the research design, dataset, experimental procedures, and evaluation methods used to train and compare five supervised machine learning classifiers in the WEKA environment. The methodology ensures that all classifiers are evaluated using the same dataset and experimental conditions to support a fair and consistent comparison of their predictive performance.

Research Questions

The analysis was organized around two questions:

(a). How effectively can the selected supervised machine-learning classifiers distinguish HTTP DDoS traffic from normal traffic in the prepared CICDDoS2019 dataset?

(b). Which of the five classifiers provides the strongest combination of accuracy, precision, recall and false-alarm control in the reported experiment?

Research Design

The study adopted an experimental comparative design. The same prepared dataset and WEKA environment were used to train and evaluate five supervised classifiers. This design makes the reported model scores more comparable than results assembled from unrelated datasets, although its conclusions remain dependent on the quality of the split, feature preparation and model configuration.

Dataset and Working Sample

The study used the CICDDoS2019 benchmark dataset, filtered to focus on HTTP-related traffic. Following preprocessing and feature selection, the working dataset was reported to contain 55,508 instances and 78 attributes, including the target class. The reported class distribution was 48,255 DDoS attack flows (approximately 86.9%) and 7,253 normal flows (approximately 13.1%). The original dissertation contains a minor count variation in one table (48,254 versus 48,255 attack records); the article uses the 48,255-figure stated in the summary and class-distribution description. This small discrepancy should be checked against the actual data file before a final journal submission.

Data Preparation

The records were inspected in WEKA for missing or problematic values, and problematic rows were removed during preprocessing. A normalization filter scaled numeric attributes to a common range. The ‘Label’ field described in the dissertation as potentially exposing target information was removed from the predictor set, as was the ‘s/n’ identifier because it represented record identity rather than a meaningful traffic characteristic. The class field was retained as the prediction target. These steps were intended to reduce leakage risk, improve consistency and avoid allowing a record identifier to act as a spurious predictor. The source document describes feature reduction from the original extracted flow features to a working set of 78 attributes. The selected attributes include flow duration, total forward and backward packets and bytes, flow and packet rates, inter-arrival-time statistics, packet-length statistics, TCP flag counts, segment sizes, sub flow statistics and active/idle timing measures. Such features describe the volume, timing and structure of traffic flows and can help classifiers distinguish ordinary behaviour from attack-related patterns.

The dissertation reports a 65:35 training-test allocation, approximately 36,080 training instances and 19,428 test instances. It also mentions 10-fold cross-validation in the methods chapter. Since the relationship between the 65:35 split and the cross-validation procedure is not fully

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

specified in the reported results, the values in this article are presented as the reported experimental outcomes rather than as a claim that the two procedures were combined in a particular way. For reproducibility, a journal submission should state precisely whether cross-validation was performed on the training partition only, whether the final table came from a held-out test set, and whether all preprocessing was fitted on training data only.

Classifiers

Five classifiers were evaluated. Random Forest was used as an ensemble of decision trees. J48 was included as an interpretable tree-based classifier. Logistic Regression represented a compact probabilistic baseline. Naïve Bayes provided a lightweight probabilistic comparator. MLP represented a neural-network approach capable of modelling non-linear patterns. These methods cover different assumptions and complexity profiles, making them appropriate for a comparative experiment.

Evaluation Metrics

The following performance metrics are commonly used to evaluate supervised machine-learning classifiers for detecting attacks in network traffic. They assess overall correctness, the reliability of attack predictions, the ability to detect actual attacks, and the balance between precision and recall.

