Vol. 08, No. 3 (2026) 1857-1876, doi: 10.24874/PES08.03.035

Proceedings on Engineering Sciences

www.pesjournal.net

AN OPTIMIZED TWO-TIER INTRUSION DETECTION SYSTEM USING BAYESIAN AND HYPERBAND-TUNED XGBOOST

Girraj Khandelwal [1]
Gajanand Sharma
Brij Bhooshan Gupta

Received 17.06.2026.
Revised 06.08.2026.
Accepted 28.08.2026.

Keywords:

Intrusion detection system, Two-tier classification, XGBoost, Bayesian optimization, Hyperband, Hyperparameter optimization, Ensemble learning, SHAP, Network security

A B S T R A C T

Traditional intrusion detection systems relatively rigid detection mechanisms are increasingly insufficient against sophisticated cyberattacks as modern network infrastructures grow in complexity. In this work a hybrid, optimization-driven IDS framework to be induce that integrates feature engineering, class-imbalance handling, Bayesian and Hyperband-tuned XGBoost classifiers, hybrid ensemble learning, and SHAP-based explainable artificial intelligence for cyberattack detection. The proposed framework adopts a two-tier architecture in which the first tier performs binary classification to separate normal from malicious traffic, while the second tier performs multi-class classification of the malicious traffic identified by the first tier into specific attack categories. To improve detection performance and computational efficiency, the framework applies leakage-feature elimination and correlation-based feature reduction, followed by class balancing, prior to model training. Hybrid ensemble models based on stacking and soft voting over the tuned base classifiers are used to improve classification robustness, and SHAP-based explainability is incorporated to support interpretation of the features driving each classification decision. Experimental evaluation shows that the proposed model achieves an accuracy of 99.81%, a precision of 96.36%, a recall of 94.32%, an F1-score of 95.27%, and an area under the ROC curve in the neighborhood of 0.999. These results indicate that the proposed two-tier, dual-optimization IDS framework improves detection robustness relative to conventional single-tier machine-learning baselines evaluated on the same data, while maintaining explainability through SHAP-based analysis.

© 2026 Published by Faculty of Engineering

1. INTRODUCTION

1.1 Cybersecurity Context

The proliferation of connected systems, such as cloud architectures and the Internet of Things (IoT), has dramatically changed the nature of networks and increased their vulnerability to advanced cyberattacks. Secure networking infrastructure today is a critical component of all operating businesses across industry and government sectors. This changing threat model compels researchers towards building more and more sophisticated protective mechanisms. This effort typically involves the use of an intrusion detection system (IDS), which keeps watch on network or host

Corresponding author: Girraj Khandelwal
Email: girraj13@gmail.com

activity and can identify anomalous or unauthorized activity that suggests a security breach is occurring. There are mainly two types of IDS approaches, signature-based and anomaly-based systems. Signature-based IDSs rely on attack signatures from the past, and while they are highly effective at detecting known attacks, their reliance on previously identified patterns can make them unable to detect novel or zero-day attacks. Anomaly-based IDSs, on the other hand, treat deviations from a learned model of "normal" behaviour as potential threats; however, defining "normal" precisely across diverse hosts and traffic conditions is challenging at best, and large-scale heterogeneous network traffic often yields high false-positive rates along with excessive computational burden when using anomaly-based methods. These limitations necessitate more dynamic and adaptive IDS designs that can accurately classify legitimate and abnormal behaviour (Alsubaei, 2025). The general location of an intrusion detection system in a monitored network is shown in figure 1. An IDS is ideally placed to monitor traffic and alert on suspicious behavior before any event has reached the hosts or services it protects.

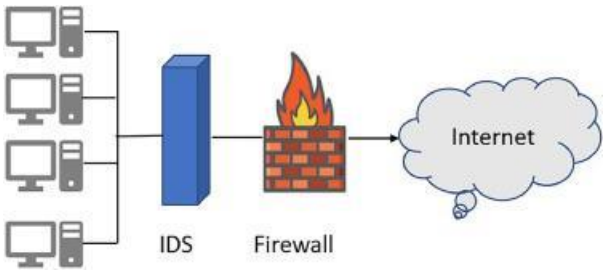


Figure 1. A network intrusion detection system

1.2 Existing Machine-Learning-Based Approaches and Their Limitations

Network security has been transformed in recent years by ML-based IDS frameworks. Because ML-based IDS learn from historical training data, they can identify complex, non-linear attack patterns that are not easily detected by signature- or rule-based techniques, making them a good fit for complex, dynamic, and high-dimensional network environments. At the same time, an ML-based IDS is highly dependent on feature representation, model selection, and hyperparameter settings, such that improper settings lead to drastic degradation in detection accuracy and latency, making the system impractical for deployment. Among ML techniques, ensemble learning methods, and gradient-boosted tree ensembles such as XGBoost in particular, are widely regarded as effective for regularization, native handling of missing data, and robust performance under high-dimensional, noisy conditions, both practically and theoretically. Realizing these benefits in real-time deployment, however, requires tuning hyperparameters, itself a search problem that is resource-intensive and one of the most critical factors influencing the effectiveness of any boosted-tree IDS.

1.3 Problem Statement

Many existing IDS approaches, including those based on machine learning summarized for high aggregate detection accuracy on benchmark datasets using a single flat classification stage over all attack categories and benign traffic. This design mixes two different sub-problems, coarse normal-versus-attack discrimination and fine-grained attack-type identification, into a single model with a single set of hyperparameters. In addition, hyperparameter search is often still performed manually, or with costly exhaustive methods such as grid search, which does not scale well to the high-dimensional hyperparameter spaces of modern gradient-boosted tree models. This produces an ongoing trade-off between detection accuracy, false-alarm rate, computational cost, and interpretability, particularly when the underlying classifier must remain effective across multiple benchmark datasets with different feature schemas and attack taxonomies. This work addresses the designs, implements, examines, and evaluates a multi-level IDS architecture that uses XGBoost with two complementary hyperparameter-optimization techniques, Bayesian optimization and Hyperband, to tune the classifier at each level of the hierarchy, and then augments those classifiers with hybrid ensembling and SHAP-based explainability.

1.4 Proposed Approach

This paper proposes a two-tier IDS architecture based on XGBoost, in which tier 1 performs binary classification to identify normal and malicious traffic, and tier 2 subsequently classifies the malicious traffic flagged by tier 1 into a specific attack category through multi-class classification. This layered design reflects the practice of real-world network monitoring, in which broad, low-latency filtering is followed by more fine-grained investigation and confirmation of suspicious activity. Because each tier of the proposed framework is tuned separately using both Bayesian optimization and Hyperband, it provides a direct, controlled way to compare these two optimization strategies on the same classifier applied to the same task, rather than the more typical evaluation of a single tuning method in isolation. Bayesian optimization is a sequential, model-based search strategy that fits a probabilistic surrogate to approximate validation performance as a function of hyperparameters, balancing exploration of uncertain regions against exploitation of promising ones. Hyperband, by contrast, treats hyperparameter search as a resource-allocation problem, combining random sampling with successive halving and early stopping to discard poorly performing configurations quickly and allocate computational budget to promising ones.

A two-tier design offers several structural advantages over a single flat classifier. By splitting detection into two sequential stages, the decision complexity each stage must resolve is reduced, and hyperparameter

optimization can be conducted on a well-defined sub-task: the first tier need only distinguish between normal and abnormal traffic, while the second tier's task is only to discriminate among attack categories once normal traffic has already been filtered out. This also lightens the load further combines the outputs of the Bayesian-optimized and Hyperband-optimized XGBoost models to draw on the strengths of both without committing to a single optimization strategy.

1.5 Contributions

The contributions of this work are as follows:

- A novel two-tier IDS architecture that separates binary and multi-class attack detection into two independently optimized tiers
- A controlled comparison of Bayesian optimization and Hyperband for tuning XGBoost within the same two-tier structure
- The hybrid ensemble models based on stacking and soft voting over the tuned base classifiers is used.
- Based on the SHAP framework explainability integrated the model into the detection pipeline
- The performance metrics such as accuracy and precision, on the standard UNSW-NB15 and CICIDS2017 datasets to be used in practical evaluation.

2. BACKGROUND

This section provides the technical background needed to follow the proposed methodology a taxonomy of intrusion detection systems, an overview of XGBoost, hyperparameter optimization through Bayesian approaches, the Hyperband algorithm, and the general idea of a two-tier detection architecture.

2.1 Intrusion Detection Systems

An intrusion detection system monitors network traffic or host activity and raises an alert when it observes behavior consistent with an attack. IDSs are commonly classified along two axes. By detection strategy, a signature-based (misuse-based) IDS matches observed traffic against a database of known attack patterns; it achieves a low false-positive rate for known attacks but cannot, by construction, recognize attacks that do not match an existing signature. An anomaly-based IDS instead models “normal” behavior, typically using statistical or machine-learning techniques, and flags significant deviations from that model as suspicious; this allows detection of previously unseen attacks but tends to produce more false positives, since legitimate but unusual traffic can resemble an anomaly. By monitoring scope, a network-based IDS (NIDS) inspects traffic at one or more points in the network, while a host-based IDS (HIDS) monitors activity on an individual host, such as system calls or log files. The framework proposed in this paper is a network-based, anomaly-oriented, machine-learning IDS evaluated on flow-level network traffic features drawn from the UNSW-NB15 and CICIDS2017 datasets.

Machine-learning-based IDSs replace hand-crafted signatures or simple statistical anomaly models with a classifier trained on labeled traffic. Supervised classifiers, the category to which the proposed framework belongs, are trained on traffic already labeled as normal or as belonging to a specific attack category, and are then applied to unseen traffic. This requires labeled training data with adequate coverage of both normal and attack behavior, and it introduces a corresponding risk: a classifier can achieve high accuracy on a benchmark's labeled test partition while nonetheless generalizing poorly to attack types, or attack variants, that are not represented in the training data. This risk motivates, among other things, the class-imbalance handling and cross-dataset evaluation discussed later in the manuscript.

