

AN AI-DRIVEN NETWORK SECURITY FRAMEWORK FOR INTELLIGENT THREAT DETECTION AND MITIGATION IN MODERN COMPUTER SYSTEMS

Gullapalli Sathar¹, Subhrajyoti Dutta², Vatsalya Bapugoud Madnur³, Dr Syed basha Shaik⁴, Dr. Rakhi Dua⁵

¹Assistant Professor, CSE-Data Science Department, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, Email Id: sathar9000@gmail.com

²Assistant Professor, Information Technology, B.Borooah College(Autonomous), Ulubari, Guwahati 781007, Assam, India, Email Id: Subhroo@gauhati.ac.in, Orcid Id: 0009-0005-4851-4419

³Teaching Assistant, CSE (Data science), Data Science, Ballari Institute of Technology and Management, Email Id: ms.vatsalyabm@gmail.com, Orcid Id: 0009-0007-9955-6787

⁴Professor, Department of Computer Science and Enginrring, KKR & KSR Institute of Technology and Sciences, Guntur, AndhraPradesh, IndiaEmail Id: sybal139@gmail.com

⁵Assistant Professor, K. R Mangalam University, Email Id: rakhi.dua@krmangalam.edu.in

ABSTRACT

Network intrusion detection research can overstate deployment readiness when preprocessing, repeated traffic patterns, or train-test overlap are not controlled. This research proposes a leakage-aware AI network security system on the basis of UNSW-NB15 benchmark and combines comparative machine learning, independent evaluation, Model-Agnostic Network security interpretation, and attack-informed mitigation suggestion. A total of 175,341 training records were used in the analysis and an 82,332-record prepared test partition (excluding 8,541 test records in which predictor vectors were duplicated in training) was used for primary evaluation. A stratified group-aware cross validation has been performed for comparing the logistic regression, decision tree, random forest and extremely randomized trees. Random forest had the highest mean cross-validated ROC-AUC (0.9908) and after tuning, it got the highest accuracy (0.9071), recall (0.9642), F1-score (0.9150), ROC-AUC (0.9820) and average precision (0.9848) on the clean test set. Permission analysis revealed contextual traffic characteristics and error analysis revealed that most attacks missed were Fuzzer traffic. A two stage response layer provides mitigative actions that are based on severity. The framework focuses on the transparent and reproducible evaluation, and the clear separation of detection evidence and unvalidated automated mitigation effectiveness.

KEYWORDS: AI-driven cybersecurity; network intrusion detection; UNSW-NB15; random forest; explainable AI; data leakage

INTRODUCTION

The number of enterprise services, cloud services, mobile endpoints, industrial devices and cyber-physical components in networking computing infrastructure continues to grow, and become more heterogeneous, further increasing the quantity and diversity of traffic to be observed. Under these conditions, IDSs need to be able to differentiate between real malicious activity and the normal variations of legitimate operations without causing an unacceptably high rate of false alarms. To help solve this issue, a UNSW-NB15 benchmark was designed based on modern normal and malicious traffic, designed flow features and various attack families, and is now a widely used dataset for data-driven network intrusion detection (NID) [1].

However, there is more to benchmark utility than just having labels. The UNSW-NB15 has been statistically analysed and found that the traffic and feature distributions significantly vary from those of the older benchmarks and it has been thus concluded that evaluation design is crucial to make any statement about detection capability [2]. The draw of machine learning is the ability of classifiers to learn multivariate decision boundaries that are hard to manually code, but that same "flexibility" implies that how data is preprocessed, how many features are redundant, how well the classes are balanced and even how you validate your protocol can all affect your results. Supervised learning, ensembles and anomaly oriented methods have been consistently highlighted as key tools in broad surveys of cyber-security analytics and the extent of reported performance is heavily dependent on the characteristics of the benchmark and experimental setup [3].

The goal of a network detector is not just to maximize overall accuracy, but to achieve some practical goal. Missed attacks, specificity for benign traffic, and an auditable decision process are required for a security system. Intrusion detection is still facing the problem of class imbalance, diverse attack behavior, shift in traffic and sensitivity and false-positive load in the contemporary review [4]. These are key factors to consider if the model is designed to be used to guide downstream mitigation, for example, in which case an overly sensitive detector may minimize the number of false-negative events but could cause disruption of operations if false-positive events automatically block or isolate the detector.

Another longstanding issue of uncertainty is design of the dataset. The studies show that there are significant differences between the benchmarks in the aspects of recording environment, labeling process, flow construction, class composition, duplicated and correlated observations on the network intrusion data sets [5]. Therefore, a high test score might be an actual discriminatory structure, regularities in a specific benchmark, or an inadvertent leakage. It is therefore essential to

include in the experimental design a description of how the train-test boundary is protected, and whether repeated predictor patterns happen to be in either benchmark partition or to cross validation folds.

The literature on network anomaly detection also emphasizes the importance of unifying the data acquisition, the feature representation, the model learning, the decision thresholds and the response in a single security pipeline rather than separate components [6]. This point of view is applicable straight to intelligent mitigation. While a traffic classifier can give the evidence that traffic is malicious, deployment decisions, including rate limiting, quarantining, connection blocking, or analyst escalation, will be based on the context of the attack, risk tolerance and operational limitations. Therefore, it is important to distinguish between detection performance and effectiveness of mitigation efforts using experimental means.

Intrusion detection research has seen a significant boost from deep learning in its ability to model complex nonlinear relationships, but as shown in comparative studies, there is no inherent trade-off of complexity to avoid having to rely on representative datasets, reproducible preprocessing and defensible validation [7]. Strong tree ensembles are scientifically relevant for tabular flow data (such as UNSW-NB15) as they capture the nonlinear interaction between features efficiently, evaluate probabilities, and can be analyzed with regard to individual features, without the burden of deep architectures.

In this context, the current study offers a leakage aware and reproducible machine learning framework for binary threat detection of networks with a response layer which can be interpreted for attack-informed mitigation. The contribution is methodological and not a new classifier; the predictor overlap for the benchmark partitions is exact, but is verified prior to independent evaluation; the predictor vectors are grouped during cross validation; the preprocessing is all learned on the training folds; the competing models are selected before the held-out test set is used; and the final detector is analysed using permutation importance and structured error analysis. The design is aimed to make more transparent and operationally interpretable, strong performance claims. In view of this, the study has the following three research objectives:

To develop a reproducible and leakage-controlled machine learning pipeline for binary detection of malicious network traffic using UNSW-NB15, including explicit inter-partition overlap auditing and group-aware validation.