(a). Accuracy: Accuracy is the proportion of all instances that the classifier correctly classifies, including both normal traffic and attacks.
Accuracy = (TP + TN) / (TP + TN + FP + FN)

(b). Precision: Precision is the proportion of instances predicted as attacks that are actually attacks. Higher precision generally means fewer false alarms among predicted attacks.
Precision = TP / (TP + FP)

(c). Recall (Sensitivity or True-Positive Rate): Recall measures the proportion of actual attacks correctly detected by the classifier.
Recall = TP / (TP + FN)

(d). F1-Score: The F1-score is the harmonic mean of precision and recall. It provides a single measure that balances the two metrics.
F1-score = 2 × (Precision × Recall) / (Precision + Recall)

(e). False Positive Rate (FPR): The false positive rate is the proportion of normal instances incorrectly classified as attacks. A high rate can lead to unnecessary alerts or blocking.
FPR = FP / (FP + TN)

(f). False Negative Rate (FNR): The false negative rate is the proportion of actual attacks incorrectly classified as normal traffic. These are missed attacks.
FNR = FN / (FN + TP)

Experimental Environment

The experiments were carried out using WEKA on a Windows 10 Pro 64-bit computer described in the dissertation as having an Intel Core i5-4600U processor operating at approximately 2.10–2.70 GHz and 4 GB RAM. This hardware context matters when interpreting the training effort, particularly for MLP. The dissertation notes that the first MLP training run took approximately 11 hours, after which parameters were adjusted and the final reported run was used for comparison.

IV. RESULTS AND DISCUSSION

The main reported results are in Table 1. Random Forest achieved the highest accuracy at 99.9125%, followed by J48 at 99.8816%. Logistic Regression and MLP achieved 99.4853%

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

and 99.4647%, respectively. Naïve Bayes achieved 95.9440%, substantially below the other four classifiers. Random Forest also recorded the lowest reported false-positive count, while Naïve Bayes generated by far the largest number of false positives.

Table 1. Comparative Performance of the Five Classifiers

Classifier	Accuracy (%)	Precision	Recall / TP rate	TP	FP	TN	FN
Random Forest	99.9125	0.9998	0.9992	16,865	3	2,546	14
J48 Decision Tree	99.8816	0.9996	0.9991	16,863	7	2,542	16
Logistic Regression	99.4853	0.9979	0.9962	16,815	36	2,513	64
Multilayer Perceptron	99.4647	0.9983	0.9956	16,804	29	2,520	75
Naïve Bayes	95.9440	0.9603	0.9944	16,784	693	1,856	95

Note: Values are transcribed from the source dissertation's results table. TP = true positive; FP = false positive; TN = true negative; FN = false negative. The reported confusion-matrix totals and class support should be rechecked against the original WEKA output before publication. Random Forest's 99.9125% accuracy and three false positives represented the overall result in the reported comparison. Its 14 false negatives indicate that some attack instances were still missed. J48 was close in accuracy, at 99.8816%, and had seven false positives and 16 false negatives. Although it did not lead the accuracy ranking, its tree structure can offer a practical advantage when analysts need interpretable decision rules. Logistic Regression recorded 99.4853% accuracy, precision of 0.9979 and recall of 0.9962. Its 36 false positives and 64 false negatives were higher than those of the two tree-based models, but its overall results remained strong. The compact form of a logistic model may be useful in environments where model size and inference simplicity are priorities.

MLP recorded 99.4647% accuracy, precision of 0.9983 and recall of 0.9956, with 29 false positives and 75 false negatives. Its performance was close to Logistic Regression in accuracy but required substantial training time during the initial run. This demonstrates that a classifier's suitability depends on operational requirements as well as predictive performance.

Naïve Bayes recorded the lowest accuracy, 95.9440%, with 693 false positives and 95 false negatives. The false-positive count was notably larger than that of the other classifiers. This could create a greater burden for security operators and could lead to legitimate traffic being flagged. The result does not mean Naïve Bayes is universally unsuitable; it means that, under this dataset and configuration, it was the weakest of the five methods by the reported measures.

The dissertation's precision plot reported 0.9996 for J48, 0.9998 for Random Forest, 0.9979 for Logistic Regression, 0.9603 for Naïve Bayes and 0.9983 for MLP. The accuracy plot gave the same ranking described above. The recall plot showed high values across models, but the source does not provide a complete numeric recall series in the narrative beyond the tabulated values. The discussion therefore relies on the numeric values explicitly reported in Table 1.