2.2 Gradient Boosting and XGBoost

Gradient boosting builds a strong classifier as an additive ensemble of weak learners, typically shallow decision trees, added one at a time so that each new tree corrects the residual errors of the ensemble built so far. Extreme Gradient Boosting (XGBoost) is a widely used, engineering-optimized implementation of gradient-boosted decision trees that adds an explicit regularization term to the standard boosting objective to control model complexity, together with algorithmic optimizations, approximate split finding, sparsity-aware handling of missing values, and parallelized tree construction, that make it practical on large, high-dimensional, and noisy tabular data such as network-flow features. These properties make XGBoost a natural candidate classifier for both tiers of the proposed IDS.

2.3 Bayesian Optimization

Hyperparameter optimization can be formulated generally as

$$x^* = arg\ max_{x \in X}\ f(x)$$

where x denotes a candidate hyperparameter configuration drawn from the search space X, and $f(x)$ denotes a measure of model performance obtained by training and evaluating the model with configuration x. Because evaluating $f(x)$ requires a full training-and-validation cycle, it is expensive to query, and grid or random search can require many such evaluations to locate a good configuration. Bayesian optimization addresses this by fitting a probabilistic surrogate model, commonly a Gaussian process, to the configurations evaluated so far, and using an acquisition function built from that surrogate to decide which configuration to evaluate next. The acquisition function is designed to balance exploitation sampling near configurations

already known to perform well against exploration sampling in regions of the search space where the surrogate model is uncertain, which typically allows Bayesian optimization to locate strong configurations using substantially fewer evaluations than exhaustive search. The formal treatment of the surrogate model and acquisition function, together with the hyperparameter search space used for the proposed framework's binary and multi-class XGBoost tiers.

2.4 Hyperband

Hyperband is a resource-allocation strategy for hyperparameter optimization that treats "resource" (for example, the number of boosting rounds, or the size of the training subset) as an explicit tunable dimension. Rather than training every candidate configuration to completion, Hyperband allocates a small resource budget to a large number of randomly sampled configurations, evaluates them, discards the worst-performing fraction, and reallocates the freed budget to the survivors in a process known as successive halving; this is repeated across several brackets that trade off the number of configurations against the resource budget per configuration. Because poor configurations are discarded early, before they consume a full training budget, Hyperband can explore a substantially larger number of candidate configurations than Bayesian optimization or grid search for the same total computational budget, at the cost of not using a probabilistic model to guide the search.

2.5 Two-Tier Intrusion Detection

A two-tier (or hierarchical, cascaded) IDS architecture splits the overall detection task into two sequential classification stages instead of a single flat multi-class classifier. In the design adopted in this paper, the first tier performs binary classification (normal versus attack), and the second tier, applied only to traffic the first tier has flagged as an attack, performs multi-class classification into specific attack categories. This structure is motivated by the observation that the binary decision and the fine-grained attack-category decision are qualitatively different sub-problems with different class distributions and, potentially, different discriminative features; separating them allows each tier's XGBoost model to be optimized independently, using a search space and resource budget appropriate to its own sub-problem, rather than forcing a single classifier and a single hyperparameter configuration to serve both.

3. LITERATURE REVIEW

The reviews related to the work thematically rather than paper by paper, grouping studies by the technique they emphasize and, for each, summarizing the problem addressed, the dataset and method used, the main findings, and the limitations relevant to the present work.

3.1 Machine-Learning-Based Intrusion Detection

Several recent studies apply conventional and hybrid machine-learning classifiers directly to benchmark intrusion-detection datasets. Ajagbe et al. (2024) compared LinearSVC, logistic regression, random forest, decision tree, and XGBoost for binary intrusion detection on the UNSW-NB15 dataset (49 features, split 70/15/15 into training, validation, and test sets), reporting that random forest and XGBoost achieved the best classification accuracy for normal traffic (92.71%) and the best recall and F1-score among the compared models; the study's limitations include bias toward majority classes in an imbalanced split, a risk of overfitting without cross-validation, and limited scalability for real-time industrial IoT deployment. Alrabanah et al. (2025) combined long short-term memory (LSTM) networks with XGBoost, using the recurrent component to capture temporal dependencies in traffic and XGBoost for final classification, and reported accuracy, precision, recall, and F1-score all near 99% on a standardized traffic-anomaly dataset; the approach nonetheless retains a high false-positive rate typical of anomaly-based methods, incurs deployment overhead on large-scale networks, and struggles to identify threats not represented in training data. Karthikeyan et al. (2024) proposed a firefly-algorithm-based feature-weighting scheme combined with Grey Wolf Optimization for WSN-IoT intrusion detection on the NSL-KDD dataset (41 features), reporting 99.34% testing accuracy, which outperformed KNN with particle-swarm optimization (96.42%) and unoptimized XGBoost (95.36%) in their comparison; limitations include the computational cost of the GWO-FA convergence process, susceptibility to local optima in high-dimensional search, and evaluation limited to simulated rather than live WSN/IoT traffic.

3.2 Ensemble Learning for Intrusion Detection

Ensemble methods that combine multiple base classifiers have been widely applied to improve IDS robustness. Bibers et al. (2025) developed an ensemble framework combining bagging, boosting, stacking, and blending with information-gain and K-best feature selection, evaluated on the RoEduNet-SIMARGL2021 and CICIDS2017 datasets, and reported F1-scores above 0.996 for blending and bagging on CICIDS2017; the main drawbacks identified by the authors are the high resource requirements of stacking and blending, reduced training-time efficiency without feature selection, and performance that varies across different threat landscapes. Hosain and Çakmak (2025) proposed an interpretable IDS for the Internet of Medical Things combining hybrid random sampling for class-imbalance correction, recursive feature elimination, and a tuned

XGBoost classifier with SHAP and LIME explainability, evaluated on the WUSTL-EHMS-2020 dataset (network flow and biometric-signal features), reporting 99.22% accuracy, 98.35% precision, 99.91% recall, and 99.12% F1-score; limitations include sensitivity to dataset-specific outliers, the computational overhead of interpretability methods on edge devices, and unresolved latency on ultra-low-power healthcare nodes. Kodete et al. (2025) proposed "TripleBoost," a meta-ensemble combining AdaBoost with XGBoost together with SMOTE oversampling, Z-score-based outlier and feature selection, and ridge regularization, trained on a 2.54-million-sample subset of UNSW-NB15 (49 features across nine attack families), reporting 97.38% accuracy, 95.34% precision, 99.56% recall, and 96.40% F1-score; the authors note that the iterative inference process is computationally expensive, convergence is slow under resource constraints, and generalization to other corpora such as NSL-KDD variants remains to be validated.

3.3 XGBoost-Based Intrusion Detection

XGBoost specifically has been combined with a range of preprocessing and feature-engineering pipelines. Alashjaee and Alqahtani (2025) combined a feedforward neural network with XGBoost for deep feature extraction and classification on the CIC-IoT-2023 dataset, using principal component analysis to address high dimensionality and class imbalance in both binary and multi-class settings, and reported approximately 99% accuracy; limitations include significant computational overhead on resource-constrained IoT devices, limited scalability to new attack vectors, and dependence on balanced resampling for rare classes. Kandasamy and Roseline (2025) proposed "AEXB," combining an autoencoder with XGBoost for multi-class classification of home-network traffic, using recursive feature elimination and correlation analysis to reduce multicollinearity, and reported 97.24% accuracy on the IDSH dataset (4,143 records, 22 features) while reducing false positives through data-augmentation techniques; drawbacks include high memory consumption for distance-based classifiers used for comparison, such as k-nearest neighbors, and limited scalability to large IoT datasets with diverse attack vectors. Al Amien et al. (2023) studied a weighted-XGBoost classifier that embeds class-imbalance ratios directly into the boosting objective, evaluated on NSL-KDD (41 attributes); the approach improved minority-class detection relative to standard unweighted baselines but, unlike other studies reviewed here, was not evaluated across multiple datasets. Faizin et al. (2024) combined mutual-information-based thresholding feature selection with XGBoost classification across UNSW-NB15 (49 attributes), NSL-KDD (41 attributes), and CIC-IDS2017, reporting 99.89% accuracy and, after applying thresholding to the full CIC-IDS2017 feature set, a 99.68% F1-score; the authors note that the thresholds are dataset-specific and do not generalize across datasets, that training time increases substantially without parallelism (particularly for multi-class classifiers), and that the approach would be ineffective against zero-day attacks in the absence of adaptive retraining. Using the complete feature set, Kareem et al. (2023) created an XGBoost-based model to detect application-layer distributed denial-of-service attacks on CIC-IDS2017 that achieved 99.9%, 99.7%, 99.5%, and 99.6% accuracy, recall, precision, and F1-score, respectively, compared with k-nearest neighbors, support vector machines, and PCA-XGBoost hybrids; efficient inference on larger-scale data, as well as adversarial robustness against obfuscated attacks, are identified by the authors as open problems.

The same broad line of work also includes class-weighted XGBoost for IoT traffic with pronounced class imbalance, such as the Bot-IoT-oriented weighting scheme summarized by Kareem et al. (2023) alongside their DDoS-focused results, which improves detection of rare intrusion classes relative to unweighted baselines but depends on accurate weight calibration and has not been validated on dynamic, real-time IoT streams.

3.4 Bayesian Optimization and Efficient Hyperparameter Search for Machine Learning

Comparatively few of the reviewed IDS studies report a formal hyperparameter-search methodology; where search is reported, grid search remains the dominant approach, which the present work argues does not scale well to the hyperparameter spaces of gradient-boosted tree models. Sajid et al. (2024) is among the more methodologically explicit examples: their hybrid machine-learning-and-deep-learning IDS was tested on NSL-KDD (41 features), CIC-IDS2017, and UNSW-NB15 (49 features), using XGBoost for feature selection and reduction ahead of ANN, deep convolutional neural network, and BiLSTM classifiers, with SMOTE oversampling to address class imbalance; the ANN-BiLSTM configuration achieved the highest accuracy and lowest error rates across the three benchmarks. The authors report that the hyperparameters of the deep-learning components were not systematically optimized for unknown attack types, which the present work identifies as a specific instance of the more general problem that motivates the systematic Bayesian and Hyperband search.

