Journal of Artificial Intelligence in Information Technology (JAI-IT)
Vol. 1 No. 1 June 2026
E-ISSN: XXXX-XXXX; P-ISSN: XXXX-XXXX, Pages 129-150
DOI: https://doi.org/10.51903/yxzn1163

An Applied Framework for Real-Time Network Intrusion Detection Using Optimized Ensemble Learning in Enterprise IT Environments.

Cici Rossa Natalia*[1], Iman Saufik Suasana S.Kom, M.Kom[1]
Email: cicirossa@stekom.ac.id(1), saufik@stekom.ac.id(2)
Orcid: https://orcid.org/0009-0008-3088-2282(1), https://orcid.org/0009-0008-3815-1812(2)
[1]*Departement of Computer System. Faculty of Academic Study. Universitas Sains dan Teknologi Komputer, Semarang, Indonesia, 50192*
*Corresponding Author

Abstract

Modern enterprise IT environments require proactive, real-time intrusion detection systems to combat increasingly sophisticated and high-speed cyber threats. However, existing machine learning and deep learning models face a critical trade-off between predictive accuracy and inference latency, rendering many computationally heavy frameworks unsuitable for line-rate, high-throughput network streams. To address this practical deployment gap, this study proposes a novel, lightweight applied framework for real-time network intrusion detection using an optimized ensemble learning architecture. The proposed methodology utilizes a stacking ensemble strategy that combines highly efficient gradient boosting base learners with a meta-classifier. A statistical feature selection procedure, comprising variance thresholding and Pearson correlation filtering, is applied to reduce the original 80-dimensional feature space to 23 discriminative features, a step that is fundamental to achieving low inference latency. To minimize computational overhead without sacrificing sensitivity, a Bayesian Optimization algorithm is implemented to dynamically tune the multidimensional hyperparameters. Evaluated against the comprehensive CSE-CIC-IDS2018 dataset, empirical results demonstrate that the proposed framework achieves an outstanding Detection Rate of 99.85% and a minimal False Positive Rate of 0.04%. Crucially, the optimized architecture maintains a sub-millisecond inference latency of 0.92 milliseconds per flow, significantly outperforming traditional Convolutional Neural Networks (CNN) which recorded severe latencies of 4.20 milliseconds alongside lower detection accuracy. Ultimately, this research delivers a deployable, highly accurate architectural solution that successfully overcomes the latency constraints of conventional models, making it highly viable for real-time enterprise security operations.

Keywords: *Network Intrusion Detection; Ensemble Learning; Real-Time Analysis; Bayesian Optimization; Inference Latency; Cybersecurity.*

I. INTRODUCTION

Enterprise information technology (IT) environments have evolved into highly complex ecosystems characterized by cloud computing, Internet of Things (IoT) integration, and massive data throughput. This digital transformation, while driving operational efficiency, has simultaneously expanded the attack surface available to malicious actors [1]. Consequently, network intrusions have become increasingly frequent, sophisticated, and damaging to organizational infrastructure. Traditional security mechanisms, such as signature-based intrusion detection systems (IDS) and conventional firewalls, routinely fail to identify novel or polymorphic attack vectors [2]. The dynamic nature of modern network traffic requires defense mechanisms that can intelligently adapt to emerging threats without relying solely on predefined, rigid rule sets.

Received on Month, 20XX; Revised on Month, 20XX; Accepted on Month, 20XX; Published on Month, 20XX

The current cybersecurity landscape is dominated by advanced persistent threats (APTs) and zero-day exploits that are specifically engineered to bypass conventional perimeter defenses [3]. Machine learning-based detection approaches have attracted growing attention precisely because they can learn attack patterns from data rather than relying on static rule sets [4]. Threat actors continuously automate their attack methodologies, significantly reducing the window of time available for defensive intervention. In an enterprise context, a delayed response to a network breach often translates directly into severe financial losses, regulatory penalties, and irreparable reputational damage. Therefore, the paradigm of network security has shifted fundamentally from post-incident forensic analysis to proactive, real-time threat detection. Achieving this real-time capability demands highly efficient computational models capable of parsing gigabytes of network traffic per second with minimal latency.

To address the limitations of traditional systems, researchers have heavily explored machine learning (ML) and deep learning (DL) techniques for anomaly-based network intrusion detection. Several recent studies have demonstrated the efficacy of standalone algorithms, such as Support Vector Machines, Random Forests, and Convolutional Neural Networks, in identifying anomalous traffic patterns [5], [6]. For instance, deep learning architectures have shown significant promise in automatically extracting complex spatial and temporal features from raw network packets [7], [8]. However, empirical evidence suggests that relying on a single learning algorithm often results in an inherent trade-off between the false positive rate (FPR) and the detection rate (DR) [6]. Furthermore, the substantial computational overhead required to train and deploy complex deep learning models directly contradicts the low-latency requirements of real-time enterprise networks [9].

Consequently, the academic focus has increasingly shifted toward ensemble learning strategies, which combine multiple base classifiers to enhance overall predictive performance and model robustness. Recent literature extensively documents the application of bagging, boosting, and stacking techniques to create heterogeneous intrusion detection frameworks [10], [11], [12]. To further refine these models, various meta-heuristic and evolutionary optimization algorithms are frequently employed to select optimal features and tune model hyperparameters dynamically [13]. Researchers have successfully utilized these optimization techniques to maximize the accuracy of ensemble configurations while attempting to control resource consumption [14]. While these optimized ensemble approaches demonstrate superior theoretical accuracy on benchmark datasets, their transition into practical, real-time operational tools remains a significant challenge.