To comparatively evaluate suitable machine learning classifiers, identify the strongest model using training-only evidence, and characterize its independent test performance, feature influence, and error structure.

To translate model outputs into an attack-aware mitigation decision layer that links detected threat context to severity-appropriate response recommendations while clearly delimiting the boundary between predictive validation and operational mitigation validation.

2. LITERATURE REVIEW

Initial studies of UNSW-NB15 tried to determine if the engineered characteristics of UNSW-NB15 contained information which could discriminate beyond what was provided by the legacy intrusion benchmarks. Moustafa and Slay have analyzed the important characteristics of the two datasets, UNSW-NB15 and KDD99, and have proven that certain features are not equally applicable to all datasets [8]. This observation is still relevant as it warns against claiming the results of any single feature-ranking as a cause for malicious behavior. Thus, in the current study, feature interpretation is used after model evaluation and is not causal attribution but predictive influence.

The feature-space complexity has prompted a lot of research on dimensionality-reduction. Abdulhammed et al. demonstrated strong performance for intrusion classification in high-dimensional representations of traffic, which is achieved by reducing the representations using autoencoders and principal components [9]. This shows the computational utility of representation reduction, though if the transformation is learnt from the whole dataset, it may contain information about the global distribution of data, which may lead to leakage of information during validation if it is done before partitioning the data. In the current framework, imputation, encoding and scaling are performed within the estimators that are fitted at each training fold.

In particular, Kasongo and Sun compared feature selection with various classifiers on UNSW-NB15 and found that the reduced representations can enhance efficiency and for some classifiers, predictive ability [10]. It is also evident from their class level observations that minority attack categories continue to be a challenge. This encourages two decisions in the current study: binary detector selection is not based on accuracy alone, and post hoc error analysis is used to gain insight into the false negatives of the binary detector in terms of attack categories.

Recently, deep learning has been suggested as a way of learning complex attack representations. In this way, Vinayakumar et al. have tested the DNN-based intelligent intrusion detection and found that it exhibits an excellent capability in cybersecurity datasets [11]. However, deep models have extra model design, optimization and calibration decisions. Classical and ensemble models offer a more manageable test bed for studying leakage control, transparent comparison, and reproducibility for a study with a primary contribution of leakage control and where nonlinear modeling capability is not lost.

The second strand of research is on standardisation of features between the datasets. To render the evaluation among the different datasets more meaningful, Sarhan et al. built common NetFlow representations and showed that the comparison of detectors is complicated because each benchmark shows a different feature space [12]. Their work confirms a general methodological lesson: On-dataset performance does not equal real-world generalization. The current study further enhances the independence of UNSW-NB15 by eliminating exact inter-partition predictor overlaps, but does not mean it as an alternative to cross-dataset or live-network validation.

Other methods that exploit complementary classifier behavior are based upon ensemble learning. Kabir et al. used stacking with UNSW-NB15 and demonstrated that stacking of different heterogeneous models can improve the intrusions classification [13]. This, however, does not apply to ensemble gains, which rely on the validation of base learners, and the actual error diversity of base learners. To avoid adding another layer of model complexity, the present study compares

logistic regression, a single decision tree, random forest and extremely randomized trees under the same group-aware protocol using an attempt to evaluate the gain from bagged tree ensembles.

The difference between binary and multiclass ID can also be of importance. Rathod et al. experimented with the behavior of the models on both binary and multiclass UNSW-NB15 tasks and demonstrated that the performance is dependent on formulation of the classification task and distribution of attacks [14]. This enables a staged architecture: The main experimental focus in this objective was to detect the presence of attacks, with attack-category routing being a secondary operational component that does not leak the attack label into the binary detector.

Random-forest type models have been recently evaluated in comparison with other models, with good performance on UNSW-NB15. Fathima et al. achieved competitive results when comparing the conventional machine learning methods, and highlighted the usefulness of ensemble trees in cyber-threat detection [15]. Comparing the results of such studies directly with results from this research is still challenging due to different preprocessing, split definitions, overlapping, balancing, and feature-selection decisions. Therefore, the present work does not assert superiority of unmatched protocols, and does not hesitate to make also the validation boundary part of the contribution.

Alongside this comes the requirement for interpretability because AI-based ID is advancing towards operational decision support. Human-AI Interaction (HCI) is a key factor in AI deployment, and in the case of network intrusion detection, the decisions must be transparent for understanding and trust. Arreche et al. proposed an explainable-AI (XAI) framework for network intrusion detection (NID) and argued that transparency is needed to understand and trust the decisions made by the model [16]. However, the majority of IDS research still considers explanation as an optional visualization step following the optimization of the accuracy of the benchmark. The existing research gaps tackled by this work are the combination of leakage-aware validation, model selection, model-agnostic feature influence, error analysis by attack and incorporation of a well-defined mitigation recommendation layer in a single reproducible workflow.

3. METHODOLOGY

3.1 Research Design and Dataset Configuration

The study was designed using secondary data experimental approach and the partitions were prepared from UNSW-NB15 [17]. The dataset consists of flow-level network observations that provide information about the behavior of connections, protocol/service data, packet/byte statistics, timing characteristics, state/TTL variables, and contextual counters of connections. There were 45 columns in the prepared files. For binary modelling, all the variables except the identifier and the binary target were used, resulting in a predictor matrix of 42 variables (39 numeric and three categorical: proto, service and state). The binary labelling was done as 0: normal traffic and 1: attack traffic.

The distributed archive names were used inconsistently with the conventional split sizes, so canonical benchmark row counts were used to file roles. The 175,341-record partition was used as the model-development data while the 82,332-record partition was used as the raw held-out evaluation data. These prepared partitions were checked for missing values and complete-row duplicates using data-quality validation and no such issues were found. However, the missing value handling code was kept within the modeling pipelines to make them robust and reproducible in case the same files are loaded again, but with a different pandas backend.

Modelling was based on an examination of class structure. The number of training set attack and normal records were 119,341 and 56,000, respectively. This imbalance, of around 2.13:1, was not severe and so synthetic resampling was not applied. Instead, class weighting was applied to the candidate classifiers to ensure that class balancing was done in the process of training, and hence could not affect the independent test set. The nine attack families were used only for descriptive analysis and for error characterization for the binary detector, in post hoc fashion.