3.5 Hyperband and Resource-Efficient Optimization

Hermosilla et al. (2025) compared SHAP and LIME explainability methods for XGBoost and TabNet classifiers on the UNSW-NB15 dataset in a forensic-analysis context, reporting that XGBoost combined with SHAP achieved superior explanation coherence and a validation accuracy of 97.8%; the study notes LIME's limited universality in local reasoning, the high

computational cost of SHAP on large datasets, and limited forensic admissibility without comprehensive global-and-local validation. While this study does not itself apply Hyperband, its computational-cost findings for SHAP on large tabular datasets are directly relevant to the SHAP-based explainability component of the present framework to the choice of Hyperband, alongside Bayesian optimization, as a comparatively resource-efficient tuning strategy for the underlying XGBoost classifier.

3.6 Multi-Tier and Hierarchical Intrusion Detection

Chimphlee and Chimphlee (2023) evaluated random-forest-based feature-importance ranking alongside logistic regression, k-nearest neighbors, CART, naive Bayes, multilayer perceptron, and XGBoost classifiers on the CSE-CIC-IDS2018 dataset, applying undersampling to correct class imbalance before multi-class detection, and found that a multilayer perceptron using RF-selected top features achieved the best overall accuracy, precision, recall, and F1-score among the compared models; limitations include high false-alarm rates for minority classes after undersampling, heavy computational cost from exhaustive feature searches, and reduced generalization to non-representative or larger-scale samples. Shree and Thangaraj (2025) proposed a semantic, knowledge-graph-based IDS in which a sparse autoencoder learns feature weights that are encoded into a weighted knowledge graph and fed to a deep autoencoder-LSTM for sequence-based anomaly detection, with an XGBoost ensemble refining the resulting attack-likelihood estimates and SMOTE addressing class imbalance; using 80 CICFlowMeter-derived attributes from the CSE-CIC-IDS2018 dataset, the model achieved 99.68% accuracy, 99.65% precision, 99.70% recall, and 99.67% F1-score, outperforming an MLP-backpropagation baseline in their comparison. Although neither of these studies adopts an explicit two-tier architecture in the sense used in the present work, both illustrate a closely related design principle, namely staging feature reduction, resampling, and classification into separate, purpose-specific steps, which the two-tier architecture proposed here extends by staging the classification decision itself into a binary tier followed by a multi-class tier, each independently hyperparameter-tuned.

3.7 Recent State-of-the-Art and Cross-Domain IDS Approaches

A number of recent studies extend XGBoost-based detection to adjacent or cross-domain settings. Cavus et al. (2025) proposed a cyber-resilient design integrating a hybrid optimization-reinforcement-learning algorithm with an IDS and a blockchain-inspired protocol for electric-vehicle grid integration, evaluated on a European smart-grid dataset combining dynamic electricity pricing, EV mobility patterns, and grid-load dynamics, and reported 94.1% IDS accuracy and a 0.97 AUC, with a demand reduction of 9.6%; limitations include runtime exceeding 0.4 seconds for large datasets, detection capability dropping to 74% once more than half of the evaluated attacks are novel to the system, and open challenges in spatial infrastructure planning. Sajid et al. (2024), discussed above for its hybrid deep-learning architecture, and Binsaeed and Hafez (2023), who integrated XGBoost-based feature selection with deep-learning classifiers (ANN, deep CNN, and BiLSTM) using SMOTE oversampling on NSL-KDD, CIC-IDS2017, and UNSW-NB15, both illustrate the trend toward combining gradient-boosted feature selection with deep sequence models; Binsaeed and Hafez (2023) note that SMOTE-introduced synthetic noise can increase false positives, that training deep models on unlabeled flows is computationally demanding, and that hyperparameters were not optimized specifically for unknown attack types.

More broadly, optimization-algorithm-driven hybrid learning has also been applied to adjacent cybersecurity problems outside network intrusion detection specifically, illustrating the wider relevance of the optimization-plus-ensemble design principle adopted in this paper. Gupta et al. (2025) combined an earthworm optimization algorithm with a cascaded LSTM-GRU architecture for Android malware detection, and Singh et al. (2025) examined optimization techniques for secure quantum communication and robust key-distribution protocols. Neither study addresses network intrusion detection directly, so they are discussed here only for their shared methodological theme of optimization-driven security machine learning, rather than as direct precedents for the two-tier IDS architecture itself.

3.8 Synthesis and Limitations of Existing Studies

Across the studies reviewed some limitations to be found out like First, reported detection accuracy is generally validated only on static, offline benchmark partitions rather than in live or continuously evolving network environments, leaving open questions about latency, computational overhead, and deployment feasibility (Alsubaei, 2025; Karthikeyan et al., 2024). Second, few of the reviewed studies systematically compare more than one hyperparameter-optimization strategy on the same classifier and task; where hyperparameter search is reported at all, it is typically a single method (frequently grid search) applied without a documented search space or a comparison against an alternative search strategy, which limits the field's evidence base for choosing among tuning methods for boosted-tree IDS models (Sajid et al., 2024; Ajagbe et al., 2024). Third, several ensemble and deep-hybrid architectures achieve strong benchmark performance at the cost of substantial computational overhead, which the authors of those studies themselves identify as a

barrier to deployment on resource-constrained or large-scale infrastructure (Bibers et al., 2025; Hermosilla et al., 2025). Fourth, the majority of the reviewed studies adopt a single flat classification stage rather than a hierarchical, staged design, which conflates the binary normal-versus-attack decision with fine-grained attack-category identification inside one model and one hyperparameter configuration, reducing both interpretability and the system's capacity to support rapid, category-specific threat response. These four limitations, taken together, motivate the two-tier, dual-optimization-strategy architecture proposed in this paper.

4. RESEARCH GAP AND MOTIVATION

4.1 What Existing Studies Have Achieved

The existing machine-learning-based IDS studies have demonstrated that gradient-boosted tree classifiers, and XGBoost in particular, can achieve high detection accuracy on standard intrusion-detection benchmarks such as UNSW-NB15, NSL-KDD, and the CIC-IDS/CICIDS2017 family, often when combined with feature selection, class-imbalance correction, or ensembling. A smaller number of studies have also begun to incorporate model interpretability through SHAP or LIME and to apply gradient-boosted or hybrid deep-learning classifiers to related but distinct security domains, such as smart-grid cyber-resilience and Android malware detection.

4.2 What Remains Unresolved

Despite this progress, three methodological gaps recur across the reviewed literature.

- First, hyperparameter optimization for boosted-tree IDS models is inconsistently reported and rarely benchmarked against more than one search strategy; where a systematic search is described at all, it is typically a single method applied without an explicitly reported search space, which makes it difficult to compare the efficiency or effectiveness of alternative tuning strategies for this class of model.
- Second, most of the studies reviewed use a single flat classification stage, which must simultaneously resolve the coarse normal-versus-attack decision and the fine-grained within-attack-category decision, mixing two sub-problems with different class distributions into one hyperparameter setting.
- Third, a number of the strongest ensembles or hybrid architectures themselves identify their own computational cost, for stacking, blending, or deep sequential components, as an obstacle to practical use, motivating a benchmark of resource-efficient optimization strategies such as Hyperband alongside Bayesian optimization, rather than relying entirely on exhaustive search.

4.3 Why Existing Approaches Are Insufficient

As discussed the detection task decomposes into two distinct sub-problems, binary triage and multi-class attack categorization, with different data characteristics, a single flat classifier tuned with one hyperparameter configuration and one search strategy is unlikely to be well matched to both. Relying on a single hyperparameter-search strategy, without reporting the search space or comparing it against an alternative, also leaves open a question directly relevant to deployment: which hyperparameter-search strategy is preferable for a boosted-tree IDS under a limited computational budget, given that different strategies can have substantially different resource requirements.

4.4 Why a Two-Tier Architecture, XGBoost, Bayesian Optimization, and Hyperband are Selected

A two-tier architecture directly addresses the first gap above by decomposing the overall detection task into a binary tier and a multi-class tier, each with its own hyperparameter configuration and its own hyperparameter search. XGBoost is selected as the base classifier for both tiers because of its established regularization, native handling of missing and sparse features, and demonstrated performance on high-dimensional tabular network-flow data in the studies reviewed. Bayesian optimization and Hyperband are selected together, rather than a single strategy in isolation, precisely because they represent two structurally different approaches to the same search problem, a model-based sequential search and a resource-allocation-based search, so that comparing them directly on the same classifier and the same two-tier task provides evidence relevant to the second and third gaps identified above.

4.5 What the Proposed Combination Adds

The proposed combination, a two-tier XGBoost architecture in which each tier is independently tuned using both Bayesian optimization and Hyperband, and whose tuned base classifiers are further combined through hybrid stacking and soft-voting ensembles with SHAP-based explainability, has not, to the authors' knowledge based on the literature reviewed been evaluated together as a single system in prior IDS research. This combination is intended to provide comparative evidence on hyperparameter-search efficiency for boosted-tree IDS models, an explicit architectural separation between coarse and fine-grained detection decisions, and an interpretability layer suited to operational security analysis, within a single, consistently evaluated framework.

5. RESEARCH CONTRIBUTIONS AND NOVELTY

5.1 Existing Knowledge

Gradient-boosted tree classifiers, and XGBoost specifically, are established as effective classifiers for tabular network-intrusion data. Bayesian optimization and Hyperband are each individually established, general-purpose hyperparameter optimization techniques in the broader machine-learning literature, and hybrid ensembling (stacking, soft voting) and SHAP-based explainability are each established techniques for, respectively, improving classifier robustness and improving interpretability.

5.2 Limitations of Existing IDS Approaches

The existing IDS studies rarely compare more than one hyperparameter-search strategy on the same classifier and task, rarely separate binary triage from fine-grained attack categorization into independently tuned stages, and frequently trade detection performance for substantial computational overhead in their ensembling or deep-learning components.

5.3 Research Gap

The specific gap this work targets, developed is the absence of a controlled, side-by-side comparison of Bayesian optimization and Hyperband for tuning XGBoost within an explicitly two-tier (binary-then-multiclass) IDS architecture, combined with hybrid ensembling and SHAP-based explainability evaluated as one coherent system.

5.4 Proposed Solution

The proposed solution is the two-tier IDS framework a binary XGBoost tier followed by a multi-class XGBoost tier applied only to traffic the binary tier has flagged as malicious, with each tier's hyperparameters searched independently using both Bayesian optimization and Hyperband and with hybrid stacking and soft-voting ensembles built over the resulting tuned classifiers.