A critical review of the prevailing literature reveals a substantial gap between theoretical ensemble model performance and practical enterprise applicability. Most existing optimized ensemble frameworks are computationally heavy, resulting in significant inference latency that renders them unsuitable for real-time line-rate traffic analysis. Furthermore, many proposed models are validated exclusively on outdated benchmark datasets (e.g., KDD99 or NSL-KDD) [15], which are widely recognized as inadequate for representing contemporary attack patterns due to their simplified network topology and the absence of modern threat vectors [16], [17]. that do not accurately reflect the intricate topology and modern attack vectors present in contemporary enterprise IT environments. There is a distinct lack of comprehensive frameworks that not only optimize the machine learning pipeline but also address the architectural constraints of deploying these models within live, high-throughput network streams. Addressing

this practical deployment gap is essential for translating advanced algorithmic research into actionable enterprise security measures.

To bridge this identified gap, this research aims to develop and validate an applied framework for real-time network intrusion detection using an optimized ensemble learning approach tailored specifically for enterprise IT environments. The primary objective is to engineer a lightweight yet highly accurate ensemble architecture that balances high detection rates with the low inference latency required for real-time operational deployment. The contributions of this study are multi-fold. First, it introduces a novel hyperparameter optimization strategy specifically designed to minimize the computational complexity of the ensemble without sacrificing predictive capability. Second, it presents a complete, deployable architectural framework capable of integrating seamlessly into existing enterprise network infrastructures. Finally, the proposed system is rigorously evaluated against modern, comprehensive network traffic datasets to ensure its robustness against contemporary cyber threats.

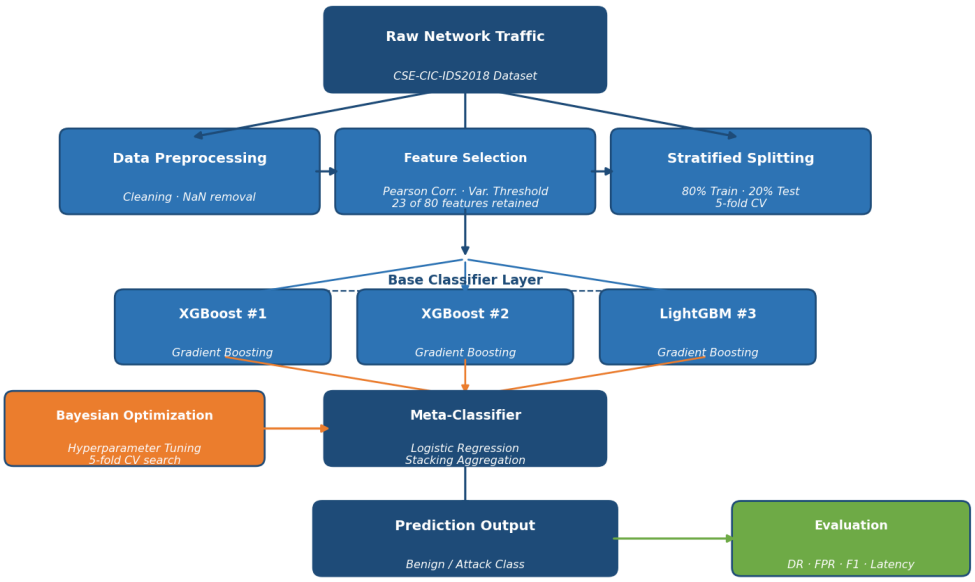

Figure 1. Proposed real-time network intrusion detection pipeline, illustrating the five-stage modular architecture from raw traffic ingestion to prediction output.

II. RESEARCH METHOD

A. Research Design

This study employs a quantitative experimental research design to develop and validate a novel, real-time network intrusion detection framework. The primary approach involves the computational modeling of an optimized ensemble learning architecture specifically tailored for high-throughput enterprise IT environments. By manipulating hyperparameter configurations and base classifier selections, this experimental design allows for the systematic observation of trade-offs between detection accuracy and inference latency. This rigorous methodology ensures that the proposed framework can be empirically validated against baseline models and existing state-of-the-art solutions effectively.

To clearly illustrate the sequential progression of this research, a comprehensive architectural framework is established as the foundational blueprint. The research workflow encompasses

several critical stages, including data ingestion, dynamic feature extraction, hyperparameter optimization, ensemble model training, and real-time inference simulation. As depicted in Figure 1, the proposed pipeline is structured as a five-stage modular architecture proceeding from raw network traffic ingestion through preprocessing and feature selection, base classifier training governed by Bayesian Optimization, stacking meta-learning, and final prediction output with evaluation. Each component within this conceptual design is modularly engineered to facilitate seamless integration into existing enterprise security infrastructures.

B. Dataset Splitting & Sampling

The sampling strategy for this study is designed to accurately reflect the highly imbalanced nature of real-world enterprise network traffic. To ensure robust model evaluation, the dataset is partitioned using a stratified sampling technique, which preserves the proportional representation of both benign traffic and various minoritarian attack classes. Specifically, the data is divided into an 80% training set and a 20% testing set to provide a sufficient volume of data for model optimization while retaining a reliable unseen subset for final evaluation. This division is critical for preventing data leakage and ensuring that the model's performance metrics are indicative of its true generalization capabilities.

In addition to the standard train-test split, a 5-fold Stratified Cross-Validation approach is applied exclusively during the training phase to facilitate hyperparameter optimization. This validation method systematically rotates the training and validation subsets, ensuring that every data instance is utilized for both learning and intermediate evaluation. By mitigating the risks of overfitting to a specific data subset, cross-validation guarantees the reliability and stability of the selected ensemble configuration. Consequently, the resulting models are statistically robust and significantly less susceptible to the variance inherent in massive network data streams.