3.2 Leakage Audit and Independent Evaluation Construction

The requirement for leakage control was taken as a top methodological requirement. The first safeguard was “target isolation”, attack_cat was never available for the binary detector and label was never used as a predictor, record identifier was not used for learning. The second protection needed to take care of duplicate predictor patterns. A deterministic row hash was calculated on the entire set of 42 predictors for each training and raw-test observation. If the predictor vector of any raw-test row was the same as the predictor vector of a training row, then the row was removed from the main set of rows to be evaluated. This identified 8,541 test records that overlapped and resulted in a leakage clean test cohort of 73,791 test records. The data on the evaluation side were only filtered, and no information from the test labels were used to modify the data on the model-development side.

The third safeguard took care of repeated predictor patterns within the development of the model(s). Stratified group-aware cross-validation was used in the analysis to avoid having exactly identical predictor cases in different folds, using predictor hashes for group identification. This minimizes the chance that the same observation is seen twice while training and validation a classifier. The held out clean test set was not checked for model family ranking or hyper parameter selecting. Its labels were not employed until the final model-selection (training only) was performed.

The interquartile-range rule was used descriptively to evaluate outliers for numeric training data. Clipping or winsorization was not performed, since, for example, large or small flow statistics can be a valid indication of malicious traffic and indiscriminate outlier removal would remove security-relevant information. This makes the choice preserve the empirical distribution and record variables that exhibit much tail behavior.

3.3 Preprocessing and Candidate Machine Learning Models

All preprocessing steps that can learn from data were wrapped in scikit-learn pipelines. Median imputation for numeric variables was performed. Numeric features were also standardized using training-fitted standardization (for logistic regression only). Categorical data were filled with the most common training value and one hot encoded with the unknown category flag to allow the categories to be less frequent than 5 to be grouped together by the encoder. Validation/ test observations were then refitted with the same fitted transformations.

Four classifiers were chosen to represent linear, single-tree and ensemble classifiers. Balanced class weights and liblinear solver were used in logistic regression with a maximum of 1,000 iterations. The decision tree was trained with balanced class weights, a maximum depth of 20 and a minimum leaf size of two. Random forest had 160 trees, square-root feature sampling and balanced-subsample weighting. The 180 trees, square-root feature sampling and balanced class weights were used for the extremely randomized trees. A fixed random seed of 42 was used for all the stochastic components. The model set was carefully designed to be compact so it can be used for transparent comparison and workstation level reproduction, as opposed to an indiscriminate benchmark of a multitude of algorithms.

No feature selection was done prior to cross validation. This prevented a frequently used leakiness pipeline where target-informed feature ranking was done across the entire dataset prior to the folds being generated. But rather, deferred to the end of model selection, at which point permutation importance could be computed on held-out data without changing the fitted detector.

3.4 Group-Aware Model Selection, Hyperparameter Tuning, and Evaluation Metrics

Three-fold StratifiedGroupKfold validation of a stratified subsample of the 35,000 records in the development data was used for model comparison. The exact predictor-vector hash (as outlined above) was used as the grouping variable. Mean ROC-AUC was used as the primary selection criteria, as it is a measure of ranking discrimination across thresholds, while accuracy, balanced accuracy, precision, recall, F1-score and average precision were kept as complementary selection criteria. This way, model selection is not based on any one statistic that is dependent on a threshold.

The winning model family was then tuned with RandomizedSearchCV with the same 3-fold group-aware validation on a stratified subsample of the data, containing only the training records (45,000 records). In the case of random forest, the search space included either 160 or 220 or 300 estimators, maximum depths of 10, 14 or 16, minimum leaf sizes of 1, 2 or 4, minimum split sizes of 2, 4 or 8 and square-root feature sampling. To have a practical and reproducible compute budget, three randomly sampled configurations were evaluated. All 175,341 training observations were then refitted using the cross-validation selected configuration. The workflow of execution automatically archives the parameter dictionary that has been selected with the trained model artifacts.

On the leakage-clean test set, there are 73,791 records, which were used to quantify independent performance. The metrics included were accuracy – overall correctness, balanced accuracy – class-wise recall, precision – proportion of the predicted attacks that were indeed malicious, recall/sensitivity – proportion of attacks which were detected, specificity – proportion of benign-traffic which were detected, F1-score – that summarized the precision-recall balance, ROC-AUC – threshold independent discrimination, and Matthews correlation coefficient – that provided a balanced correlation measure. For security oriented error interpretation the confusion matrix and false-negative distribution were kept.

3.5 Explainability, Mitigation Logic, Reproducibility, and Ethical Considerations

Permutation importance was used to assess model explainability with ROC-AUC being used as the scoring function. The mean decrease in ROC-AUC was computed for 3 repeats on a held-out subsample of the 3000 records, for each original predictor. Importance remained interpretable at the original level of features because categorical information was permuted prior to pipeline transformation so it remained whole and not broken up into one-hot indicators. Permutation importance was considered as model-dependent measure and not causal.

The threat-mitigation component was conceived as a decision layer that is transparent, and not an additional "experimental" claim of control effectiveness. The last binary detector is used to discern if traffic is normal or malicious. Then the traffic that is predicted as malicious is directed to an attack-category classifier to map actions for context response. When the binary detector indicates an attack, but the category router indicates Normal, it is conservatively escalated as an unspecified attack. Categories are assigned to severity levels and actions including monitoring, rate limiting, source filtering, endpoint isolation, containment or analyst escalation are recommended. None of the above actions actually occurred on live firewalls, endpoints or production networks; they are actions that constitute decision support.

All of this was done with a fixed random seed, explicit validation of the directories and files, saved environment versions, determinate split-role identification, reusable preprocessing pipelines, saved fitted model, exported metrics and predictions, and 600-DPI figures. Public network-security benchmark was used and there was no human-participant or personally identifiable data used in the experiment. Governance controls would be necessary for operational use, however, in order to avoid disruption of legitimate services from false positive results; false negative results may lead to attacks remaining uncontained; and, explanation output may be misinterpreted when used as causal evidence.

4. RESULTS

4.1 Dataset Characteristics and Class Structure

The principal dataset dimensions, leakage audit, and class composition used for modeling are summarized in **Table 1**.