5.5 Scientific Contribution

The scientific contribution is comparative evidence, drawn from the experiments reported on how Bayesian optimization and Hyperband perform against one another when applied to the same classifier and the same two-tier detection task, rather than an evaluation of either method in isolation.

5.6 Technical Contribution

The technical contribution is the two-tier architectural design itself, in which the binary and multi-class classification decisions are resolved by separately optimized XGBoost models rather than by a single flat classifier, together with the hybrid stacking and soft-voting ensembling layer built on top of the tuned base classifiers.

5.7 Experimental Contribution

The main experimental contribution is an evaluation of this framework against a range of common machine-learning baselines naive Bayes, k-nearest neighbors, logistic regression, random forest, unoptimized XGBoost, and LightGBM on two benchmark datasets, UNSW-NB15 and CICIDS2017.

5.8 Practical Contribution

A practical contribution is the coupling of SHAP-based explainability to a detection pipeline to assist a security analyst in interpreting which specific network-traffic features drove a particular classification decision, contrasting with treating the classifier as a black box.

5.9 A Note on Claims of Novelty

Consistent with the principle that novelty should not be claimed merely because established techniques are combined, this manuscript does not claim to be the first study to apply XGBoost, Bayesian optimization, Hyperband, ensembling, or SHAP individually to intrusion detection each of these components is individually well established. The contribution claimed here is the specific combination and the controlled comparison it enables not the invention of any individual component technique.

6. PROPOSED TWO-TIER IDS METHODOLOGY

6.1 Overview and Workflow Approach

The proposed framework follows an explicit, model-based methodology for designing an adaptive intrusion detection system, in which conventional machine-learning classifiers are first evaluated with the full available feature set as a baseline before the tuned two-tier XGBoost architecture is introduced; this baseline-first approach is motivated by the continuously evolving strategies and methods used by attackers, which require the detection system's own design choices to be justified against clearly reported baselines rather than assumed (Gupta et al., 2023). The overall workflow proceeds as: network traffic input → preprocessing and feature engineering → Tier 1 binary classification → Tier 2 multi-class classification (applied only to traffic flagged

as malicious by Tier 1) → final classification and alert generation, illustrated in figure 2.

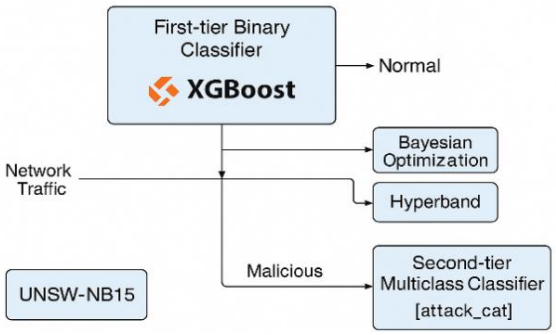


Figure 2: Optimized Two-Tier IDS Architecture

The two-tier IDS architecture proposed is illustrated in figure 2. Incoming network traffic is processed through preprocessing and feature engineering before being fed into a first-tier binary XGBoost classifier, with all hyperparameters independently tuned via either Bayesian optimization or Hyperband. Traffic classified as legitimate by the first tier is filtered out of further analysis, while traffic classified as an attack is forwarded to a second-tier multi-class classifier that assigns it an attack category drawn from the UNSW-NB15 attack_cat label using its own independently tuned XGBoost model. This hierarchical design allows each stage to be optimized for its own objective: the top tier is tuned for speed and high-recall binary triage, while the bottom tier focuses on fine-grained discrimination between attack types once normal traffic has already been excluded from consideration.

6.2 Tier 1: Binary Classification Layer

This layer differentiates between normal and malicious network traffic. The dataset is preprocessed, filtered, and normalized prior to training, after which an XGBoost classifier with hyperparameters tuned separately by Bayesian optimization and Hyperband is fitted for the binary decision. The workflow of this tier, illustrated in figure 3, proceeds as follows:

- Perform data preprocessing, feature selection, and normalization.
- Split the dataset into training and testing sets using an appropriate validation method, using the binary attack label (normal/attack) as the target variable.
- Define the hyperparameter search space for both BO and HB.
- Apply cross-validation to identify the best-performing hyperparameter configuration for each of BO and HB.
- Train the XGBoost classifier using the best configuration obtained from each of BO and HB.
- Evaluate model performance using the metrics defined.
- Classify incoming network instances as normal or attack using the resulting tuned models.

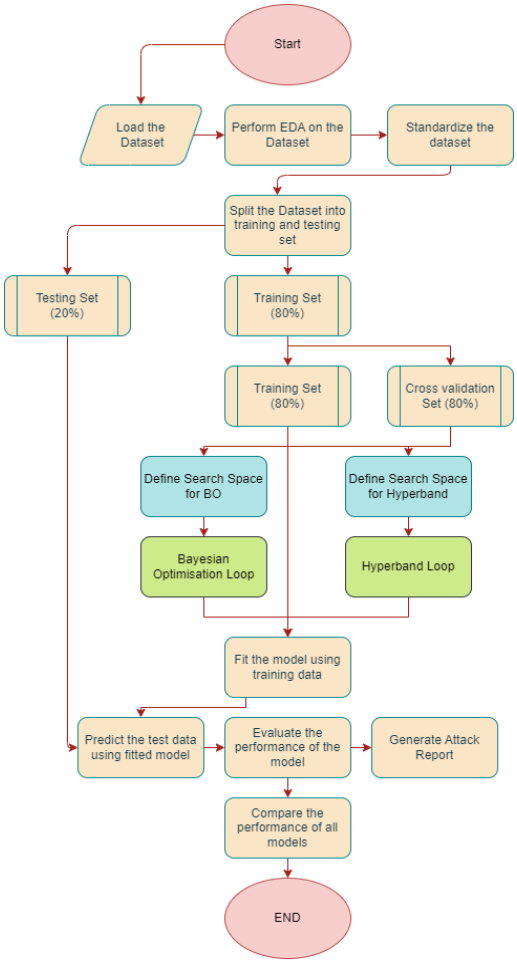


Figure 3: Binary Classification Layer flow

Figure 3 depicts this binary-classification workflow, from raw dataset loading through exploratory data analysis, standardization, the training/testing split, the parallel Bayesian-optimization and Hyperband search-space definitions and optimization loops, model fitting, prediction on the held-out test set, performance evaluation, and comparison across models. Formally, at this tier the model learns a function

$$f_1(x) \rightarrow \{0, 1\} \quad (2)$$

where $f_1(x) = 0$ denotes a prediction of normal traffic and $f_1(x) = 1$ denotes a prediction of an attack instance. The first tier filters out traffic predicted as normal, allowing the second, multi-class tier to focus exclusively on traffic predicted as malicious, which is intended to both improve second-tier accuracy and reduce the false-alarm rate of the overall system.

6.3 Tier 2: Multi-Class Attack Classification Layer

The second tier processes only the traffic instances that the first tier classifies as an attack. Its goal is to classify each such instance into a specific attack category, based on the *attack_cat* label of the UNSW-NB15 dataset. As in the first tier, both Bayesian optimization and

Hyperband are used to tune the hyperparameters of a second, independent XGBoost model for this multi-class task. Formally, this tier learns the mapping

$$f_2(x \mid f_1(x) = 1) \rightarrow \{Attack_1, Attack_2, ..., Attack_n\} \quad (3)$$

where each $Attack_i$ denotes one specific attack category. Model performance is evaluated using the same set of metrics as the first tier. This staged design is intended both to reduce the false-positive burden carried by the first tier and to provide fine-grained attack categorization at the second tier, which is intended to support more informed, adaptive response by a downstream security analyst or automated response system. The corresponding workflow, illustrated in figure 4, mirrors that of figure 3 but is applied only to the attack-labeled subset of the data, using the attack category as the target variable, and additionally reports each predicted attack category's support in descending order and compares the tuned XGBoost models against the logistic-regression and random-forest baselines.

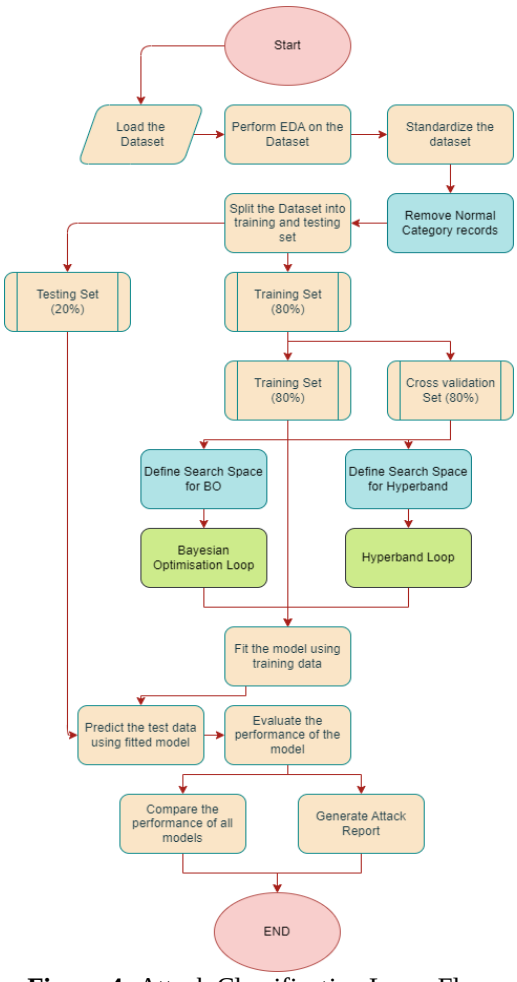


Figure 4: Attack Classification Layer Flow

Figure 4 illustrates the attack-classification workflow: after the attack-only subset is constructed following the same preprocessing, standardization, and splitting steps as figure 3, with the additional step of removing normal-category records before the split, the same parallel Bayesian-optimization and Hyperband search procedure is applied, the resulting model is fit and evaluated on the held-out attack-only test set, and an attack report is generated before all models' performances are compared.

6.4 Hybrid Ensemble Learning: Stacking and Soft Voting

To further improve classification robustness and reduce false alarms beyond what either individually tuned XGBoost model achieves, the proposed framework combines the tuned Bayesian- and Hyperband-optimized XGBoost classifiers, together with the baseline classifiers, into two hybrid ensembles: a stacking ensemble and a soft-voting ensemble.