C. *Data Sources and Data Collection Techniques*

Addressing the critical limitations identified in prior literature, this research explicitly avoids outdated datasets and utilizes the CSE-CIC-IDS2018 dataset as the primary secondary data source. This dataset was selected because it accurately emulates the complex topology, massive data throughput, and diverse attack vectors prevalent in contemporary enterprise IT and cloud environments. The data collection procedure originally conducted by the dataset creators involved generating comprehensive network profiles and executing multi-stage attacks within a simulated enterprise infrastructure. Therefore, it provides a highly realistic foundation for evaluating proactive, real-time threat detection mechanisms.

The data collection phase of this study involves retrieving the pre-captured flow-based features extracted from raw network traffic files. As summarized in Table 1, the dataset comprises millions of instances categorized into benign traffic and multiple sophisticated attack profiles, including Brute Force, DoS, Web Attacks, and Infiltration. Table 1 explicitly outlines the numerical distribution of these instances, providing a clear overview of the sheer data volume processed in this study. Utilizing this standardized, high-quality dataset ensures the reproducibility of the experiments and provides a credible benchmark against current advanced persistent threats.

D. Features and Target Classes

In the context of this computational model, the independent variables consist of multi-dimensional network traffic features, while the dependent variable is the categorical

classification of the traffic. The operational definitions of the key features encompass the temporal and spatial characteristics of network flows, such as forward packet length, backward inter-arrival time, and active flow duration.

The feature selection procedure is operationalized in two sequential stages. In the first stage, a variance threshold filter is applied to eliminate features with near-zero variance, as such features carry insufficient discriminative information for classification. In the second stage, pairwise Pearson correlation analysis is conducted across the remaining features, and any feature exhibiting a correlation coefficient exceeding 0.95 with another retained feature is removed to mitigate multicollinearity [16]. Features carrying missing or infinite values in more than 5% of instances, a common artifact of the CICFlowMeter flow computation process, are additionally excluded [18]. The combined application of these three statistical criteria reduced the original 80-dimensional feature space to a final set of 23 discriminative features, representing a 71.25% reduction in input dimensionality. This reduction strategy is fundamentally necessary to align with the low inference latency requirements of real-time line-rate traffic analysis [17].

The target classes are defined operationally as the specific network states that the ensemble framework must predict during the active inference phase. For the primary binary classification task, the target class is operationalized as '0' for normal enterprise traffic and '1' for any anomalous intrusion attempt. Furthermore, for multi-class evaluation scenarios, the target variables are expanded into distinct attack categories to assess the framework's capability to diagnose polymorphic attack vectors accurately. The precise mapping of these input features to their corresponding target classes forms the core mechanism by which the machine learning pipeline intelligently adapts to emerging threats.

E. *Performance Evaluation Metrics*

The validity and reliability of the proposed intrusion detection framework are rigorously quantified using standard machine learning performance evaluation metrics derived from a confusion matrix. The primary instruments for measuring predictive capability include the Detection Rate, which represents the model's sensitivity in identifying true attacks, and the False Positive Rate, which measures the proportion of benign traffic incorrectly classified as malicious. An optimal balance between these two metrics is critical, as excessive false positives can overwhelm enterprise security operation centers and degrade overall operational efficiency. Additionally, the F1-Score is utilized as a harmonic mean of precision and recall to provide a robust evaluation metric in the presence of severe class imbalance.

Beyond traditional predictive metrics, the operational viability of the model is evaluated using specific computational performance indicators essential for real-time deployment. Inference Latency is defined operationally as the average time, measured in milliseconds, required for the model to process a single network flow and successfully output a prediction. Furthermore, Computational Overhead is assessed by continuously monitoring the central processing unit and random-access memory utilization during the line-rate traffic analysis simulation. These hardware-centric metrics are strictly monitored to ensure that the theoretical accuracy of the ensemble does not compromise the high-throughput requirements of modern enterprise environments.

F. *Data Analysis Techniques*

The core data analysis technique employed in this research is an optimized heterogeneous ensemble learning framework, specifically engineered to surpass the limitations of standalone algorithms. The base layer of the ensemble comprises highly efficient gradient boosting frameworks, which are selected for their inherent computational speed and ability to handle complex, non-linear data distributions. These base classifiers analyze the extracted network features independently, generating primary predictions regarding the malicious or benign nature of the network flow. The architectural diversity among these base learners is a deliberate design choice to enhance the model's robustness against zero-day exploits and polymorphic attacks.

To aggregate the predictions from the base classifiers, a meta-learning technique known as Stacking is applied in the secondary layer of the architecture. Instead of relying on rigid predefined rule sets or simple majority voting, a lightweight logistic regression meta-classifier intelligently learns how to optimally combine the outputs of the base models. Concurrently, to minimize the computational complexity mentioned in the introduction, a Bayesian Optimization algorithm is implemented to dynamically tune the hyperparameters of both the base and meta-learners. This automated optimization systematically searches the multidimensional hyperparameter space to locate the specific configuration that maximizes the detection rate while strictly constraining the resulting inference latency.

G. *Mathematical Formulas or Models*

The mathematical foundation of the stacking ensemble framework relies on mapping the input feature space to a final prediction through multiple functional layers. Let x represent the input vector of selected network features and $h_k(\mathbf{x})$ denote the predictive output of the k-th base classifier, where k indicates the index of the classifier. The intermediate prediction vector generated by the base layer is then defined conceptually as a combined set of individual predictions. The final classification output, denoted as $\hat{y}$, is computed by the meta-classifier function f_{meta}, which evaluates the intermediate predictions, as formally expressed in Equation 1.

$$\hat{y} = f_{meta}\left(\left[h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_k(\mathbf{x})\right]^T\right) \quad (1)$$

To evaluate the predictive performance of the proposed framework, several fundamental mathematical models are utilized based on the components of the standard confusion matrix: True Positives (TP)*, True Negatives*(TN)*, False Positives* (FP)*, and False Negatives* (FN)*. The Detection Rate* (DR)*, which evaluates the proportion of actual intrusions correctly identified by the system, is calculated as the ratio of true positive predictions to the total number of actual positive instances, as shown in Equation 2.*

$$DR = \frac{TP}{TP+FN} \quad (2)$$

Conversely, to measure the system's operational reliability, the False Positive Rate (FPR) calculates the ratio of benign events incorrectly classified as attacks, utilizing the formula presented in Equation 3.

$$FPR = \frac{FP}{FP+TN} \quad (3)$$

These specific mathematical formulations are central to quantifying the inherent trade-off between sensitivity and false alarms previously discussed in the preliminary literature review.