Table 1. Validated dataset configuration and leakage-clean evaluation cohort

Dataset attribute	Observed value
Prepared training records	175,341
Prepared raw test records	82,332
Original columns per prepared file	45

Dataset attribute	Observed value
Binary modeling predictors	42 (39 numeric, 3 categorical)
Training normal records	56,000 (31.9%)
Training attack records	119,341 (68.1%)
Raw-test predictor-overlap records removed	8,541 (10.37% of raw test)
Leakage-clean test records	73,791
Clean-test normal records	35,528
Clean-test attack records	38,263 (51.9%)
Missing values in prepared partitions	0
Complete-row duplicates in prepared partitions	0

Table 1 shows that the leakage audit removed 8,541 records, equivalent to 10.37% of the nominal test partition. The resulting clean test set is also less attack-heavy than the training set (51.9% versus 68.1% attacks), creating a more demanding threshold-based evaluation and reducing the chance that performance is inflated by exact repeated predictor patterns. The binary class distribution of the model-development partition is introduced in **Figure 1**.

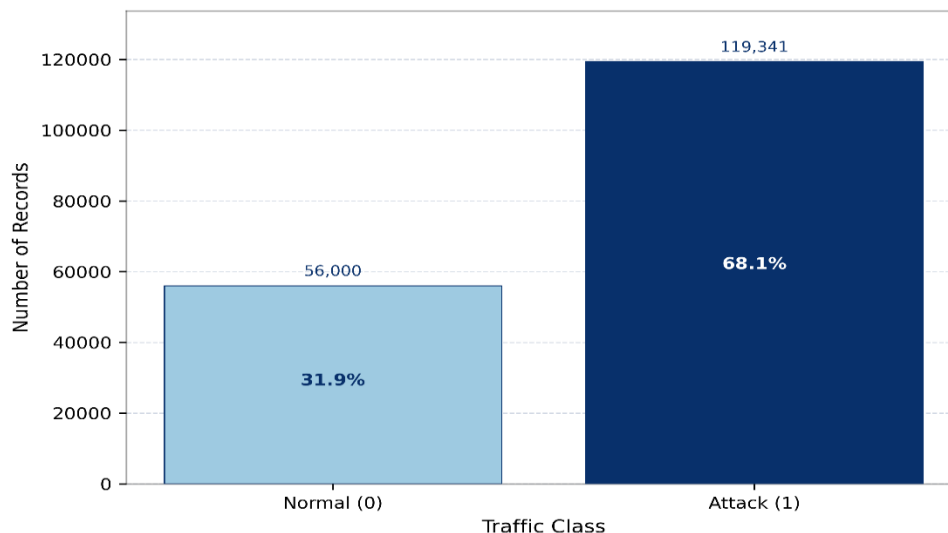


Figure 1. UNSW-NB15 training distribution: normal versus attack

Figure 1 shows that attacks constitute 68.1% of training records, compared with 31.9% normal traffic. The imbalance is meaningful but not extreme, supporting class-weighted learning without synthetic oversampling and allowing precision, recall, and balanced accuracy to remain interpretable. The composition of the training data across normal traffic and the nine UNSW-NB15 attack families is presented in Figure 2.

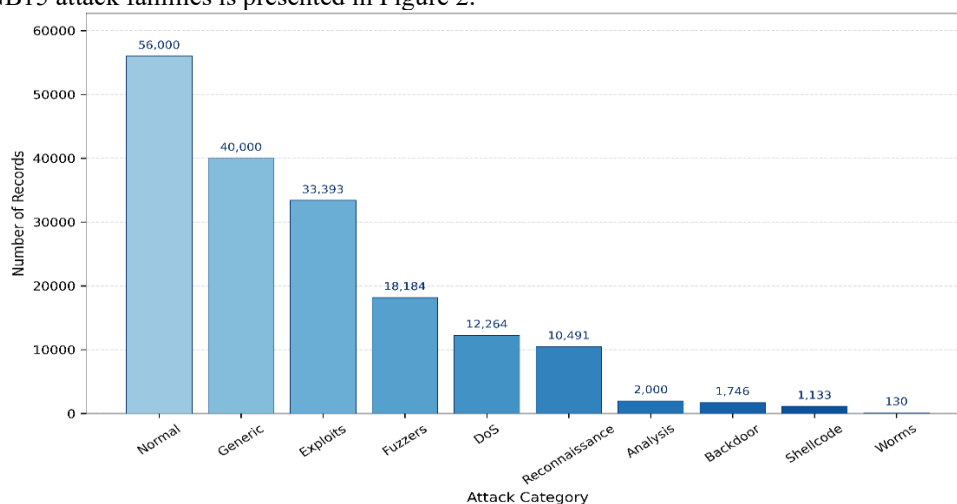


Figure 2. Training distribution by attack category

Figure 2 demonstrates pronounced heterogeneity across attack categories: Generic (40,000), Exploits (33,393), Fuzzers (18,184), DoS (12,264), and Reconnaissance (10,491) dominate the malicious observations, whereas Analysis (2,000), Backdoor (1,746), Shellcode (1,133), and especially Worms (130) are scarce. This imbalance is a major reason that the binary detector is evaluated separately from the secondary attack-routing component.

4.2 Leakage-Safe Comparative Model Performance

Training-only group-aware cross-validation results for the four candidate binary classifiers are reported in **Table 2**.

Table 2. Group-aware cross-validated performance of candidate binary threat detectors

Model	Accuracy	Precision	Recall	F1	ROC-AUC	AP
Random Forest	0.9490	0.9510	0.9752	0.9630	0.9908 +/- 0.0003	0.9954
Extra Trees	0.9476	0.9500	0.9743	0.9620	0.9901 +/- 0.0005	0.9949
Logistic Regression	0.9323	0.9483	0.9525	0.9504	0.9840 +/- 0.0004	0.9917
Decision Tree	0.9297	0.9649	0.9305	0.9474	0.9506 +/- 0.0038	0.9600

Table 2 indicates that Random Forest ranked first on the prespecified primary measure, with mean ROC-AUC 0.9908 +/- 0.0003. Extra Trees was a close second at 0.9901, while logistic regression reached 0.9840 and the single decision tree 0.9506. Random Forest also produced the highest F1-score (0.9630) and average precision (0.9954). The narrow ROC-AUC margin over Extra Trees should not be interpreted as proof of universal superiority; it identifies Random Forest as the best candidate under this specific group-aware development protocol. The relative cross-validated ROC-AUC of the candidate models is visualized in Figure 3.