Discussion of Findings

The results indicate that the five classifiers did not perform equally on the prepared HTTP DDoS dataset. Random Forest ranked first in reported accuracy and produced the fewest false positives. J48 was a close second, suggesting that a single decision tree was able to separate the classes effectively under the selected features and experimental settings. The small reported

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

difference in accuracy between Random Forest and J48 should not be interpreted as a statistically significant difference because the dissertation does not report a statistical significance test or confidence intervals. Random Forest's performance is consistent with the rationale for ensemble learning. A collection of trees can capture complex relationships among flow duration, packet rates, packet sizes and timing statistics. Aggregating their predictions may reduce the instability of a single tree. In this experiment, the low number of false positives is operationally important because false alerts can overwhelm security staff and may lead to legitimate users being blocked. Nevertheless, the 14 false negatives show that even the best-performing model did not identify every attack instance.

J48's near-equivalent accuracy is notable because decision trees can be inspected more readily than an ensemble of many trees. For an organization that requires transparent rules or an explanation of classification decisions, the interpretability of J48 may offset a small difference in aggregate accuracy. A practical deployment decision should therefore consider the cost of errors, the need for auditability, model update procedures and available computing resources.

Logistic Regression achieved high precision and recall, supporting its role as a baseline classifier. Its lower accuracy relative to Random Forest and J48 suggests that the traffic patterns in the dataset may include interactions or non-linear boundaries that are more naturally captured by tree-based approaches. This interpretation is plausible but not proven by the reported experiment; feature attribution or further controlled tests would be needed to establish which characteristics explain the ranking.

MLP also performed strongly, but its training process was more demanding in the reported environment. The approximately 11-hour initial run, followed by parameter adjustment, illustrates that neural-network performance can depend on configuration and that training cost should be recorded. The source does not provide a separately measured inference latency or memory footprint, so it is not possible to conclude from this experiment that MLP would be slower during real-time prediction. Training time and inference time are distinct operational considerations. Naïve Bayes had the weakest reported accuracy and produced 693 false positives. The classifier's conditional-independence assumption may be poorly matched to correlated flow attributes, and the imbalance between attack and normal records may also affect its behaviour. These are possible explanations, not demonstrated causes. A further analysis of per-class confusion matrices, feature dependencies and probability calibration would be needed to determine why its false-positive count was high.

The results also highlight the importance of reporting more than accuracy. The working dataset was imbalanced, with attack records comprising approximately 86.9% of instances. In an imbalanced dataset, accuracy can be dominated by the majority class. Precision and recall provide complementary information, while confusion-matrix counts show the practical number of errors. The source dissertation did not provide a complete numeric F1-score table, despite naming F1-score among the intended evaluation measures. Future reporting should include F1-score, specificity, false-positive rate, false-negative rate, balanced accuracy and per-class support.

A further methodological consideration concerns the exact evaluation procedure. The dissertation reports a 65:35 training-test split and also mentions 10-fold cross-validation. It does not fully clarify whether cross-validation was confined to training data or whether the final reported scores came from a separate held-out test set. The very high accuracy values make clear documentation of data leakage controls particularly important. In addition to removing the label-related field and record identifier, all preprocessing operations that learn from data should be fitted only on the training partition and then applied to the test partition. If related flows from the same capture or attack episode appear in both partitions, results may overestimate generalization to genuinely new traffic. The comparison is aligned with the broader literature

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

reviewed in the dissertation, which reports strong performance from Random Forest and J48 in several DDoS or web-attack classification studies. However, numerical comparisons across studies should remain cautious because datasets, traffic types and evaluation methods differ. The principal contribution is the direct comparison of five algorithm families under one reported experimental workflow, not a claim that the winning algorithm will dominate in every network or attack scenario.