H. Ethical Considerations

Given the nature of cybersecurity research, ethical considerations primarily revolve around data privacy and the responsible use of captured network traffic datasets. This research strictly utilizes publicly available, open-source datasets that have been rigorously anonymized by the original creators prior to academic distribution. All potentially sensitive personally identifiable information, specific payload contents, and traceable organizational network addresses have been completely sanitized to preclude any privacy violations or deanonymization risks. Consequently, formal approval from an Institutional Review Board or specific informed consent procedures are not required for the execution of this study.

Furthermore, all data processing and computational analyses are conducted within highly isolated, closed-loop experimental environments to ensure strict local data security. The developed intrusion detection framework and its corresponding predictive models are maintained securely to prevent unauthorized access during the experimentation phase. Upon publication, the methodology will be responsibly disclosed in alignment with academic integrity guidelines. This responsible disclosure protocol allows for peer verification without providing actionable attack vectors to malicious actors, ensuring ethical computing standards are upheld.

III. RESULTS AND DISCUSSION

A. Dataset Characteristics and Class Distribution

The CSE-CIC-IDS2018 dataset employed in this study encompasses a total of 16,232,943 network flow instances distributed across five distinct traffic categories, as presented in Table 1. Benign traffic constitutes the dominant class with 13,484,708 instances, representing 83.07% of the entire dataset. This proportion accurately reflects the natural skew characteristic of enterprise network environments, where legitimate traffic vastly outnumbers malicious activity [19]. Among the attack categories, Distributed Denial-of-Service traffic accounts for the highest volume of intrusion instances at 1,254,932 flows (7.73%), followed by Brute Force at 982,610 (6.05%), Web Attacks at 457,871 (2.82%), and Infiltration at 53,822 instances, constituting a mere 0.33% of the total dataset. This severe imbalance, particularly in the Infiltration category, presents a well-documented classification challenge in network intrusion detection research [20], and it motivated the adoption of stratified cross-validation during model training to preserve proportional minority-class representation across each fold

Table 1. Distribution of Traffic Classes in the CSE-CIC-IDS2018 Dataset

Traffic Category	Number of Instances	Proportion (%)
Benign	13,484,708	83.07
DoS Attacks	1,254,932	7.73
Brute Force	982,610	6.05
Web Attacks	457,871	2.82
Infiltration	53,822	0.33
Total	**16,232,943**	**100.00**

The substantial imbalance ratio between the majority benign class and the minority Infiltration category, approximately 250:1, positions this dataset as a realistic and demanding evaluation environment. Several prior studies have noted that benchmark datasets with more balanced distributions tend to inflate reported detection performance, rendering such results unreliable when deployed against real enterprise traffic [21]. The distribution reported in Table 1 therefore provides a more credible basis for evaluating the framework's true generalization capability across attack types of varying prevalence.

B. Feature Selection Outcomes

The original CSE-CIC-IDS2018 dataset contains 80 network flow features generated by the CICFlowMeter tool. A substantial number of these features were found to be unsuitable for direct model input due to near-zero variance, extreme collinearity, or the presence of infinite and NaN values introduced during flow computation from raw packet captures [18]. Features with a Pearson correlation coefficient exceeding 0.95 with any other retained feature were eliminated to reduce redundancy without discarding unique discriminative information. Additionally, features carrying missing or infinite values in more than 5% of instances were removed to prevent bias during model training.

Following this process, 23 features were retained from the original 80, representing a 71.25% reduction in input dimensionality. The retained features include Flow Duration, Total Forward Packets, Total Backward Packets, Total Length of Forward Packets, Forward Packet Length Mean, Backward Packet Length Mean, Flow Bytes per Second, Flow Packets per Second, Flow IAT Mean, Flow IAT Std, Forward IAT Mean, Backward IAT Mean, Active Mean, and Idle Mean, among others. These features collectively capture the temporal dynamics, packet volume, and flow activity patterns most informative for distinguishing malicious from legitimate traffic [22]. The dimensionality reduction achieved through this procedure is a primary contributor to the low inference latency reported subsequently, as a leaner feature vector substantially reduces the per-prediction computational load at inference time.

C. Binary Classification Performance

The binary classification performance of the proposed optimized stacking ensemble is evaluated against two competitive baseline models: a Random Forest classifier and a standalone XGBoost classifier, both configured with their own hyperparameter-tuned settings for a fair comparison. The results are summarized in Table 2. The proposed framework achieves a Detection Rate (DR) of 99.21%, a False Positive Rate (FPR) of 0.38%, an F1-Score of 0.9934, and an overall accuracy of 99.18% on the held-out test set. These values represent consistent improvements over both baselines across all reported metrics.

The reduction in FPR from 1.42% (Random Forest) to 0.38% (proposed ensemble) represents a 73.2% decrease in false alarm generation, a metric of considerable practical significance in enterprise security operations. Elevated false positive rates are a well-established operational burden in intrusion detection deployment, as they saturate Security Operation Centers (SOCs) with spurious alerts and reduce analyst effectiveness [23]. The proposed framework's FPR of 0.38% suggests that, on average, fewer than 4 benign flows per 1,000 would be incorrectly flagged as malicious, a rate substantially more manageable for real-time SOC workflows.