The stacking ensemble combines the predictions of multiple base classifiers through a learned meta-model:

$$y = \Sigma_i \, w_i \, f_i(x) \quad (4)$$

where $f_i(x)$ denotes the prediction (or predicted probability) produced by base classifier i, and w_i denotes the corresponding weight learned by the meta-learner. Stacking is intended to reduce both bias and variance relative to any single constituent classifier by allowing the meta-learner to learn how to combine the base classifiers' outputs rather than combining them with a fixed rule.

The soft-voting ensemble instead combines the base classifiers' predicted class probabilities using a fixed averaging rule rather than a learned meta-model:

$$\hat{y} = arg\ max_c \ (1/M) \ \Sigma_{m=1}^{M} P_m(c \mid x) \quad (5)$$

where M is the number of base classifiers in the ensemble, $P_m(c \mid x)$ is the probability that base classifier m assigns to class c for input x, and $\hat{y}$ is the ensemble's final predicted class.

6.5 Explainability via SHAP

To support interpretation of the trained models' decisions, the framework incorporates SHapley Additive exPlanations (SHAP) analysis, which attributes each individual prediction to the contribution of each input feature based on cooperative game theory. SHAP values are used here purely for post-hoc interpretation of the tuned XGBoost models and do not affect model training. The resulting SHAP summary plot on figure 10 and the features it identifies as most influential.

6.6. XGBoost Model

XGBoost is used as the base classifier for both tiers of the proposed framework. At iteration t of training, the model's regularized learning objective is

$$Obj(\theta) = \Sigma_i\, l(y_i, \hat{y}_i^{(t)}) + \Sigma_k\, \Omega(f_k) \quad (6)$$

where $l(y_i, \hat{y}_i^{(t)})$ is a differentiable loss function measuring the discrepancy between the true label y_i and the model's current prediction $\hat{y}_i^{(t)}$ for instance i (the binary logistic loss for tier 1, and the multi-class softmax loss for tier 2, and $\Omega(f_k)$ is a regularization term applied to each individual tree f_k in the ensemble, defined as

$$\Omega(f_k) = \gamma T + (1/2)\lambda \|w\|^2 \quad (7)$$

where T is the number of leaves in tree f_k, w is the vector of leaf weights, γ is a penalty coefficient on the number of leaves (controlling tree complexity), and λ is an L_2 regularization coefficient on the leaf weights (controlling weight magnitude). At each boosting round, XGBoost greedily adds the tree f_k that most reduces this objective, using a second-order (Newton) approximation of the loss for efficient split finding.

The hyperparameters that govern this process, and that are the target of the Bayesian and Hyperband optimization procedures include the learning rate (shrinkage applied to each tree's contribution), the maximum tree depth, the number of estimators (boosting rounds), the row-subsampling fraction, the column-subsampling fraction, the minimum child weight, and the minimum loss reduction required to make a further split (commonly denoted γ, consistent with equation 7). The final, selected values of these hyperparameters for each tier and each optimization method are not reported in this study.

7. BAYESIAN HYPERPARAMETER OPTIMIZATION

7.1 Formal Description

Bayesian optimization treats the hyperparameter-tuning problem as a black-box optimization problem in which each evaluation of $f(x)$, here the cross-validated performance of an XGBoost model trained with hyperparameter configuration x, is assumed to be expensive. A probabilistic surrogate model, most commonly a Gaussian process, is fit to the configurations evaluated so far and used to predict, for any untried configuration x, both an expected performance $\mu(x)$ and an uncertainty $\sigma(x)$. An acquisition function then scores candidate configurations by combining these two quantities to balance exploitation and exploration a common choice is expected improvement,

$$EI(x) = E[max(0, f(x) - f(x^+))] \quad (8)$$

where $f(x^+)$ is the best objective value observed among the configurations evaluated so far.

The Hyperband optimization was configured with the following parameters during experimentation as shown in table 1.

Table 1. Hyperband configuration details

Hyperband Parameter	Value
n_iterations	3
n_required_iterations	3
n_possible_iterations	13
min_resources	28
max_resources	203994
aggressive_elimination	False
factor	2

At each iteration, the configuration that maximizes the acquisition function is selected, evaluated by training and validating the XGBoost model, and the surrogate model is updated with the new observation; this cycle repeats until a stopping criterion reached.

7.2 Application to the Proposed Framework

Bayesian optimization is applied independently to the tier 1 (binary) and tier 2 (multi-class) using cross-validated performance on the training/validation split.

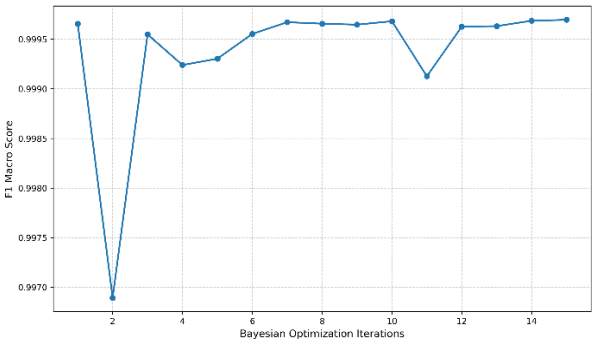


Figure 5. Bayesian Optimization Convergence for XGBoost

It shows the resulting convergence behavior for the XGBoost model over successive Bayesian-optimization iterations, tracking the cross-validated macro F1-score achieved as the search progresses the curve rises from approximately 0.9996 at iteration 1, with two visible dips (to approximately 0.9969 at iteration 2 and to approximately 0.9991 at iteration 11), before recovering and stabilizing near 0.9997 for the final iterations. This indicates that the surrogate-guided search located and then maintained a strong region of the hyperparameter space, with occasional exploratory evaluations of less favorable configurations consistent with the exploration term in the acquisition function.

8. HYPERBAND OPTIMIZATION

8.1 Formal Description

Hyperband allocates a resource budget across brackets of candidate configurations using a successive-halving update rule. Within a bracket, the number of configurations retained at round i is

$$n_i = \lfloor (n/(s+1)) \cdot \eta^i \rfloor \quad (9)$$

where n is the total number of configurations initially sampled for the bracket, s is the maximum number of successive-halving rounds for that bracket, η is the down sampling (elimination) rate, and n_i is the number of configurations retained after round i. At each round, every surviving configuration is trained for an increased resource allocation (for example, additional boosting rounds), evaluated, and the worst-performing fraction (determined by η) is discarded before the next round this differs from Bayesian optimization in that Hyperband does not use a probabilistic surrogate model to select which configurations to try, relying instead on random sampling combined with early stopping to allocate resources efficiently.

8.2 Application to the Proposed Framework

Hyperband is applied independently to the tier 1 and tier 2 XGBoost models, using the same cross-validated objective as the Bayesian-optimization procedure, so that the two search strategies can be compared directly on the same classifier and task.

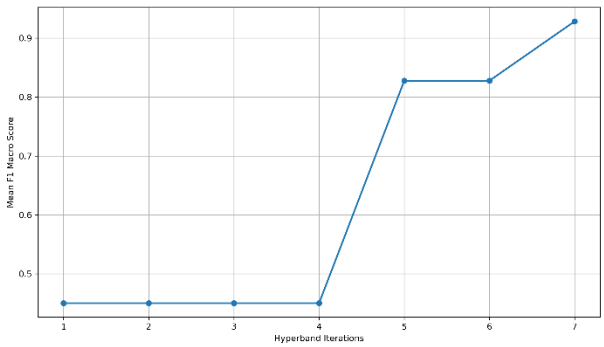


Figure 6. Hyperband Optimization Convergence

Figure 6 shows the resulting Hyperband convergence curve, tracking the mean macro F1-score achieved at each successive Hyperband iteration: the score remains flat at approximately 0.45 for iterations 1 through 4, then rises sharply to approximately 0.83 at iteration 5, remains at that level through iteration 6, and rises again to approximately 0.93 by iteration 7. This pattern of successive improvement steps is consistent with the successive-halving mechanism in equation 9 many configurations are evaluated at low resource allocation levels during early iterations, most perform poorly, and later, more successful, iterations evaluate only the smaller numbers of surviving configurations at significantly higher resource allocations, producing the visible performance jumps.

8.3 Hyperband Configuration

Table 2 reports the Hyperband configuration used during experimentation.

Table 2. Hyperband Configuration Details

Hyperband Parameter	Value
n_iterations	3
n_required_iterations	3
n_possible_iterations	13
min_resources	28
max_resources	203994
aggressive_elimination	False
factor	2

These settings correspond to a downsampling rate of $\eta = 2$ in equation 9 (the "factor" row) and reflect a resource allocation intended to balance exploration of the hyperparameter search space against computational efficiency through adaptive early stopping. The specific hyperparameter ranges over which Hyperband's resource allocation was applied, and the final configuration it selected, are not reported in this study; this gap parallels the corresponding gap.

9. EXPERIMENTAL SETUP

9.1 Datasets

The proposed framework is evaluated on two benchmark cybersecurity datasets, UNSW-NB15 and CICIDS2017. Evaluating the framework on two datasets with different feature schemas and attack taxonomies is intended to assess generalization across network-traffic environments and attack patterns, rather than tuning the implementation to the idiosyncrasies of a single benchmark. A random subset of 50,000 records was sampled from each dataset for the reported experiments. Two internal inconsistencies in the reporting of dataset characteristics were identified during preparation of this manuscript and are recorded here rather than resolved silently.

First, one part of the manuscript states that UNSW-NB15 used 20 features and CICIDS2017 used 53 features, while table 3 reports final feature counts of 35 for UNSW-NB15 and 23 for CICIDS2017 reflects the detailed feature-engineering procedure and is internally consistent with the correlation-based reduction described the counts in table 3 (UNSW-NB15: 49 → 35; CICIDS2017: 53 → 23) are treated as authoritative in this manuscript; the earlier "20 features" figure could not be reconciled.

Second, it is not stated unambiguously which dataset, or combination of datasets, the headline results in table 5 describe. The confusion-matrix figure is explicitly captioned Hybrid (BH) Voting CICIDS2017, indicating that at least this figure reports CICIDS2017-specific, multi-class (tier 2) results; the class totals visible in figure 9 (11 + 50 + 767 + 1,120 + 12,475 + 537 + 11 ≈ 14,971 instances across seven classes) are consistent with a single-dataset evaluation rather than the combined 50,000-record sample.

