IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131106

Network Intrusion Detection System with Machine Learning

S Varshini[1], Sanvi D Kenjige[2], Vishrutha MS[3], Mrs Sindhu L[4]

Don Bosco Institute of Technology, Visvesvaraya Technological University, Bengaluru, India[1-3]

Assistant Professor, Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru, India[4]

Abstract: Network intrusion detection requires methods that can distinguish legitimate network activity from traffic associated with different types of attacks. This paper presents a machinelearning-based Network Intrusion Detection System (NIDS ML) developed using the CICIDS2017 dataset. The implemented workflow includes data preprocessing, feature selection, model comparison, multiclass classification, model explainability, and a Streamlit-based interface for examining prediction results. Random Forest and Gradient Boosting were evaluated as baseline models, followed by four XGBoost configurations: Baseline, Shallow, Deep, and Regularized. From 70 usable input features, 47 features were selected for the final model. Regularized XGBoost was chosen because it achieved the highest macro F1-score among the evaluated XGBoost configurations, although the Deep configuration achieved slightly higher accuracy on the validation set. On the test set, the selected model achieved 99.87% accuracy, 95.27% macro precision, 92.39% macro recall, and 92.27% macro F1-score. SHapley Additive exPlanations (SHAP) was used to examine global feature importance and contributions to individual predictions. The Streamlit dashboard supports cleaned CSV upload, prediction summaries, flow-level investigation, SHAP explanations, and result export. The results demonstrate strong performance on the evaluated dataset, while class imbalance and the limited number of Infiltration test examples remain important limitations.

Keywords: Network Intrusion Detection, Machine Learning, CICIDS2017, XGBoost, Feature Selection, SHAP, Explainable Artificial Intelligence, Streamlit.

I.INTRODUCTION

Computer networks enable communication and information exchange across organisations, institutions, and digital services. Their widespread use also creates opportunities for malicious activities, including denial-of-service attacks, bruteforce attempts, port scanning, botnet activity, and web-based attacks. These activities can affect network availability and compromise the security of connected systems. Detecting suspicious traffic is therefore an important part of network security.

Traditional intrusion detection methods commonly rely on predefined rules, signatures, or manually configured conditions. Such methods can be effective for recognising known patterns, but their effectiveness depends on the rules available and the characteristics of the traffic being examined. Machine-learning methods provide an alternative by learning relationships between network-flow features and labelled traffic categories. Their performance, however, depends on the quality of the data, the distribution of classes, the selected features, and the evaluation procedure.

The CICIDS2017 dataset provides labelled networkflow records containing benign traffic and multiple attack scenarios, making it suitable for supervised intrusion-detection experiments [1]. However, network traffic datasets may contain substantial class imbalance. When benign traffic dominates the dataset, a model can achieve high overall accuracy while performing less consistently on less frequent attack categories. Evaluating macro-averaged metrics alongside accuracy helps reveal this difference.

This project develops a Network Intrusion Detection System with Machine Learning (NIDS ML) using CICIDS2017. The workflow includes data preprocessing, feature selection, baseline model evaluation, comparison of XGBoost configurations, test-set evaluation, and SHAPbased interpretation. The final model classifies network flows into seven consolidated categories: BENIGN, Botnet, Brute Force, DoS/DDoS, Infiltration, PortScan, and Web Attack.

The project also provides a Streamlit dashboard through which users can upload a cleaned CSV file, inspect predicted traffic categories and confidence scores, investigate individual flows, view feature contributions, and export prediction

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131106

results. The current implementation analyses uploaded data rather than capturing packets directly from a live network interface.

The main contributions of this work are the comparison of multiple tree-based classification configurations, the selection of a final model using macro F1-score as an important evaluation criterion, the use of SHAP to interpret predictions, and the integration of these capabilities into an interactive analysis interface. The results are interpreted within the limits of the dataset and experimental setup rather than as evidence of guaranteed performance on unseen networks.

II. LITERATURE SURVEY

In 2018, Sharafaldin, Lashkari, and Ghorbani presented the CICIDS2017 dataset and described its generation process and network traffic characteristics. The dataset contains labelled benign traffic and multiple attack scenarios, providing a basis for evaluating machine-learning approaches to network intrusion detection. The present project uses CICIDS2017 to train and evaluate a multiclass network-flow classifier [1].