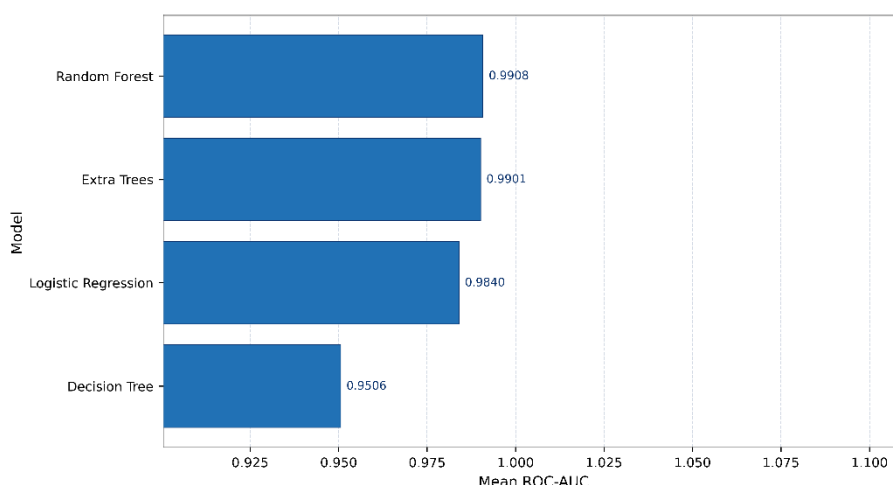


Figure 3. Group-aware cross-validated ROC-AUC by candidate model

Figure 3 reinforces the advantage of ensemble trees under the leakage-aware validation protocol. The two ensemble models cluster near ROC-AUC 0.99, logistic regression remains competitive, and the single decision tree exhibits the largest decline, suggesting that averaging across decorrelated trees improves ranking stability for the nonlinear UNSW-NB15 feature space.

4.3 Independent Test Performance of the Selected Random Forest

The final Random Forest was refitted using the complete training partition and evaluated once on the leakage-clean test cohort; its primary metrics are given in Table 3.

Table 3. Independent leakage-clean test performance of the selected Random Forest

Metric	Value
Clean test records	73,791
Accuracy	0.9071
Balanced accuracy	0.9049
Precision	0.8706
Recall / sensitivity	0.9642

Metric	Value
Specificity	0.8456
F1-score	0.9150
ROC-AUC	0.9820
Average precision	0.9848
Matthews correlation coefficient	0.8184
True positives	36,893
True negatives	30,044
False positives	5,484
False negatives	1,370

Table 3 shows a security-oriented performance profile. The detector identifies 96.42% of attacks while maintaining 84.56% specificity, producing an F1-score of 0.9150 and MCC of 0.8184. Accuracy declines from the training-only cross-validation mean to 0.9071 on the cleaned test distribution, but threshold-independent discrimination remains high (ROC-AUC 0.9820; average precision 0.9848). The difference between high ranking performance and lower threshold accuracy indicates that calibration or threshold optimization could further adapt the detector to deployment-specific false-positive costs without retraining the ranking model. The class-specific prediction counts of the final detector are presented in Figure 4.

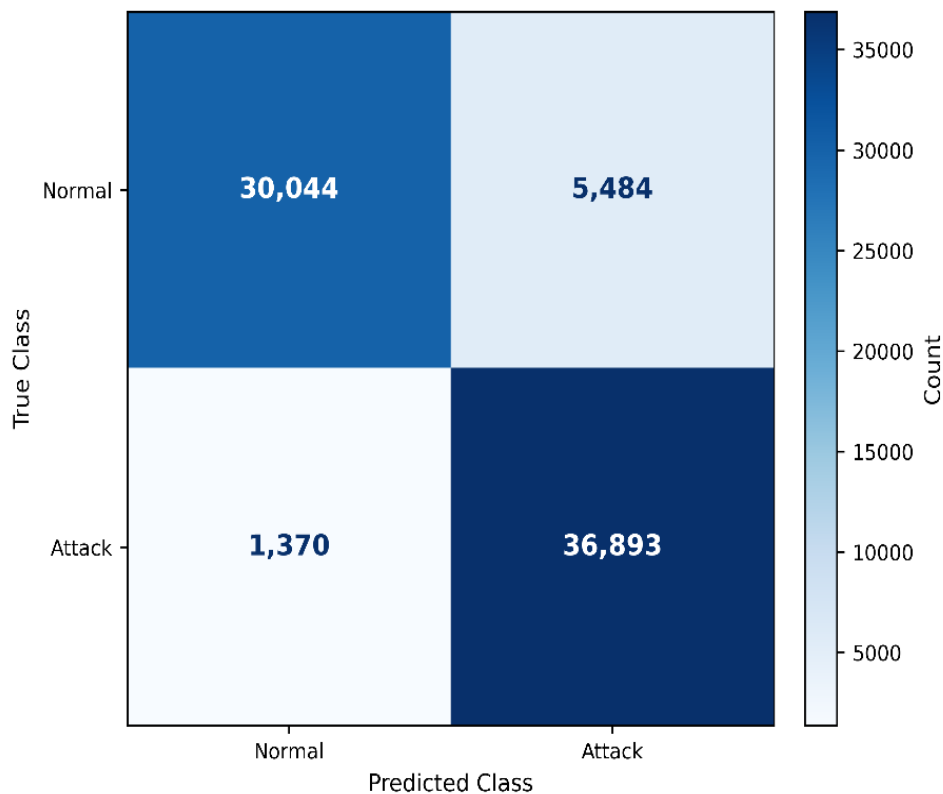


Figure 4. Confusion matrix of the final Random Forest binary threat detector

Figure 4 shows 36,893 correctly detected attacks and 30,044 correctly recognized normal records. The 1,370 missed attacks are substantially fewer than the 5,484 false alarms. In security terms, the chosen operating point favors sensitivity over benign-traffic specificity, which is reasonable for detection but requires downstream mitigation safeguards to avoid unnecessary disruption. Threshold-independent ranking behavior is introduced in Figure 5 through the receiver operating characteristic curve.