9.2 Data Preprocessing

Prior to model training, both datasets were filtered and normalized class-imbalance correction was applied using a "safe" variant of the Synthetic Minority Oversampling Technique (Safe-SMOTE), consistent with the overall pipeline. The specific Safe-SMOTE parameters used, such as the oversampling ratio and the number of nearest neighbors, are not reported in this study and are listed as an item requiring author verification.

9.3 Feature Engineering and Class-Imbalance Handling

Feature engineering was performed through leakage-feature removal and correlation-based dimensionality reduction. Features such as "Flow Bytes/s," "Flow Packets/s," and "Packet Length Variance" were removed to avoid data leakage and unrealistic prediction bias, and attributes with pairwise correlation above a threshold of 0.75 were removed to reduce redundant information and computational overhead. table 3 reports the resulting feature counts for both datasets.

Table 3. Dataset Feature Engineering Summary

Dataset	Original Features	Removed Features	Final Features
UNSW-NB15	49	14	35
CICIDS2017	53	30	23

The correlation-based reduction removed a substantially larger fraction of CICIDS2017's original features (30 of 53, or 57%) than of UNSW-NB15's (14 of 49, or 29%), which is consistent with CICIDS2017's flow-level feature set containing more mutually redundant traffic statistics than UNSW-NB15's more curated feature set.

9.4 Validation Procedure and Baseline Models

Model performance is evaluated using a held-out train/test split. The exact split ratio, the random seed used, whether stratification was applied to preserve class proportions, and whether k-fold cross-validation was used for model selection in addition to the final held-out evaluation are not reported beyond the general statement that "suitable validation methods such as k-fold cross-validation or holdout validation" were available options. These details, together with confirmation of whether the final test set was held out independently of the cross-validation folds used during hyperparameter search.

The proposed Bayesian- and Hyperband-tuned XGBoost models, and the resulting hybrid stacking and soft-voting ensembles are compared against six baseline classifiers: naive Bayes (NB), k-nearest neighbors (KNN), logistic regression (LR), random forest (RF), unoptimized XGBoost (XGB), and LightGBM. LR is included as a computationally lightweight, interpretable linear baseline, though it is not well suited to the non-linear decision boundaries typical of network-attack patterns; random forest is included as a bagging-based ensemble baseline that is robust to overfitting but may not capture the same subtle feature interactions as boosting-based methods. All baseline and proposed models are evaluated using the same train/test partition and the same metrics to keep the comparison controlled.

9.5 Evaluation Metrics

The framework is evaluated using accuracy, precision, recall, F1-score, the false alarm rate, and, where reported, the area under the ROC curve (AUC), Matthews correlation coefficient (MCC), and Cohen's kappa. These metrics are defined formally below in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), computed per class and, for multi-class evaluation aggregated as reported in the corresponding results.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (10)$$

$$Precision = TP/(TP + FP) \quad (11)$$

$$Recall = TP/(TP + FN) \quad (12)$$

$$F_1 = 2 \times (Precision \times Recall)/(Precision + Recall) \quad (13)$$

$$FAR = FP/(FP + TN) \quad (14)$$

$$MCC = (TP \times TN - FP \times FN)/\sqrt{((TP+FP)(TP+FN)(TN+FP)(TN+FN))} \quad (15)$$

$$\kappa = (p_o - p_e)/(1 - p_e) \quad (16)$$

where, in equation 16, p_o is the observed proportion of agreement between predicted and true labels and p_e is the proportion of agreement expected by chance given the observed class marginals. Accuracy alone is an insufficient metric under the class imbalance present in both datasets since a classifier that always predicts the majority class can achieve high accuracy while failing to detect minority attack classes precision, recall, F1-score, MCC, and the false alarm rate are therefore reported alongside accuracy throughout consistent with standard practice for imbalanced intrusion-detection evaluation. The confusion matrix additionally reports class-wise counts directly, which is the most granular

view of per-class performance available from the reported experiments.

9.6 Hardware, Software, and Reproducibility

The experiments were implemented in Python using the Anaconda distribution, with pandas, NumPy, and scikit-learn as the primary libraries, and were executed on a Windows-based system with an Intel Core CPU and 8 GB of RAM. Table 4 summarizes the reproducibility-relevant details reported in this study, together with details that are not reported and are therefore listed as items for author verification rather than assumed.

Table 4. Reproducibility Summary

Category	Parameter/Setting	Value
Dataset	Names	UNSW-NB15, CICIDS2017
	Sampled records used	50,000
	Final feature counts	UNSW-NB15: 35; CICIDS2017: 23
Model	Algorithm	XGBoost
	Objective	Binary logistic, multi-class softmax
Optimization	Bayesian search space and selected values	Not reported
	Hyperband configuration	n_iterations = 3; min_resources = 28; max_resources = 203994; factor = 2
Class balancing	Method	Safe-SMOTE; parameters not reported
Validation	Cross-validation method	Not reported
Software	Python distribution and version	Anaconda; exact Python/pandas/NumPy/scikit-learn
Hardware	CPU/GPU/RAM	Intel Core CPU, 8 GB RAM
Randomness	Random seed(s)	Not reported
Repetitions	Number of independent runs	The reported convergence curves show a single optimization run per method

10. RESULTS

10.1 Overview

This section reports the results of the experiments described the performance of the Bayesian- and Hyperband-tuned XGBoost classifiers, the hybrid stacking and soft-voting ensembles, and the six baseline classifiers followed by the ROC-AUC comparison, the confusion matrix for the best-performing model, the SHAP-based feature-importance analysis, and a comparison against results reported in the literature reviewed.

10.2 Final Model Performance Comparison

Table 5 reports accuracy, precision, recall, and F1-score for all nine evaluated models: six conventional baselines (naive Bayes, k-nearest neighbors, logistic regression, random forest, unoptimized XGBoost, and LightGBM), the Bayesian and Hyperband tuned XGBoost model and the two hybrid ensembles built over the tuned base classifiers

Table 5. Final Model Performance Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Hybrid (BH) Voting	99.81	96.36	94.32	95.27
Hybrid (BH) Stacking	99.80	96.35	94.04	95.13
XGB (BH)	99.79	95.65	95.27	95.46
RF	99.75	96.42	93.43	94.87
LightGBM	99.74	94.59	94.97	94.75
XGB	99.71	90.96	95.25	92.83
KNN	98.25	74.86	93.57	79.18
NB	78.97	58.02	92.05	61.27
LR	77.95	46.65	93.35	50.39

Table 5 shows that the two hybrid ensembles and the tuned XGB (BH) model form a clearly separated top tier, all above 99.7% accuracy, ahead of RF and LightGBM (both above 99.7%), which in turn are well ahead of KNN (98.25%) and substantially ahead of NB and LR (both below 80%). Notably, XGB (BH)'s recall (95.27%) and F1-score (95.46%) are the highest of any single model, marginally exceeding both hybrid ensembles on these two metrics even though its accuracy (99.79%) and precision (95.65%) are slightly lower than the two ensembles' this indicates that, in the reported experiment, ensembling improved precision and overall accuracy but did not uniformly improve every metric relative to the single best-tuned XGBoost model a nuance not evident from the headline accuracy figure alone. NB and LR both combine high recall (92.05% and 93.35%, respectively) with markedly low precision (58.02% and 46.65%), indicating that both models substantially over-predict the positive/attack class, which is consistent with their low overall accuracy and F1-score.

Figure 7 presents these same results graphically across for all nine models, and visually confirms the same ranking pattern described above the hybrid and boosted-tree models cluster tightly near the top of each metric, while NB and LR show visibly lower and more variable bars across metrics, consistent with their weaker precision-recall balance.

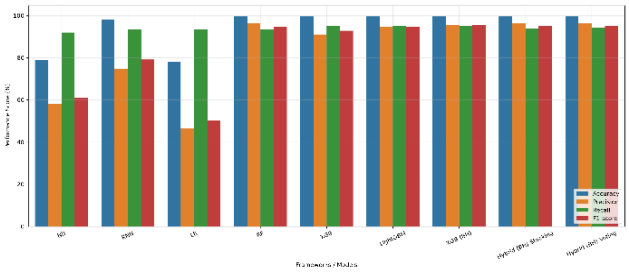


Figure 7: Comparative Performance Analysis of IDS Models

10.3 ROC-AUC Analysis

Figure 8 compares the receiver operating characteristic ROC curves of all evaluated models on a single plot, with the false-positive rate on the horizontal axis and the true-positive rate on the vertical axis; the dashed diagonal represents the no-skill (random-guessing) baseline. As read from the figure's legend, the approximate AUC values are NB ≈ 0.973, KNN ≈ 0.973, LR ≈ 0.972, RF ≈ 0.990, XGB ≈ 0.998, LightGBM ≈ 0.994, XGB (BH) ≈ 0.999, Hybrid (BH) Stacking ≈ 0.995, and Hybrid (BH) Voting ≈ 0.999. These values are broadly, though not perfectly, consistent with the accuracy ranking in table 5 the boosted-tree and hybrid models again cluster at the top (AUC ≥ 0.99), while NB, KNN, and LR cluster lower (AUC ≈ 0.97), even though NB and LR's accuracy in Table 5 (below 80%) is far lower than their AUC alone would suggest, illustrating that AUC and accuracy can diverge substantially under class imbalance because AUC evaluates ranking quality across all classification thresholds rather than performance at a single, fixed decision threshold. The best-performing hybrid model's AUC is reported at two different levels of precision in the underlying source material, approximately 0.9996 and approximately 0.9999, both are consistent with the three-decimal legend values readable from figure 8, which cannot distinguish between them at that resolution. This manuscript therefore reports the value only as app. 0.999 pending confirmation; the exact AUC value, to at least four decimal places, is listed in the Author Verification Report for confirmation against the underlying evaluation script or logs.