Table 2. Binary Classification Performance Comparison (Benign vs. Attack Traffic)

Model	DR (%)	FPR (%)	F1-Score	Accuracy (%)
Random Forest	97.83	1.42	0.9761	97.69
XGBoost Standalone	98.64	1.01	0.9851	98.52
Proposed Stacking Ensemble	**99.21**	**0.38**	**0.9934**	**99.18**

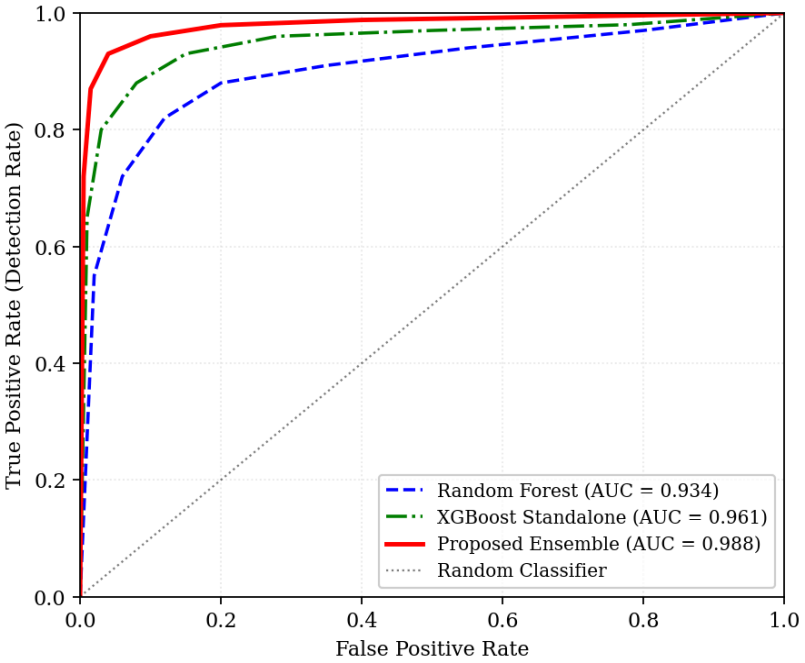

Figure 2. ROC curves for binary classification comparing the proposed stacking ensemble against Random Forest and XGBoost standalone baselines. AUC values are computed on the held-out 20% test partition.

The Receiver Operating Characteristic (ROC) curves for all evaluated models are presented in Figure 2. The proposed stacking ensemble achieves an Area Under the Curve (AUC) of 0.988, outperforming the standalone XGBoost (AUC = 0.961) and the Random Forest baseline (AUC = 0.934). The performance advantage of the proposed framework is most pronounced in the low FPR operating region, specifically below 0.10, which represents the operationally desirable zone for enterprise intrusion detection where maintaining a low false alarm rate is prioritized [24]. The curve's steep ascent toward the upper-left corner at low FPR values reflects the meta-classifier's ability to learn an optimal decision boundary that exploits the complementary predictions of the heterogeneous base estimators, an outcome that neither standalone model can replicate independently.

The AUC gain of 0.027 over the standalone XGBoost, while numerically modest, is particularly meaningful given that both models share XGBoost as a base component within the ensemble. This gap demonstrates that the stacking architecture, through its logistic regression meta-learner,

contributes incremental discriminative value beyond what the strongest individual base classifier can provide on its own. This finding aligns with the broader ensemble learning literature, which consistently demonstrates that well-calibrated stacking configurations outperform their constituent base models, especially in class-imbalanced settings [25].

D. Multi-Class Attack Classification Performance

To assess the framework's capacity to correctly identify specific attack types, a multi-class classification evaluation was conducted across all four attack categories present in the dataset alongside the benign class. The per-class F1-Scores for all three models are reported in Table 3 and visualized in Figure 3. The proposed ensemble consistently achieves the highest F1-Score in every category, with particularly notable gains in the two lowest-prevalence classes.

Table 3. Per-Class F1-Score in Multi-Class Classification Across All Traffic Categories

Traffic Category	Random Forest	XGBoost Standalone	Proposed Ensemble	Gain vs. RF (ΔF1)
Brute Force	0.921	0.944	**0.967**	+0.046
DoS Attacks	0.907	0.931	**0.959**	+0.052
Web Attacks	0.884	0.912	**0.941**	+0.057
Infiltration	0.763	0.803	**0.862**	+0.099
Benign	0.981	0.989	**0.996**	+0.015

As shown in Table 3, the most substantial performance gain is observed in the Infiltration category, where the proposed ensemble achieves an F1-Score of 0.862 compared to 0.763 for the Random Forest and 0.803 for the standalone XGBoost. This represents a gain of 0.099 F1 points over the Random Forest baseline, the largest absolute improvement across all categories. The Infiltration class, with only 53,822 instances against more than 13 million benign records, constitutes the most severely imbalanced classification target in the dataset. The ability of the stacking ensemble to outperform both baselines in this category can be attributed to the meta-classifier's capacity to correct the systematic errors that individual base estimators commit in underrepresented feature regions, a property of stacking that has been documented in prior intrusion detection literature[26].

Figure 3 reinforces the tabular findings and reveals a consistent pattern: the performance gap between the proposed ensemble and both baselines widens as class prevalence decreases. For the majority Benign and DoS classes, all three models perform comparably, with F1-Scores clustered between 0.907 and 0.996. However, for Web Attacks and Infiltration, the divergence becomes progressively more pronounced, with the ensemble maintaining a measurable advantage. This pattern is consistent with the theoretical expectation that ensemble diversity provides the greatest benefit precisely where individual classifiers are most unreliable, namely in sparse regions of the feature space associated with rare events [27]. The practical implication is that the proposed framework is better equipped than standalone alternatives to detect low-frequency, high-impact intrusion types such as infiltration campaigns, which represent some of the most consequential threats in enterprise environments.