In 2016, Chen and Guestrin introduced XGBoost, a scalable tree-boosting system designed to improve the efficiency of boosted-tree learning. Their work established a practical approach to structured-data classification. In this project, XGBoost is evaluated using four configurations to compare classification performance and select a final model [2].

In 2017, Lundberg and Lee proposed SHAP, a unified framework for interpreting machinelearning predictions. The method assigns contribution values to input features, supporting the interpretation of individual predictions and broader model behaviour. In NIDS ML, SHAP is used to examine influential network-flow features and explain the predictions produced by the selected classifier [3].

In 2016, Buczak and Guven reviewed machinelearning and data-mining methods applied to cybersecurity, including intrusion detection. Their survey discussed data-driven approaches for identifying patterns associated with cyber threats. This work provides broader context for applying supervised learning to labelled network traffic [4].

In 2019, Khraisat and colleagues reviewed intrusion-detection systems, including their techniques, datasets, and evaluation challenges. Their study discussed issues that affect the assessment of detection systems, including false alarms and changing attack patterns. These considerations are relevant when evaluating a classifier across multiple traffic categories rather than relying only on overall accuracy [5].

In 2019, Ring and colleagues surveyed datasets used in network-based intrusion-detection research. Their work examined dataset characteristics and the challenges involved in selecting appropriate datasets for evaluation. This highlights the importance of interpreting NIDS ML results in the context of CICIDS2017 and recognising that performance on one dataset does not establish generalisation to every network environment [6].

Table 1: Comparative Analysis of Reviewed Network Intrusion Detection Research

S.No.	Research paper	Technique / methodology	Main focus	Relevance
1	Sharafaldin et al. (2018) [1]	CICIDS2017 dataset generation and traffic characterization	Labelled networkflow data	Dataset foundation
2	Chen and Guestrin (2016) [3]	XGBoost tree boosting	Scalable tree learning	Selected classifier
3	Lundberg and Lee (2017) [2]	SHAP explainability	Feature contributions	Prediction interpretation
4	Buczak and Guven (2016) [4]	ML and survey data-mining	Cybersecurity analytics	Research background
5	Khraisat et al. (2019) [5]	Intrusion detection survey	Detection challenges	Evaluation context
6	Ring et al. (2019)[6]	Network IDS dataset survey	Dataset properties	Dataset considerations

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131106

The reviewed studies provide a foundation for dataset selection, tree-based classification, and explainable machine learning. Since these publications use different datasets, methods, and evaluation protocols, their results should not be treated as directly comparable performance measurements for the implemented NIDS ML system.

Table 2: Core Modules of NIDS ML, Techniques and Parameters Used

Module	Technique / parameters	Purpose
Dataset ingestion	CICIDS2017 labelled CSV records	Provides network-flow data
Preprocessing	Data cleaning and handling missing/non-finite values	Prepares consistent inputs
Feature optimization	70 usable features reduced to 47 selected features	Reduces input feature set
Baseline models	Random Forest and Gradient Boosting	Provides baseline comparisons
XGBoost configurations	Baseline, Shallow, Deep, Regularized	Compares model settings
Final classification	Regularized XGBoost; seven consolidated classes	Predicts network-flow category
Evaluation	Accuracy, macro precision, macro recall, macro F1, weighted F1	Measures performance
Explainability	Global and selected-flow SHAP analysis	Explains influential features
Dashboard	Streamlit CSV analysis interface	Displays results and flow investigation
Results export	Export of prediction results	Supports further review

III. FINDINGS

The experiments evaluated tree-based classification methods for assigning labelled network-flow records to consolidated traffic categories. Random Forest and Gradient Boosting were used as initial baselines, followed by four XGBoost configurations: Baseline, Shallow, Deep, and Regularized. The evaluation considered accuracy, macro precision, macro recall, macro F1-score, and weighted F1-score.

The results show that high overall accuracy does not necessarily indicate equally strong performance across all classes. The final model was selected using the recorded validation results, with particular attention to macro F1-score. The following subsections describe the system workflow, data preparation, model selection, prediction interpretation, dashboard functionality, and limitations.

A. System Architecture

NIDS ML follows a CSV-based processing and prediction workflow. The user begins by uploading a cleaned network-flow CSV file through the Streamlit dashboard. The application prepares the input data using the saved preprocessing artifacts and aligns the available feature columns with those expected by the trained model. The resulting feature data is passed to the selected XGBoost model for classification.