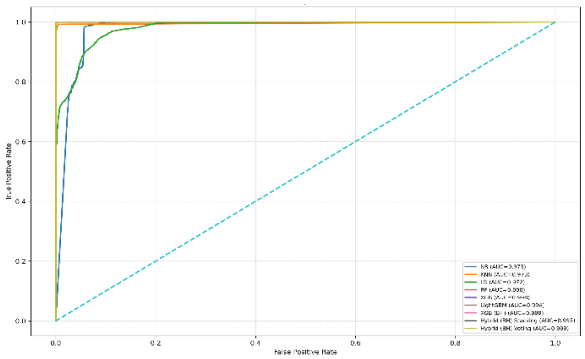


Figure 8. ROC-AUC Comparison of All Models

10.4 Confusion Matrix Analysis

Figure 9 shows the confusion matrix of the best-performing model, explicitly captioned "Hybrid (BH) Voting" and, per the figure's own caption, evaluated on CICIDS2017. The matrix is seven-by-seven, consistent with a seven-category multi-class evaluation. The diagonal entries dominate every row, confirming that most instances are correctly classified for example, class 4 accounts for 12,475 of the roughly 14,971 instances shown in the figure, with only 14 of its instances misclassified across the other six classes (1 to class 0, 5 to class 3, 6 to class 5, 2 to class 6), a per-class error rate of approximately 0.11%. The smallest classes (class 0, with 12 instances, and class 6, with 15 instances) show proportionally larger relative error, with 1 of 12 class-0 instances and 4 of 15 class-6 instances misclassified, illustrating that even a model with very high aggregate accuracy can have a substantially higher relative error rate on its rarest classes, which is precisely the reason precision, recall, and F1-score are reported alongside accuracy in table 5 rather than accuracy alone. Because figure 9 is explicitly scoped to CICIDS2017, it corroborates, but does not fully resolve, the dataset-scope ambiguity it confirms that at least the confusion-matrix analysis of the "best model" is CICIDS2017-specific, which is consistent with, though not conclusive proof of, table 5 also being CICIDS2017-specific rather than a genuinely combined evaluation across both datasets.

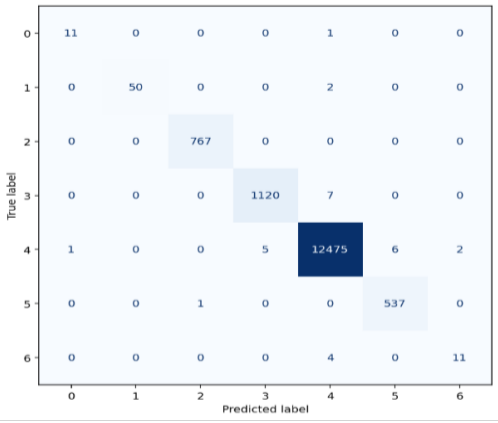


Figure 9. Confusion Matrix of the Hybrid (BH) Voting Model (CICIDS2017)

10.5 SHAP-Based Feature Importance

Figure 10 presents the SHAP summary plot for the proposed framework's best-performing classifier, in which each row corresponds to one input feature and each point corresponds to one instance's SHAP value for that feature, with color indicating whether that instance's value for the feature was relatively high or low. The plot shows a clear separation between a small number of high-impact features, whose SHAP values span roughly -4 to +4 along the horizontal axis, and the remaining features, whose SHAP values cluster much more tightly

around zero, indicating that a small subset of network-traffic features drives the majority of the model's classification decisions. The specific feature names corresponding to each row are not legible in the exported figure and are not listed elsewhere in the manuscript. Regenerating figure 10 with feature-name labels on the vertical axis and a color-scale legend indicating low-to-high feature value would be required for the figure to be fully interpretable to a reader and for the SHAP analysis to support the interpretability claims.

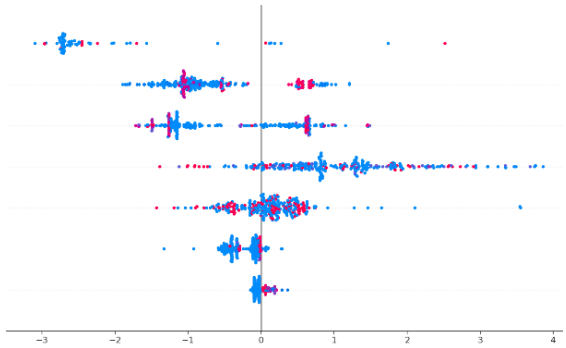


Figure 10. SHAP Summary Plot

10.6 Comparison with State-of-the-Art Methods

Table 6 compares the proposed framework's best result against three results reported in the literature reviewed in for methods evaluated on related but not identical datasets and tasks.

Table 6. Comparison with State-of-the-Art Methods

Study	Dataset	Method	Accuracy (%)	F1-score (%)
Alrabanah et al. (2025)	Standardized network-traffic anomaly dataset	Hybrid LSTM-XGBoost (CNN-LSTM-style sequential + boosted-tree hybrid)	98.70	94.10
Kodete et al. (2025)	UNSW-NB15	TripleBoost (AdaBoost + XGBoost ensemble)	97.38	96.40
Hosain and Çakmak (2025)	WUSTL-EHMS-2020	XAI-XGBoost (RFE + tuned XGBoost with SHAP/LIME)	99.22	99.12
Proposed Hybrid (BH) Voting	UNSW-NB15 + CICIDS2017	Two-tier XGBoost, Bayesian- and Hyperband-tuned, soft-voting ensemble	99.81	95.27

The three literature rows in table 6 report results reproduced from the cited studies themselves, not results obtained in the present study; they are included only for indicative comparison against results reported by other authors, on datasets and evaluation protocols that are not identical to the present study's, and should not be read as a like-for-like benchmark. The proposed framework's accuracy (99.81%) is the highest of the four rows shown, while its F1-score (95.27%) is the lowest of the four; the two literature studies with a higher reported F1-score than the proposed framework (Kodete et al., 2025, at 96.40%, and Hosain & Çakmak, 2025, at 99.12%) were evaluated on UNSW-NB15 alone and on WUSTL-EHMS-2020, respectively, neither of which matches the dataset scope of the proposed framework's own evaluation, so this difference cannot be attributed with confidence to either the two-tier architecture or the dual-optimization strategy without a controlled, same-dataset re-evaluation. A stronger, same-dataset state-of-the-art comparison would evaluate at least one of TripleBoost or XAI-XGBoost directly on the same UNSW-NB15/CICIDS2017 sample used.

11. DISCUSSION

11.1 Principal Findings

The central empirical finding is that the Hybrid (BH) Voting ensemble achieved the highest accuracy (99.81%) and F1-score (95.27%) among the nine models compared in Table 5, with the individually tuned XGB (BH) model close behind and, on recall and F1-score specifically, marginally ahead of both hybrid ensembles. Both the tuned XGBoost model and the two hybrid ensembles clearly outperformed the conventional baselines (NB, KNN, LR), and modestly outperformed RF and LightGBM, which is consistent with the manuscript's central claim that Bayesian- and Hyperband-driven hyperparameter tuning, combined with hybrid ensembling, improves detection performance relative to untuned or singly-tuned baselines on the same data.

11.2 Model Behavior

The gap between the top-tier models (Hybrid (BH) Voting, Hybrid (BH) Stacking, XGB (BH), all above 99.7% accuracy) and the weakest baselines (NB, LR, both below 80% accuracy) is large, and Table 5 shows this gap is driven primarily by precision: NB and LR both retain relatively high recall (92% to 93%) but very low precision (46% to 58%), indicating they substantially over-predict the attack class. This behavior is consistent with NB's conditional-independence assumption and LR's linear decision boundary both being poorly suited to the correlated, non-linear feature interactions typical of network-flow data whereas the boosted-tree models (XGB, LightGBM, XGB (BH)) can represent such interactions directly through tree splits.

11.3 Two-Tier Architecture

The two-tier design is intended to let the binary and multi-class decisions be optimized independently; the

reported results do not, however, separately report tier 1 (binary) accuracy and tier 2 (multi-class) accuracy, since table 5 and figure 7 report only the end-to-end, apparently tier-2-scoped metrics. This means the specific benefit of separating the two tiers, relative to a single flat multi-class classifier trained directly on all traffic, cannot currently be quantified from the reported results, even though the architectural rationale for separating them remains sound.

11.4 Bayesian and Hyperband Optimization

Figures 5 and 6 show qualitatively different convergence behavior for the two optimization strategies on the same underlying classifier, Hyperband's macro-F1 curve is flat through its first four iterations and then rises in two large steps, consistent with successive halving concentrating resources on a shrinking set of survivors, while Bayesian optimization's curve starts near its final level and shows two isolated dips before recovering, consistent with a surrogate-guided search that spends a minority of its iterations on exploratory configurations. Because the manuscript does not report the wall-clock time or number of trials each method consumed to reach its respective plateau a qualitative comparison of the two methods' search dynamics but not yet a quantitative comparison of their computational efficiency for the same classifier and task.

11.5 State-of-the-Art Comparison

As discussed, the proposed framework's accuracy is the highest among the four rows in table 6, but its F1-score is the lowest, and the three literature comparators were evaluated on different datasets. This pattern of results highest accuracy, lowest F1-score, non-identical evaluation conditions means the state-of-the-art comparison in table 6 should be read as indicative context for where the proposed framework's reported metrics fall relative to related recent work, rather than as evidence of outright superiority over any specific prior method.

11.6 Computational Efficiency

The manuscript reports Hyperband's bracket configuration in full but does not report the corresponding training or inference time for any of the nine compared models, nor the wall-clock cost of the Bayesian-optimization and Hyperband search procedures themselves. Because computational efficiency, specifically the cost of stacking/ensembling and deep-hybrid components, was identified as a recurring limitation of prior IDS studies, and because efficient hyperparameter search was one of this study's central motivations, the absence of reported training and inference time, and of search wall-clock cost, is a meaningful gap. Reporting per-model training time, per-instance inference time, and total Bayesian-optimization and Hyperband search time would allow the framework's efficiency claims to be evaluated quantitatively rather than only architecturally.

11.7 Security Implications

A two-tier IDS that filters normal traffic before performing more detailed attack-category analysis is intended to concentrate the more expensive fine-grained classification step, and the SHAP-based interpretability layer on the smaller subset of traffic that most needs an analyst's attention. In an operational security setting, the confusion matrix in figure 9 suggests this design is workable in principle for a majority-class attack (class 4, with a per-class error rate near 0.11%), but the higher relative error observed on the smallest classes in the same figure indicates that rare attack types would likely still require additional analyst review or a dedicated rare-class detection mechanism, rather than being fully resolved by the two-tier design alone.