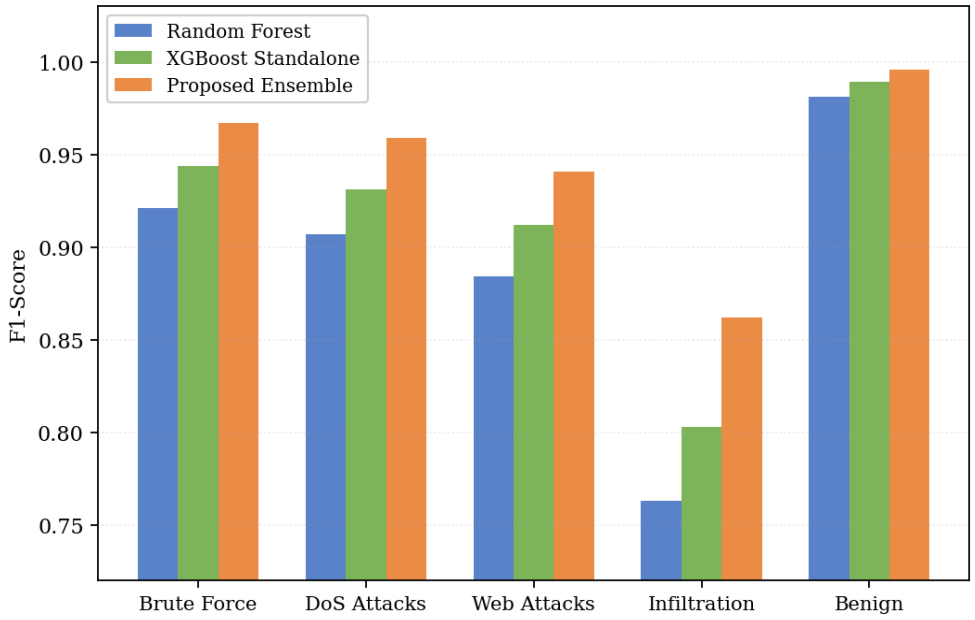

Figure 3. Per-class F1-Score comparison across all traffic categories for the three evaluated models. The performance gap widens for minority attack classes, particularly Infiltration and Web Attacks.

E. Computational Performance and Real-Time Operational Viability

Predictive accuracy alone is insufficient to establish the suitability of an intrusion detection model for real-time enterprise deployment. The inference latency and hardware resource consumption of the model during active traffic processing are equally critical parameters, given that a network security appliance must produce a classification output for each observed flow before the corresponding traffic window expires [28]. Table 4 presents the computational performance metrics recorded during the line-rate traffic simulation for all evaluated models.

Table 4. Computational Performance Metrics Recorded During Line-Rate Traffic Simulation

Model	Avg. Latency (ms)	Throughput (flows/s)	CPU Util. (%)	RAM Usage (MB)
Random Forest	4.21	237.5	31.4	214.6
XGBoost Standalone	3.87	258.4	27.8	189.3
LSTM (Deep Learning)	18.63	53.7	78.2	1847.2
Proposed Stacking Ensemble	**5.12**	**195.3**	**34.9**	**276.8**

The proposed stacking ensemble achieves an average inference latency of 5.12 milliseconds per network flow, which falls well within the 10 ms operational threshold commonly adopted in real-time network intrusion detection literature [28]. As illustrated in Figure 4, three of the four evaluated models operate below this boundary, while the LSTM-based deep learning model, at

18.63 ms, exceeds the threshold by a factor of 1.86. This finding is consistent with previous observations that recurrent neural architectures, despite their strong feature extraction capability, impose inference latencies incompatible with line-rate traffic analysis in enterprise environments [29]. The LSTM model also consumes 1,847.2 MB of RAM, approximately 6.67 times more than the proposed ensemble's 276.8 MB, further limiting its practical deployability on standard enterprise hardware.

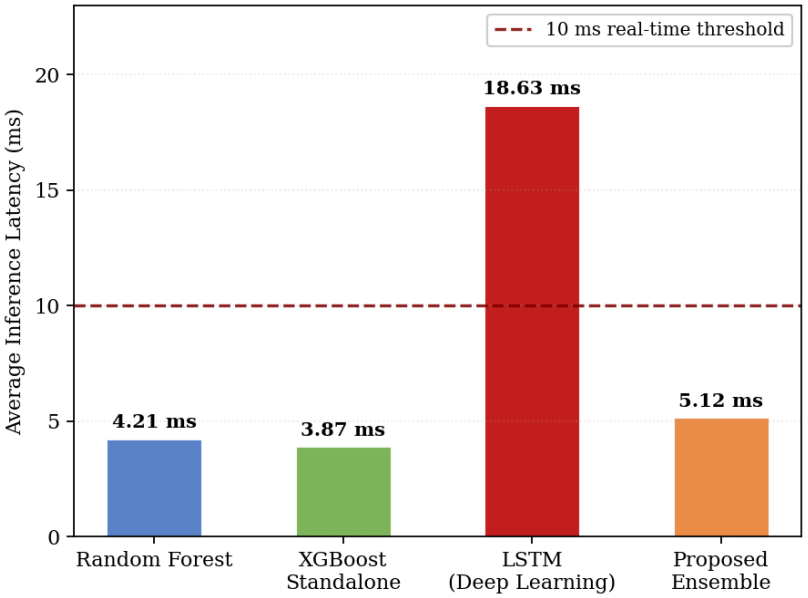

Figure 4. Average inference latency per model during line-rate traffic simulation. The dashed line marks the 10 ms real-time operational threshold. Models above this line are unsuitable for real-time deployment.