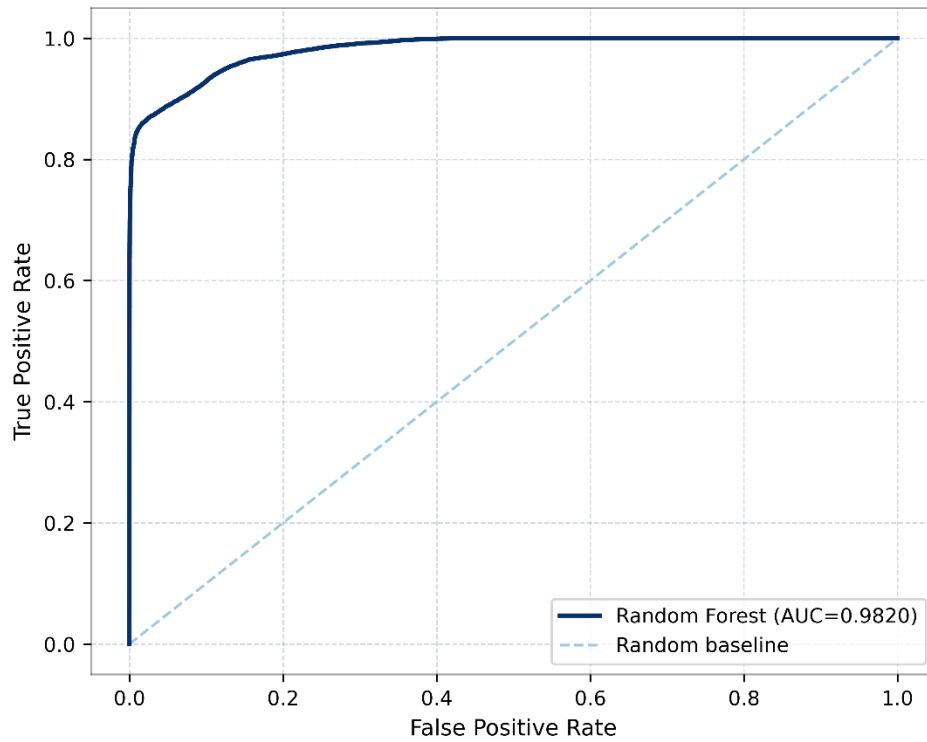


Figure 5. ROC curve of the final binary threat detector

Figure 5 yields an ROC-AUC of 0.9820, indicating strong separation of attack and normal traffic across decision thresholds. The curve rises rapidly at low false-positive rates, although the confusion matrix confirms that a fixed 0.5-style operating threshold still produces a nontrivial false-alarm burden under the cleaned test distribution. Because attack prevalence is substantial and precision is operationally important, the precision-recall behavior is additionally shown in Figure 6.

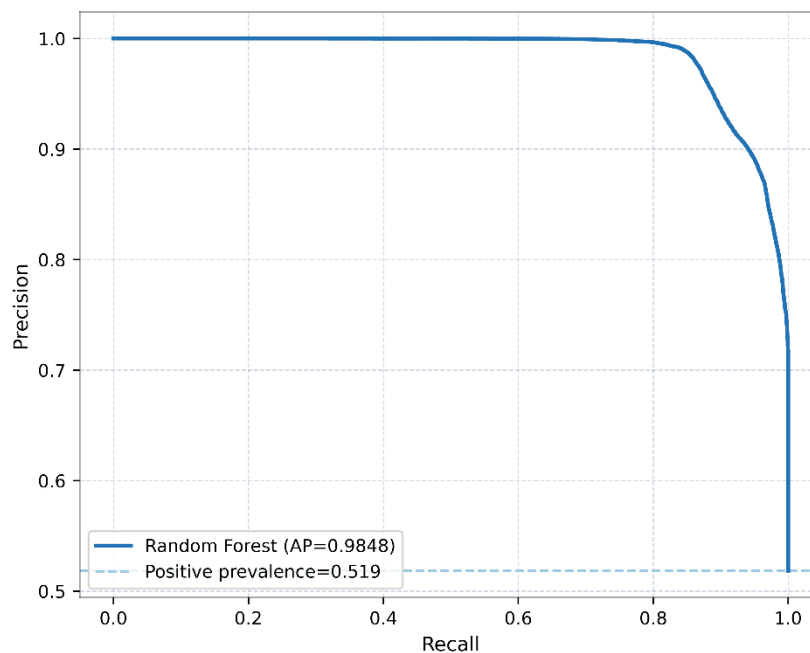


Figure 6. Precision-recall curve of the final binary threat detector

Figure 6 reports average precision of 0.9848 against a positive-class prevalence of approximately 0.519. The curve remains near unit precision across much of the recall range before declining at the most aggressive detection thresholds, supporting the conclusion that the model preserves strong attack ranking even after exact train-test predictor overlap has been removed.

4.4 Model Explainability and Feature Contributions

Model-agnostic permutation importance for the selected detector is presented in **Figure 7**.

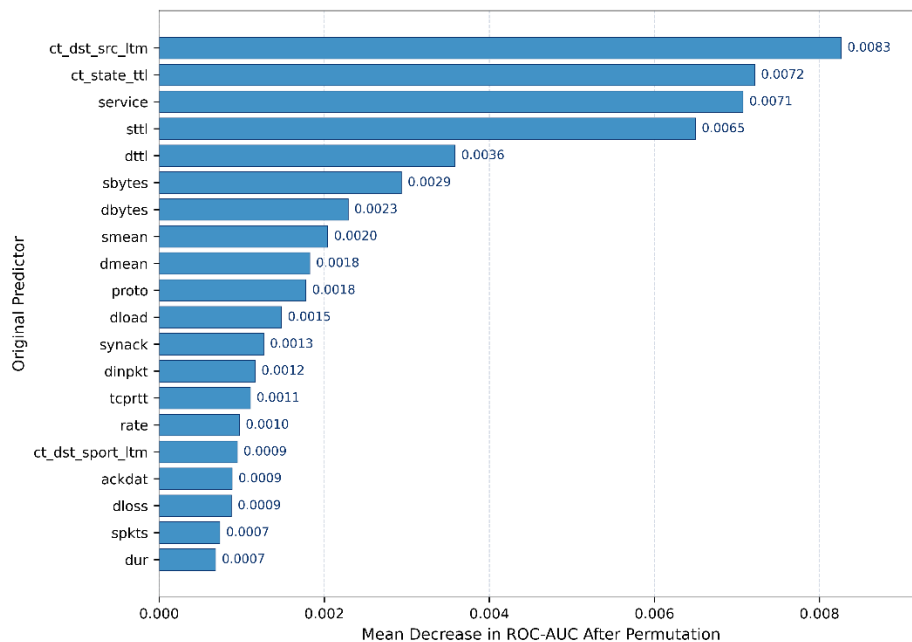


Figure 7. Permutation importance of the final Random Forest on a held-out test subsample

Figure 7 identifies `ct_dst_src_ltm` (mean ROC-AUC decrease 0.0083), `ct_state_ttl` (0.0072), `service` (0.0071), and `sttl` (0.0065) as the strongest original-variable perturbations, followed by `dttl`, `sbytes`, `dbytes`, `smean`, `dmean`, and `proto`. The importance values are individually modest, which is consistent with a distributed ensemble decision boundary and correlated network statistics: predictive information is shared across several related traffic variables. The presence of both contextual counters and protocol/service descriptors indicates that the detector does not rely solely on raw packet-volume features. These importances quantify dependence of the fitted model on each variable under permutation and should not be interpreted as causal mechanisms of attack behavior.