The model produces a predicted class for each processed flow, together with prediction probabilities used to display confidence information in the dashboard. The application summarises the analysed records and their predicted class distribution. Users can inspect prediction results, select an individual flow for further investigation, and view SHAP-based feature contributions associated with the selected prediction. The resulting analysis can be exported for further review.

The implemented workflow therefore connects data preparation, feature selection, classification, prediction review, explainability, and result export. The application operates on uploaded CSV data and does not directly capture live packets from a network interface.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131106

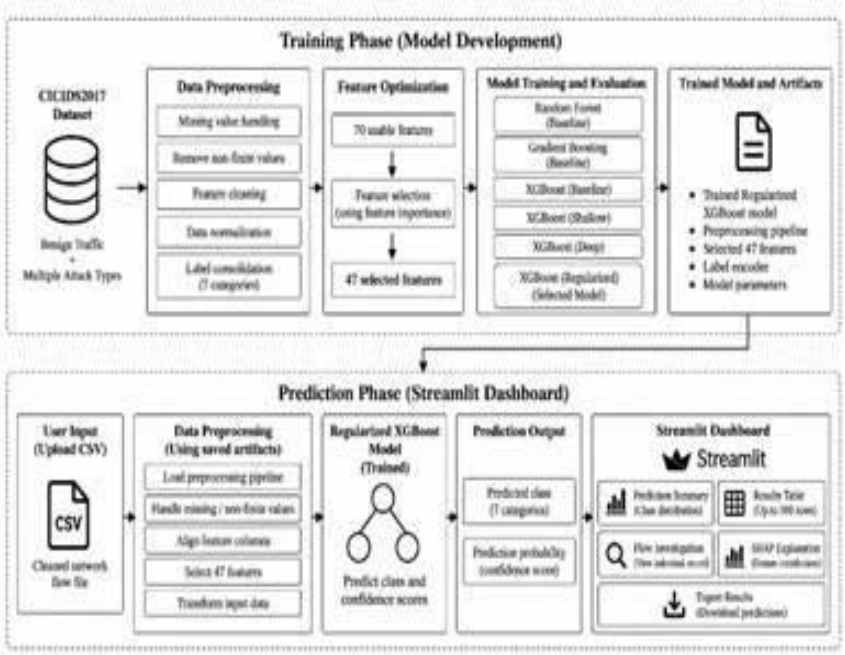


Fig. 1. Architecture and processing pipeline of the NIDS ML system.

B. Dataset Preprocessing and Feature Optimization

The project uses CICIDS2017, which contains labelled network-flow records representing benign activity and multiple attack scenarios [1]. Before classification, the data must be prepared so that the features supplied to the model are consistent with the expected input format.

The preprocessing workflow handles data-quality issues, including missing or non-finite values, and prepares the input columns required by the trained model. The feature-optimization stage reduces the usable feature set from 70 features to 47 selected features. This reduces the number of inputs supplied to the final classifier.

The saved preprocessing artifacts are reused by the dashboard so that uploaded records can be processed consistently with the trained model. Correct feature alignment is important because a mismatch between the training inputs and uploaded data could affect the validity of the predictions. The model subsequently assigns each processed flow to one of seven consolidated classes: BENIGN, Botnet, Brute Force, DoS/DDoS, Infiltration, PortScan, and Web Attack. These categories represent the consolidated labels used by the implementation rather than every individual attack label in the original dataset.

C. Model Development and Selection

The model-development process began with Random Forest and Gradient Boosting as baseline approaches. Four XGBoost configurations— Baseline, Shallow, Deep, and Regularized—were then evaluated using the validation set. The comparison included accuracy, macro precision, macro recall, macro F1-score, and weighted F1score.

The Deep configuration achieved the highest validation accuracy among the four XGBoost configurations. However, Regularized XGBoost achieved the highest validation macro F1-score and was selected as the final model. This choice reflects the project's emphasis on performance across classes rather than relying exclusively on the overall proportion of correctly classified records.

Macro F1 averages the class-level F1-scores equally, whereas weighted F1 accounts for the number of examples in each class. These metrics can therefore provide different views of model performance when the class distribution is imbalanced. Reporting both helps distinguish overall performance from the model's performance across the consolidated traffic categories.