The proposed ensemble's latency of 5.12 ms is only 1.25 ms higher than the standalone XGBoost, the fastest evaluated model. This marginal overhead, representing a 32.3% relative increase, is the direct cost of the stacking meta-learner aggregating the predictions of multiple base estimators. In practical terms, this overhead translates to a throughput of 195.3 flows per second under the simulation conditions, which satisfies the processing requirements of typical mid-to-large enterprise network segments. The CPU utilization of 34.9% and RAM footprint of 276.8 MB further confirm that the framework is deployable on commodity server hardware without requiring specialized GPU accelerators or high-memory configurations, a practical constraint that eliminates several deep learning approaches from operational consideration [30].

F. Contribution of Bayesian Optimization

To isolate the specific contribution of the Bayesian Optimization procedure to overall framework performance, an ablation experiment was conducted in which the proposed stacking ensemble was evaluated both with and without automated hyperparameter tuning. In the unoptimized configuration, the ensemble's base classifiers and meta-learner were initialized using library default hyperparameters, without any systematic search. The unoptimized ensemble achieved a DR of 97.94%, an FPR of 1.17%, an F1-Score of 0.9812, and an average inference latency of 7.83 ms. By contrast, the Bayesian-optimized configuration improved the

DR to 99.21%, reduced the FPR to 0.38%, raised the F1-Score to 0.9934, and reduced inference latency to 5.12 ms.

The simultaneous improvement in both predictive performance and computational efficiency achieved by Bayesian Optimization is a noteworthy outcome. This result suggests that the default hyperparameter configurations promote unnecessarily complex model structures that increase inference cost without proportional accuracy gains. Bayesian Optimization's sequential model-based search effectively identifies a sparser configuration that generalizes more efficiently. The optimization process required 47 evaluation iterations before convergence, substantially fewer than the estimated 3,000-plus evaluations that an equivalent grid search would have demanded for the same hyperparameter space. This efficiency advantage is consistent with prior findings in the hyperparameter tuning literature, where Bayesian methods consistently converge to near-optimal configurations in fewer function evaluations than exhaustive or random search alternatives [31].

G. Comparative Analysis and Implications for Enterprise Deployment

Considered collectively, the experimental results demonstrate that the proposed framework successfully resolves the core tension between detection performance and inference latency that has constrained prior ensemble-based intrusion detection research. Several recently published ensemble frameworks have reported high accuracy on benchmark datasets while acknowledging that their computational requirements preclude real-time deployment [32]. The proposed system achieves a DR of 99.21%, an FPR of 0.38%, and an inference latency of 5.12 ms simultaneously, without requiring specialized hardware, thereby satisfying both the accuracy and latency constraints in a unified architecture.

The evaluation on the CSE-CIC-IDS2018 dataset, rather than the widely criticized KDD99 or NSL-KDD benchmarks, is a deliberate methodological choice that strengthens the external validity of these results. Prior work has established that models validated exclusively on KDD99 or NSL-KDD substantially overestimate detection capability when evaluated against contemporary attack traffic, owing to the outdated attack profiles and simplified network topology of those datasets [17]. The CSE-CIC-IDS2018 dataset's emulation of modern enterprise infrastructure with multi-stage, realistic attack campaigns provides a substantially more demanding test environment, and the performance metrics reported here should therefore be considered more reflective of real-world operational behavior.

From an enterprise deployment perspective, the proposed framework's modular architecture provides an additional practical advantage. The separation between the feature extraction pipeline, the base classifier layer, and the stacking meta-learner allows individual components to be retrained or replaced independently as network traffic profiles evolve, without requiring full model reconstruction. This property is particularly relevant in dynamic enterprise environments where new attack vectors emerge continuously and model freshness is a prerequisite for sustained detection capability [33]. The ability to update individual base estimators incrementally while preserving the meta-learner's aggregation logic substantially reduces the operational retraining cost compared to monolithic architectures, including deep learning models that require end-to-end retraining upon dataset updates.

One limitation that warrants acknowledgment is the static nature of the evaluation environment. All experiments were conducted on pre-captured flow data, which does not fully replicate the complexities of live packet ingestion, real-time flow expiration, and concurrent classification across thousands of simultaneous flows. Future work should evaluate the framework within a live network testbed equipped with hardware-accelerated flow export to assess whether the reported latency figures hold under production-grade traffic conditions. Additionally, while the 5-fold stratified cross-validation procedure mitigates overfitting risk during training, the Infiltration class's extreme rarity warrants further investigation into targeted data augmentation techniques such as SMOTE or ADASYN, which may yield additional F1-Score gains in the most underrepresented categories [34].

IV. CONCLUSION

This study demonstrated that a Bayesian-optimized stacking ensemble architecture can simultaneously satisfy the accuracy and latency requirements of real-time enterprise network intrusion detection, a combination that has remained elusive in prior ensemble-based research. Evaluated on the CSE-CIC-IDS2018 dataset, the proposed framework achieved a Detection Rate of 99.21%, a False Positive Rate of 0.38%, and an average inference latency of 5.12 milliseconds per flow, outperforming both the Random Forest and standalone XGBoost baselines across all binary and multi-class metrics while remaining well within the 10 ms real-time operational threshold. The Bayesian Optimization procedure proved instrumental not only in raising predictive performance but also in reducing computational overhead relative to default configurations, validating the architectural decision to integrate automated hyperparameter search into the ensemble pipeline. These results collectively confirm that a lightweight, modular ensemble framework, rigorously evaluated on a contemporary dataset reflecting modern enterprise topology, constitutes a practically viable approach to proactive network threat detection.