4.5 Error Analysis and Threat-Mitigation Decision Output

To determine which malicious behaviors remain most vulnerable to binary misclassification, false negatives were decomposed by their original attack category in **Table 4**.

Table 4. Attack-category composition of false negatives from the final binary detector

Attack category	False negatives	Share of all false negatives
Fuzzers	1,267	92.48%
Exploits	77	5.62%
Reconnaissance	11	0.80%
Shellcode	5	0.36%
Generic	4	0.29%
DoS	4	0.29%
Backdoor	2	0.15%

Table 4 shows that Fuzzers account for 1,267 of 1,370 missed attacks (92.48%), whereas Exploits account for 77 and all remaining categories contribute only small counts. The concentration of errors in one family is more informative than the aggregate false-negative rate alone: it identifies a concrete weakness for class-aware thresholding, additional representation engineering, or specialized second-stage review. The response layer associated with detected threat categories is summarized in Table 5 as an operational decision-support output rather than a claim of measured mitigation efficacy.

Table 5. Severity-aware mitigation recommendations used by the framework

Predicted context	Severity	Recommended response	Operational rationale
Normal	Low	Allow traffic and retain baseline monitoring	No attack evidence from the binary detector.

Predicted context	Severity	Recommended response	Operational rationale
Generic	High	Block or filter suspicious source; initiate review	High-volume or broadly malicious patterns warrant rapid containment.
Exploits	Critical	Isolate affected session/host, block source, trigger vulnerability review	Exploit traffic may indicate active compromise of exposed services.
Fuzzers	Medium	Rate-limit source, inspect malformed inputs, intensify protocol monitoring	Most binary false negatives occurred in this family, so additional scrutiny is justified.
DoS	Critical	Rate-limit/filter abusive traffic and activate availability safeguards	Primary risk is service degradation or denial of availability.
Reconnaissance	Medium	Restrict scanning source, increase logging, inspect follow-on attempts	Reconnaissance may precede exploitation and benefits from contextual monitoring.
Analysis	Medium	Increase monitoring, retain evidence, request analyst review	Ambiguous analytical/probing behavior should be reviewed before disruptive action.
Backdoor	Critical	Quarantine endpoint, block command-and-control paths, investigate compromise	Backdoor behavior implies possible persistent unauthorized access.
Shellcode	Critical	Block traffic, isolate endpoint, inspect process/memory and exploit chain	Shellcode activity can be associated with code execution attempts.
Worms	Critical	Isolate host or segment, block propagation paths, initiate containment	Self-propagating behavior requires rapid containment to limit spread.
Unspecified Attack	High	Restrict traffic, preserve evidence, escalate to analyst	Used when binary detection is malicious but category routing is uncertain or contradictory.

Table 5 operationalizes the title-level mitigation component without overstating experimental evidence. The binary detector gates all response actions, and category-specific recommendations add context only after malicious traffic has been identified. The scheme deliberately includes analyst escalation for ambiguous cases, reducing the risk that a single classifier output automatically triggers irreversible control actions. Real mitigation effectiveness, response latency, service impact, and recovery outcomes remain subjects for deployment-level validation.

5. DISCUSSION

The results validate the ensemble trees for the structured network-flow detection, and the validation discipline is more important. Random Forest was the model with the best group-aware development ROC-AUC (0.9908) and had good discriminative ability on a leakage-clean independent test set (ROC-AUC 0.9820; average precision 0.9848). This is in line with recent studies demonstrating that ensemble learning can be used to achieve strong network anomaly detection with a post hoc explanation capability [18]. Meanwhile, the clean-test accuracy indicator of 0.9071 is significantly lower than the development accuracy indicator of 0.9490, which provides a demonstration of why benchmark scores in the headlines should not be viewed without considering the composition of the splits, and duplicated predictor structure.

There is an obvious sensitivity-specificity compromise of the detector. With a recall of 0.9642, less than four percent of attacks were missed; with a specificity of 0.8456, there were 5,484 false positives. This profile may be favored over a detector that is primarily configured for accuracy when detecting benign traffic, but the price point may be based on whether alerts are triaged, rate-limited, or used as an automated containment trigger. In the field of explainable intrusion-detection, there have also been studies on incorporating interpretative evidence along with the predictive performance, to help security operators determine whether model output needs to be acted upon and how [19]. In the current study, two complementary interpretability models are used: permutation importance for a global description of the predictive dependence of the model, and category-level error analysis for revealing where the selected threshold breaks down.

It is important to note that the percentage of false negatives was 92.48% in Fuzzer traffic. A simple recall for the class would not reveal this weakness. It implies that the learned binary boundary, under the leakage-clean test distribution, is quite general and can detect the majority of the attack families, but has some overlap with benign behavior. This could be solved in a future system by category sensitive calibration, cost sensitive secondary classification, temporal context or specialized detectors. But complexity should only be added if there is independent evidence that it is necessary and not achieved by continually optimising on the same benchmark.

Also, the gap between cross-validation and independent test results is a further testament to worries about domain generalization. Across different datasets there are studies showing that the performance of machine-learning NIDS can degrade when the collection network, traffic composition, feature extraction, or attack environment changes [20]. Once exact predictor overlap within UNSW-NB15 is removed, internal independence is increased, however, there is no new domain created. Thus, the current ROC-AUC should be viewed as proof of good within-benchmark discrimination under a more stringent split and not as proof that the model will be able to maintain the same performance in a modern enterprise, IoT or industrial network.

Permutation importance further indicates that network context is distributed across multiple signals. `ct_dst_src_ltm`, `ct_state_ttl`, `service`, and `sttl` were the strongest perturbations, but no individual feature dominated the model. Although this is desirable from a redundancy point of view, it makes the explanation narratives with simple models difficult: importance may be spread across correlated predictors; permutation scores may vary with distribution shift. For this reason, the explanation layer is not designed to identify any feature as the cause of malicious behavior, but is best used to identify model dependencies and investigation priorities.