V. CONCLUSION

This study compared Random Forest, J48 Decision Tree, Logistic Regression, Naïve Bayes and Multilayer Perceptron for HTTP-based DDoS detection using a prepared CICDDoS2019 dataset and the WEKA environment. The results show that supervised machine learning can classify the labelled traffic with high reported accuracy under the experiment's conditions. Random Forest achieved the highest reported accuracy (99.9125%) and the fewest false positives (3), while J48 followed closely with 99.8816% accuracy. Logistic Regression and MLP also performed strongly, whereas Naïve Bayes produced the lowest accuracy (95.9440%) and the highest false-positive count (693). The first objective is to establish a comparative framework was addressed through dataset preparation, feature processing, model training and common performance reporting. The second objective is to compare the selected classifiers was addressed by comparing the reported accuracy, precision, recall and confusion-matrix values. On the evidence available, Random Forest is the strongest overall candidate for the specific experimental setting, while J48 is a credible alternative where interpretability is important. These findings should be interpreted within the limits of the dataset and experimental design. They do not establish that the same ranking or performance will hold on live production traffic, unseen attack families or different network environments. The ambiguity in the reported relationship between the holdout split and cross-validation, the absence of a complete F1-score table, and the lack of statistical significance testing should be resolved before the findings are treated as definitive evidence for deployment. Based on the findings and limitations, the following recommendations are made:

(a). Random Forest should be considered as the first candidate for further validation in a controlled HTTP DDoS detection prototype because it recorded the highest reported accuracy and the fewest false positives in this experiment.

(b). J48 should also be evaluated where explainability, auditability and understandable decision rules are important. Its reported accuracy was close to Random Forest.

(c). Before deployment, both models should be tested on independent, more recent and operationally realistic network traffic. Evaluation should include different attack patterns, traffic loads, changes in user behaviour and normal peak-demand periods.

REFERENCES

Al-Khazraji, et al. (2024). Study comparing Random Forest, Logistic Regression, Naïve Bayes, J48 and XGBoost on CICDDoS2019. Full bibliographic details should be verified from the original publication.

Ebrahem, et al. (2025). Study of machine-learning algorithms for IoT-based DDoS detection. Full bibliographic details should be verified from the original publication.

Handojo, et al. (2026). Comparative evaluation of deep-learning architectures for DDoS detection using CICDDoS2019. Full bibliographic details should be verified from the original publication.

Janivasya, et al. (2024). Study evaluating J48 and other classifiers for DDoS traffic detection. Full bibliographic details should be verified from the original publication.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

Kar Suvra. (2025). Comparative study of machine-learning classifiers on CICDDoS2019. Full bibliographic details should be verified from the original publication.

Qassim, Al-Maamary, & Alwan. (2023). Study of J48, Naïve Bayes and One Rule for HTTP web-application attack detection using CSIC 2010. Full bibliographic details should be verified from the original publication.

Ugbolu, Adegboyega, & Osewa. (2023). Comparative study of classifiers for web-service attack detection. Full bibliographic details should be verified from the original publication.

Wang, Xu, & Li. (2024). Study of an optimized Multilayer Perceptron for application-layer HTTP DDoS defence. Full bibliographic details should be verified from the original publication.

Canadian Institute for Cybersecurity, University of New Brunswick. (2019). CICDDoS2019 dataset. Verify the official dataset record and citation details before submission.

WEKA. (n.d.). WEKA data-mining software. Verify the software version and official citation used in the experiment.

Adebayo, A., Chen, Y., & Kumar, R. (2025). DDoS attack mitigation systems in network environments. Journal of Cybersecurity Systems.

Al-Khazraji, M., Al-Mashhadani, H., & Al-Dulaimi, S. (2024). Performance evaluation of machine learning models on CICDDoS2019 dataset. International Journal of Network Security.

Alshamrani, A., et al. (2025). Multi-vector DDoS attacks in cloud environments. Cloud Security Review Journal.