On the test set, the selected Regularized XGBoost model achieved 99.87% accuracy, 95.27% macro precision, 92.39% macro recall, 92.27% macro F1score, and 99.87% weighted F1-score. These results describe performance on the evaluated test data and do not, by themselves, establish equivalent performance on other datasets or live network traffic.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131106

Table 3: Performance of XGBoost

Model	Features	Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted F1
XGBoost Regularized	47	99.86 84%	95.11 49%	93.76 57%	93.36 35%	99.87 16%
XGBoost Deep	47	99.89 59%	94.97 34%	93.73 48%	93.2 6 94%	99.89 92%
XGBoost Baseline	47	99.88 77%	94.75 28%	93.82 70%	93.1 3 13%	99.89 15%
XGBoost Shallow	47	99.87 79%	94.62 07%	93.68 69%	92.9 8 74%	99.88 19%

Configurations on the Validation Set

Table 4: Final Model Performance on the Test Set

Metric	Test result
Accuracy	99.87%
Macro Precision	95.27%
Macro Recall	92.39%
Macro F1-score	92.27%
Weighted F1-score	99.87%



Fig. 2. Confusion matrix of the Regularized XGBoost model on the test set.

The difference between macro F1 and weighted F1 reflects the effect of class imbalance. Benign traffic accounts for a large proportion of the dataset, so weighted metrics are strongly influenced by performance on that class. Macro F1 gives less frequent classes equal weight in the final average. The Infiltration class has only six test examples and approximately 50% recall, making this class-level estimate unstable and limiting conclusions about its general performance.

D. SHAP-Based Explainability

A classification result indicates the category predicted by a model, but it does not directly show which input features influenced that prediction. SHAP provides a method for examining feature contributions relative to the model output [2]. NIDS ML uses SHAP for global feature-importance analysis and for investigating selected network-flow predictions. In the analysed global SHAP results, Destination Port appeared as the most influential feature, followed by Init_Win_bytes_backward, Init_Win_bytes_forward, Packet Length Mean, and Packet Length Variance.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131106

Other influential features included min_seg_size_forward, Bwd Packet Length Max, Fwd IAT Min, and Fwd Packet Length Max. These findings describe the influence of features on the model's predictions for the analysed data. The dashboard also presents feature contributions for an individual flow. This allows users to inspect which features contributed towards or away from the selected class prediction. The interpretation depends on the model output and the class being explained. A feature's contribution should not automatically be treated as proof that a flow is malicious, because SHAP explains the model's behaviour rather than establishing a causal relationship between a feature and an attack.

E. Interactive Dashboard and Flow Investigation

The Streamlit dashboard provides an interface for analysing labelled network-flow data through uploaded CSV files. After the user supplies a cleaned file, the application prepares the input and generates predictions using the trained model. A summary of the analysed flows and their predicted class distribution allows users to review the overall output. The prediction table displays the predicted category and associated confidence score. The interface displays up to 500 rows in the results table, while the analysis can process the uploaded records. The flowinvestigation module allows the user to select an individual record and examine its predicted class, confidence information, and network-flow features. For further interpretation, the dashboard provides SHAP-based feature contributions for the selected flow. The export function allows prediction results to be saved for subsequent inspection. Together, these features support the examination of model outputs without requiring the user to interact directly with the underlying model code. The dashboard is designed for analysis of uploaded CSV files. It does not continuously capture network packets or provide a live packet-capture pipeline in the current implementation.

F. Implementation Status and Limitations

The implemented system combines data preprocessing, feature optimization, baseline and XGBoost model comparison, multiclass prediction, SHAP-based explainability, and a Streamlit dashboard. The final model and associated preprocessing artifacts are saved for use by the application.The evaluated results show strong overall performance on the selected dataset. However, the class distribution affects how the metrics should be interpreted. In particular, the Infiltration class has only six test examples, and its measured recall is approximately 50%. With so few examples, this classlevel estimate is highly sensitive to individual predictions and cannot provide a stable assessment of general performance.The reported metrics are based on the evaluated CICIDS2017 data. They do not establish that the model will perform equally well on traffic collected from different networks, on previously unseen attack patterns, or on zero-day attacks. Additional evaluation on independent datasets and varied traffic conditions would be necessary to assess generalisation.The dashboard also operates on uploaded CSV files rather than capturing packets directly from a live network interface. Its confidence scores and SHAP explanations should be interpreted within the context of the input data and trained model. SHAP describes feature contributions to a prediction but does not establish causality. These limitations define the scope of the current implementation and identify areas for further development.