The mitigation element offers practical relevance and an essential scientific limit. The correspondence between threat classification and actions like rate limiting, filtering, isolation, containment and analyst escalation is a decision policy, not a cyber-defense controller that has been tested in a lab setting. The measurement of alert latency, blocking accuracy, business-service disruption, recovery time, adversarial adaptation, and analyst workload would be required before production deployment. The above is even more relevant given the research on adversarial machine learning and how NIDS models can be manipulated with realistic perturbations on their features, and how robustness guarantees from other domains do not directly carry over to network traffic [21].

There are thus a number of limitations in the next research stage. UNSW-NB15 is a well known benchmark and not a real-time contemporary stream, there are still heavily imbalanced attack categories, and there are possible cases of attacks being repeated as well as different attacks. The cleaned test protocol results in an increased independence, but a decreased direct comparability with papers using the canonical prepared partitions unchanged. The study also fails to perform temporal holdout, cross-dataset testing, adversarial robustness testing, probability recalibration or live mitigation experiments. Future studies are needed to test the trained pipeline on more recent heterogeneous datasets and in real or emulated network settings, and to add drift monitoring and to explore thresholds per class in Fuzzer traffic, as well as operational consequences of the mitigation policy. Theoretical relevance: Leakage control/explanation as part of model validity; Practical relevance: Leakage control/explanation as part of a reproducible detector whose decisions can be checked prior to automation of response actions.

6. CONCLUSION

This study developed a leakage-aware AI-driven network security framework for intelligent threat detection and mitigation using UNSW-NB15. The workflow included grouping the model development and independent evaluation, excluding 8,541 test records, where predictor vectors were also used for training, and including learned preprocessing within training-fitted pipelines. For logistic regression, decision tree, random forest and Extra Trees, the random forest had the highest development ROC-AUC (0.9908) and also had 0.9071 accuracy, 0.9642 recall, 0.9150 F1-score, 0.9820 ROC-AUC and 0.9848 average precision on 73,791 leakage-clean test records. Permutation importance suggested that the most influential variables that should be included in the models are the contextual connection, the state/TTL, service and traffic-volume variables, and the error analysis showed that the main cause of false negatives is the Fuzzer traffic, which is 92.48%. The severity-aware mitigation layer was capable of correlating malicious predictions with transparent response recommendations without making any unmeasured claims of effectiveness of control. It is therefore not a new learning algorithm as the main contribution is a framework for evaluating and responding that can be reproduced. Some limitations are the use of only one well known benchmark, the residual class imbalance, a lack of cross-dataset and temporal validation, and a lack of live mitigation tests. The effects of distribution shift, adversarial robustness, threshold calibration, drift adaptation, and operational response outcomes should be further investigated on modern networks.

REFERENCES

1. N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 2015, pp. 1-6, doi: 10.1109/MilCIS.2015.7348942.
2. N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," Information Security Journal: A Global Perspective, vol. 25, nos. 1-3, pp. 18-31, 2016, doi: 10.1080/19393555.2015.1125974.
3. 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.
4. 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.
5. 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.
6. N. Moustafa, J. Hu, and J. Slay, "A holistic review of Network Anomaly Detection Systems: A comprehensive survey," Journal of Network and Computer Applications, vol. 128, pp. 33-55, 2019, doi: 10.1016/j.jnca.2018.12.006.

7. M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, art. no. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.
8. N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," in *Proc. 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, Kyoto, Japan, 2015, pp. 25-31, doi: 10.1109/BADGERS.2015.014.
9. R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, art. no. 322, 2019, doi: 10.3390/electronics8030322.
10. S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *Journal of Big Data*, vol. 7, art. no. 105, 2020, doi: 10.1186/s40537-020-00379-6.
11. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525-41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
12. M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in *Big Data Technologies and Applications*, LNICST, vol. 371, Springer, 2021, pp. 117-135, doi: 10.1007/978-3-030-72802-1_9.
13. M. H. Kabir, M. S. Rajib, A. S. M. T. Rahman, M. M. Rahman, and S. K. Dey, "Network Intrusion Detection Using UNSW-NB15 Dataset: Stacking Machine Learning Based Approach," in *Proc. 2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*, Gazipur, Bangladesh, 2022, pp. 1-6, doi: 10.1109/ICAEEE54957.2022.9836404.
14. N. A. Rathod, T. Gupta, N. V. Sharma, and S. Sharma, "Model Comparison and Multiclass Implementation Analysis on the UNSW NB15 Dataset," in *Proc. 2021 International Conference on Computational Performance Evaluation (ComPE)*, 2021, pp. 549-555, doi: 10.1109/COMPE53109.2021.9751832.
15. A. Fathima, A. Khan, M. F. Uddin, M. M. Waris, S. Ahmad, C. Sanin, and E. Szczerbicki, "Performance Evaluation and Comparative Analysis of Machine Learning Models on the UNSW-NB15 Dataset: A Contemporary Approach to Cyber Threat Detection," *Cybernetics and Systems*, vol. 56, no. 8, pp. 1160-1176, 2025, doi: 10.1080/01969722.2023.2296246.
16. O. Arreche, T. Guntur, and M. Abdallah, "XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems," *Applied Sciences*, vol. 14, no. 10, art. no. 4170, 2024, doi: 10.3390/app14104170.
17. D. Wells, "UNSW_NB15," Kaggle, [Online]. Available: <https://www.kaggle.com/datasets/mrwellsdavid/unswnb15>. [Accessed: Aug. 11, 2026]
18. M. K. Hooshmand, M. D. Huchaiah, A. R. Alzighaibi, H. Hashim, E.-S. N. Elsayed, and I. Gad, "Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (XAI)," *Alexandria Engineering Journal*, vol. 94, pp. 120-130, 2024, doi: 10.1016/j.aej.2024.03.041.
19. M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," *Information Sciences*, vol. 639, art. no. 119000, 2023, doi: 10.1016/j.ins.2023.119000.
20. M. Cantone, C. Marrocco, and A. Bria, "Machine Learning in Network Intrusion Detection: A Cross-Dataset Generalization Study," *IEEE Access*, vol. 12, pp. 144489-144508, 2024, doi: 10.1109/ACCESS.2024.3472907.
21. K. He, D. D. Kim, and M. R. Asghar, "Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538-566, 2023, doi: 10.1109/COMST.2022.3233793.