Bace, R., & Mell, P. (2024). Flooding-based denial of service attacks and mitigation techniques. NIST Special Publication on Cybersecurity.

Bansal, P., et al. (2025). Automated architectures for modern DDoS attacks. Journal of Network Threat Intelligence.

Berbiche, L., & El-Alami, M. (2024). Limitations of anomaly-based intrusion detection systems. Cybersecurity Advances Journal.

Boukraa, S., et al. (2025). Comparative evaluation of machine learning algorithms for DDoS detection in SDN. Journal of Cloud Computing Security.

Chen, Y., & Kumar, R. (2026). Network-based intrusion detection architectures. International Journal of Cyber Defense Systems.

Ebrahem, A., Mohammed, S., & Aziz, T. (2025). Machine learning techniques for IoT-based DDoS detection. IEEE Access in Cybersecurity.

Gibson, D., & Ifenthaler, D. (2024). Pattern recognition theory in machine learning systems. Artificial Intelligence Review.

Hameed, A. (2024). Logistic regression for HTTP DDoS detection. Journal of Computer Networks.

Handojo, A., et al. (2026). Deep learning-based DDoS detection using CICDDoS2019 dataset. International Journal of Intelligent Systems.

Horak, R. (2026). Distributed denial of service attack mechanisms. Network Security Handbook.

Janivasya, S., Suryawanshi, R., & Sharma, P. (2024). Machine learning approaches for DDoS detection. Cybersecurity Research Journal.

Kaur, P., & Sharma, R. (2026). Machine learning-based intrusion detection systems. International Journal of Data Science.

Kar Suvra, S. (2025). Comparative study of ensemble learning for DDoS detection. Journal of Information Security Analytics.

Kim, J., & Kang, H. (2024). Statistical learning theory in predictive modeling. Machine Learning Review Journal.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492

Kumar, S., & Singh, R. (2025). IoT-based botnet DDoS architecture. Journal of Network Systems Security.
Li, X., et al. (2024). Naïve Bayes for large-scale traffic classification. International Journal of Computer Applications.
Mohammed, A., & Aziz, T. (2025). Random forest for IoT DDoS detection. IEEE Cybersecurity Letters.
Odukoya, F., et al. (2025). Comparative analysis of machine learning algorithms for DDoS detection. International Journal of Network Security.
Ongshu, A. (2025). Performance evaluation of probabilistic classifiers. Journal of Artificial Intelligence Research.
Pavlovich, M. (2026). Multi-vector distributed denial of service attacks. Cyber Warfare Studies Journal.
Qassim, M., Al-Maamary, E., & Alwan, H. (2023). J48 decision tree for HTTP attack detection. Journal of Web Security.
Rahman, M. (2025). Logistic regression performance in DDoS detection. Journal of Cyber Defense Analytics.
Rathore, S., et al. (2026). IoT-based botnets in modern DDoS attacks. Internet Security Journal.
Scikit-learn Developers. (2025). Machine learning in Python: Model development and evaluation documentation. https://scikit-learn.org
Singh, R., et al. (2025). Internet traffic behavior in DDoS attacks. Journal of Network Security Systems.
Thalapala, K. (2025). Machine learning comparison for DDoS detection. International Journal of Cyber Research.
Ugbolu, O., Adegboyega, A., & Osewa, O. (2023). Web service attack detection using machine learning models. Nigerian Journal of Computer Science.
Wang, Y., Xu, Z., & Li, H. (2024). Optimized multilayer perceptron for application-layer DDoS detection. IEEE Transactions on Network Security.
Xu, L. (2025). Deep learning models for HTTP DDoS detection. Journal of Artificial Intelligence in Security.
Zhang, T., et al. (2026). Decentralized botnet architecture for DDoS attacks. Future Internet Security Journal.

Volume 2, Issue 3, 2026
https://doi.org/10.5281/zenodo.23669492