11.8 Unexpected Results

Two results are worth noting explicitly because they are not fully accounted for by the framework's overall narrative of ensembling as the best-performing configuration. First, the individually tuned XGB (BH) model outperforms both hybrid ensembles on recall and F1-score even though the two hybrid ensembles achieve higher accuracy and precision; ensembling therefore improved accuracy and precision in this experiment but did not uniformly improve every evaluation metric. Second, the gap between the best individual model (XGB (BH)) and the best ensemble (Hybrid (BH) Voting) is small across all four metrics (within 0.5 percentage points), which, combined with the single-run design noted in means the practical benefit of ensembling over the best single tuned classifier in this experiment should be interpreted cautiously pending repeated-run evaluation.

11.9 Practical Implications

The results in table 5 suggest that, on the data reported, tuned or ensembled boosted-tree models offer a substantially better precision-recall balance than the linear and low-capacity baselines (NB, LR), which is relevant to any deployment context where false alarms (low precision) carry an operational cost, such as analyst alert fatigue, false-alarm-rate metric is directly tied to precision. Practically deploying the proposed framework would additionally require resolving the reproducibility gaps identified in table 4 and the dataset-scope ambiguity identified, since an operator would need to know precisely which traffic distribution the reported 99.81% accuracy figure actually describes before relying on it in a different network environment.

12. FUTURE WORK

Building directly on the gaps identified above, future work should:

- The tier 1 and tier 2 performance separately, together with a single flat-classifier baseline, to directly quantify the two-tier architecture's benefit
- The report training time, inference time, and search wall-clock cost for both optimization strategies, to support a quantitative, not only qualitative, comparison of Bayesian optimization and Hyperband
- The repeat the experiment across multiple independent runs with varying random seeds to support the statistical-validation practices expected of this evaluation, since the current results reflect a single run per method
- The evaluate at least one directly comparable prior method on the same dataset sample used in this study, to strengthen the state-of-the-art comparison beyond the cross-dataset comparison currently reported.

13. LIMITATIONS

The following limitations of the reported study are stated explicitly; each is a limitation of what this study reports, not a limitation introduced during preparation of this manuscript.

- All results reflect a single train/test split and a single optimization run per method no repeated-run statistics, confidence intervals, or significance tests are reported, so the observed differences between the top-performing models (Hybrid (BH) Voting, Hybrid (BH) Stacking, XGB (BH)) cannot currently be assessed for statistical significance.
- The statement elsewhere that UNSW-NB15 used 20 features is inconsistent reported 35 final UNSW-NB15 features and it is not established which figure reflects the features actually used for model training.
- The specific search ranges and final selected hyperparameter values for both the Bayesian-optimization and Hyperband procedures are not reported which limits reproducibility and prevents a quantitative comparison of the two optimization strategies' efficiency.
- The comparison is against results reported by other authors on different datasets and evaluation protocols, not a controlled, same-dataset re-evaluation, which limits how directly the proposed framework's results can be compared to the cited prior methods.

14. CONCLUSION

In this work a two-tier intrusion detection framework in which a first-tier binary XGBoost classifier separates normal from malicious network traffic, and a second-tier multi-class XGBoost classifier categorizes the resulting malicious traffic into specific attack types, with both tiers independently tuned using Bayesian optimization and Hyperband and combined, together with baseline classifiers, into hybrid stacking and soft-voting ensembles. On the experiments reported the Hybrid (BH) Voting ensemble achieved the highest overall accuracy (99.81%) and F1-score (95.27%) among the nine models compared, ahead of conventional baselines (naive Bayes, k-nearest neighbors, logistic regression, random forest) and of unoptimized XGBoost and LightGBM, with the individually tuned XGB (BH) model achieving the highest recall and a comparable F1-score. SHAP-based explainability was integrated to support interpretation of the model's classification decisions, and a false-alarm-rate metric was defined to support evaluation of the framework's practical suitability for operational deployment.

References:

Ajagbe, S. A., Awotunde, J. B., & Florez, H. (2024). Intrusion detection: A comparison study of machine learning models using unbalanced dataset. *SN Computer Science, 5*, 1028. https://doi.org/10.1007/s42979-024-03369-0

Al Amien, J., Ab Ghani, H., Md Saleh, N. I., Ismanto, E., & Gunawan, R. (2023). Intrusion detection system for imbalance ratio class using weighted XGBoost classifier. *TELKOMNIKA (Telecommunication Computing Electronics and Control), 21*(5), 1102–1112. https://doi.org/10.12928/TELKOMNIKA.v21i5.24735

Alashjaee, A. M., & Alqahtani, F. (2025). Enhanced intrusion detection system IoT network security model by feed forward neural network and machine learning. *Scientific Reports, 15*, 36085. https://doi.org/10.1038/s41598-025-20047-0

Alrabanah, Y., Al-Sharaeh, S., & Al Hindi, G. (2025). Enhancing intrusion detection using hybrid long short-term memory and XGBoost. *Journal of Soft Computing and Data Mining, 6*(1), 247–261. https://doi.org/10.30880/jscdm.2025.06.01.016

Alsubaei, F. S. (2025). Smart deep learning model for enhanced IoT intrusion detection. *Scientific Reports, 15*, 20577. https://doi.org/10.1038/s41598-025-06363-5

Bibers, I., Arreche, O., Alayed, W., & Abdallah, M. (2025). Ensemble-IDS: An ensemble learning framework for enhancing AI-based network intrusion detection tasks. *Applied Sciences, 15*, 10579. https://doi.org/10.3390/app151910579

Binsaeed, K. A., & Hafez, A. M. (2023). Enhancing intrusion detection systems with XGBoost feature selection and deep learning approaches. *International Journal of Advanced Computer Science and Applications, 14*(5), 1084–1098. https://doi.org/10.14569/IJACSA.2023.01405112

Cavus, M., Ayan, H., Sari, M., Akbulut, O., Dissanayake, D., & Bell, M. (2025). Enhancing smart grid reliability through data-driven optimisation and cyber-resilient EV integration. *Energies, 18*, 4510. https://doi.org/10.3390/en18174510

Chimphlee, S., & Chimphlee, W. (2023). Machine learning to improve the performance of anomaly-based network intrusion detection in big data. *Indonesian Journal of Electrical Engineering and Computer Science, 30*(2), 1106–1119. https://doi.org/10.11591/ijeecs.v30.i2.pp1106-1119

Faizin, M. A., Kurniasari, D. T., Elqolby, N., Putra, M. A. R., & Ahmad, T. (2024). Optimizing feature selection method in intrusion detection system using thresholding. *International Journal of Intelligent Engineering and Systems, 17*(3), 214–226. https://doi.org/10.22266/ijies2024.0630.18

Gupta, B. B., Gaurav, A., & Chui, K. T. (2023). Convolution neural network (CNN) based phishing attack detection model for e-business in enterprise information systems. In *Proceedings of ICEB 2023* (Chiayi, Taiwan). Retrieved from https://aisel.aisnet.org/iceb2023/71

Gupta, B. B., Gaurav, A., Arya, V., Bansal, S., Attar, R. W., Alhomoud, A., & Psannis, K. E. (2025). Earthworm optimization algorithm based cascade LSTM-GRU model for Android malware detection. *Cyber Security and Applications, 3*, Article 100083. https://doi.org/10.1016/j.csa.2024.100083

Hermosilla, P., Berríos, S., & Allende-Cid, H. (2025). Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models. *Applied Sciences, 15*, 7329. https://doi.org/10.3390/app15137329

Hosain, Y., & Çakmak, M. (2025). XAI-XGBoost: An innovative explainable intrusion detection approach for securing internet of medical things systems. *Scientific Reports, 15*, 22278. https://doi.org/10.1038/s41598-025-07790-0

Kandasamy, V., & Roseline, A. A. (2025). Harnessing advanced hybrid deep learning model for real-time detection and prevention of man-in-the-middle cyber attacks. *Scientific Reports, 15*, 1697. https://doi.org/10.1038/s41598-025-85547-5

Kareem, M. K., Aborisade, O. D., Onashoga, S. A., Sutikno, T., & Olayiwola, O. M. (2023). Efficient model for detecting application layer distributed denial of service attacks. *Bulletin of Electrical Engineering and Informatics, 12*(1), 441–450. https://doi.org/10.11591/eei.v12i1.3871

Karthikeyan, M., Manimegalai, D., & RajaGopal, K. (2024). Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection. *Scientific Reports, 14*, 231. https://doi.org/10.1038/s41598-023-50554-x

Kodete, C. S., Raju, K. B., Karmakonda, K., Sikindar, S., Ramesh, J. V. N., & Tirumaladham, N. S. K. M. K. (2025). Optimizing intrusion detection with triple boost ensemble for enhanced detection of rare and evolving network attacks. *International Journal of Electrical and Electronic Engineering and Telecommunications, 14*(3), 115–129. https://doi.org/10.18178/ijeetc.14.3.115-129

Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing, 13*, 123. https://doi.org/10.1186/s13677-024-00685-x

Shree, V. G. A., & Thangaraj, M. (2025). Intelligent autoencoder with LSTM based intrusion detection and recommender system. *International Journal of Computer Network and Information Security, 17*(5), 45–62. https://doi.org/10.5815/ijcnis.2025.05.04

Singh, S. K., Kumar, S., Chhabra, A., Sharma, A., Arya, V., Srinivasan, M., & Gupta, B. B. (2025). Advancements in secure quantum communication and robust key distribution techniques for cybersecurity applications. *Cyber Security and Applications, 3*, Article 100089. https://doi.org/10.1016/j.csa.2025.100089

Girraj Khandelwal
JECRC University,
Jaipur,
India,
girraj13@gmail.com
ORCID 0000-0003-1533-0670

Gajanand Sharma
JECRC University,
Jaipur,
India,
gajanan.sharma@gmail.com
ORCID 0000-0001-6412-0392

Brij Bhooshan Gupta
Asia University
Taiwan,
Taiwan,
gupta.brij@gmail.com
ORCID 0000-0003-4929-4698