IV. CONCLUSION

This project developed a Network Intrusion Detection System with Machine Learning using the CICIDS2017 dataset. The implemented workflow combines data preprocessing, feature optimization, model comparison, multiclass classification, SHAP-based explainability, and a Streamlit dashboard for examining prediction results. The modelling experiments compared Random Forest and Gradient Boosting baselines with four XGBoost configurations. Regularized XGBoost was selected because it achieved the highest macro F1-score among the evaluated XGBoost configurations, despite the Deep configuration achieving slightly higher validation accuracy.

The final model uses 47 selected features from 70 usable features and classifies network flows into seven consolidated categories. On the test set, it achieved 99.87% accuracy, 95.27% macro precision, 92.39% macro recall, 92.27% macro F1-score, and 99.87% weighted F1-score. The difference between macro and weighted metrics highlights the importance of considering class imbalance alongside overall accuracy. The limited number of Infiltration test examples also restricts the conclusions that can be drawn about that class.

SHAP provides global feature-importance analysis and explanations for selected flow predictions. The Streamlit dashboard integrates CSV upload, prediction summaries, flow investigation, feature-contribution analysis, and result export in one interface. The current system is intended for analysis of uploaded networkflow data rather than live packet capture.

Future work could include evaluation on additional datasets, more extensive assessment of underrepresented attack classes, testing under different network conditions, and integration with a live packetcapture pipeline. These extensions

DOI: 10.17148/IARJSET.2026.131106

would require implementation and evaluation beyond the scope of the current system. Overall, the project demonstrates a workflow for classifying labelled network flows and inspecting the resulting predictions while recognising the limitations of the data and evaluation.

REFERENCES

[1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[3] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[4] 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, doi: 10.1109/COMST.2015.2494502.

[5] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," *Cybersecurity*, vol. 2, Art. no. 20, 2019, doi: 10.1186/s42400-019-0038-7.

[6] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A Survey of Network-Based Intrusion Detection Data Sets," *Computers & Security*, vol. 86, pp. 147–167, 2019, doi: 10.1016/j.cose.2019.06.005.

[7] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, 2021, doi: 10.1002/ett.4150.

[8] M. S. Habeeb and T. R. Babu, "Network Intrusion Detection System: A Survey on Artificial IntelligenceBased Techniques," *Expert Systems*, vol. 39, no. 9, Art. no. e13066, 2022, doi: 10.1111/exsy.13066.

[9] A. Drewek-Ossowicka, M. Pietrołaj, and J. Rumiński, "A Survey of Neural Networks Usage for Intrusion Detection Systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 497–514, 2021, doi: 10.1007/s12652-020-02014-x.

[10] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron," *IEEE Access*, vol. 12, pp. 30164–30175, 2024, doi: 10.1109/ACCESS.2024.3368377.

[11] F. M. Anis, M. Alabdullatif, S. Aljbli, and M. Hammoudeh, "A Survey on the Applications of Deep Learning in Network Intrusion Detection Systems to Enhance Network Security," *IEEE Access*, vol. 13, pp. 185357–185373, 2025, doi: 10.1109/ACCESS.2025.3624952.

[12] C. K. K. Reddy, P. A. Reddy, P. S. Reddy, M. Shuaib, S. Alam, S. Ahmad, et al., "Twined Ensemble Framework for Network Security: Integrating Random Forest, AdaBoost, and Gradient Boosting for Enhanced Intrusion Detection," *Discover Internet of Things*, vol. 5, Art. no. 107, 2025, doi: 10.1007/s43926-025-00199-1.

[13] B. Rafique, S. Kanwal, and X. Shi, "CorrelationBased Feature Selection for Efficient Intrusion Detection: Comparative Evaluation of Machine Learning Models on CICIDS-2017," *Journal of Data Science and Intelligent Systems*, vol. 4, no. 2, pp. 193– 201, 2026, doi: 10.47852/bonviewJDSIS62027552.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License