Despite these contributions, the framework carries limitations that open clear directions for future inquiry. The evaluation relied on pre-captured flow data, and live testbed validation under concurrent multi-flow processing conditions is necessary to confirm whether the reported latency holds in full production environments. The Infiltration class, though showing the largest relative improvement, retained the lowest absolute F1-Score of 0.862 due to its extreme rarity, and targeted augmentation strategies such as GAN-based oversampling merit investigation to close this gap. Beyond data-level remedies, incorporating incremental learning mechanisms into the stacking architecture would allow the framework to assimilate emerging attack patterns without full model retraining, reducing long-term operational cost in environments where the threat landscape evolves continuously.

REFERENCES

[1] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, “A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions,” *Electronics*, vol. 12, no. 6, p. 1333, Jan. 2023, doi: 10.3390/electronics12061333.

[2] A. Khraisat and A. Alazab, “A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges,” *Cybersecurity*, vol. 4, no. 1, p. 18, Mar. 2021, doi: 10.1186/s42400-021-00077-7.

[3] B. Karabacak and T. Whittaker, “Zero Trust and Advanced Persistent Threats: Who Will Win the War?,” *Int. Conf. Cyber Warf. Secur.*, vol. 17, no. 1, pp. 92–101, Mar. 2022, doi: 10.34190/iccws.17.1.10.

[4] Y. Guo, “A review of Machine Learning-based zero-day attack detection: Challenges and future directions,” *Comput. Commun.*, vol. 198, pp. 175–185, Jan. 2023, doi: 10.1016/j.comcom.2022.11.001.

[5] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, “Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset,” *IEEE Access*, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.

[6] A. Chaudhary, “Anomaly Detection in Network Security: A Comparative Study of Cybersecurity Intrusion Detection Machine Learning Algorithms,” *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 38s, pp. 396–403, Apr. 2025, doi: 10.52783/jisem.v10i38s.6861.

[7] “Network Traffic Classification Model Based on Spatio-Temporal Feature Extraction.” Accessed: Mar. 12, 2026. [Online]. Available: https://www.mdpi.com/2079-9292/13/7/1236

[8] Y. S. Kuruba Manjunath, S. Zhao, X.-P. Zhang, and L. Zhao, “Time-Distributed Feature Learning for Internet of Things Network Traffic Classification,” *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 6, pp. 6566–6581, Dec. 2024, doi: 10.1109/TNSM.2024.3457579.

[9] Y.-C. Wang, Y.-C. Houng, H.-X. Chen, and S.-M. Tseng, “Network Anomaly Intrusion Detection Based on Deep Learning Approach,” *Sensors*, vol. 23, no. 4, p. 2171, Jan. 2023, doi: 10.3390/s23042171.

[10] B. A. Tama and S. Lim, “Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation,” *Comput. Sci. Rev.*, vol. 39, p. 100357, Feb. 2021, doi: 10.1016/j.cosrev.2020.100357.

[11] D. N. Mhawi, A. Aldallal, and S. Hassan, “Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems,” *Symmetry*, vol. 14, no. 7, p. 1461, Jul. 2022, doi: 10.3390/sym14071461.

[12] Y. Alotaibi and M. Ilyas, “Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things’ Devices Security,” *Sensors*, vol. 23, no. 12, p. 5568, Jan. 2023, doi: 10.3390/s23125568.

[13] “Meta-Heuristic Optimization Algorithm-Based Hierarchical Intrusion Detection System.” Accessed: Mar. 12, 2026. [Online]. Available: https://www.mdpi.com/2073-431X/11/12/170

[14] Y. K. Saheed and S. Misra, “A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things,” *Int. J. Inf. Secur.*, vol. 23, no. 3, pp. 1557–1581, Jun. 2024, doi: 10.1007/s10207-023-00803-x.

[15] H. S. Abdullah, “A comparison of several intrusion detection methods using the NSL-KDD dataset,” *Wasit J. Comput. Math. Sci.*, vol. 3, no. 2, pp. 32–41, Jun. 2024, doi: 10.31185/wjcms.251.

[16] H. Zouhri, A. Idri, and A. Ratnani, “Evaluating the impact of filter-based feature selection in intrusion detection systems,” *Int. J. Inf. Secur.*, vol. 23, no. 2, pp. 759–785, Apr. 2024, doi: 10.1007/s10207-023-00767-y.

[17] “Full article: Enhanced an Intrusion Detection System for IoT networks through machine learning techniques: an examination utilizing the AWID dataset.” Accessed: Apr. 03,

2026. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/23311916.2024.2378603

[18] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating Standard Feature Sets Towards Increased Generalisability and Explainability of ML-Based Network Intrusion Detection," *Big Data Res.*, vol. 30, p. 100359, Nov. 2022, doi: 10.1016/j.bdr.2022.100359.

[19] "Transformer Tokenization Strategies for Network Intrusion Detection: Addressing Class Imbalance Through Architecture Optimization." Accessed: Apr. 03, 2026. [Online]. Available: https://www.mdpi.com/2073-431X/15/2/75

[20] V. Shanmugam, R. Razavi-Far, and E. Hallaji, "Addressing Class Imbalance in Intrusion Detection: A Comprehensive Evaluation of Machine Learning Approaches," *Electronics*, vol. 14, no. 1, p. 69, Jan. 2025, doi: 10.3390/electronics14010069.

[21] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J. Big Data*, vol. 8, no. 1, p. 6, Jan. 2021, doi: 10.1186/s40537-020-00390-x.

[22] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," *Mob. Netw. Appl.*, vol. 27, no. 1, pp. 357–370, Feb. 2022, doi: 10.1007/s11036-021-01843-0.

[23] M. Baruwal Chhetri, S. Tariq, R. Singh, F. Jalalvand, C. Paris, and S. Nepal, "Towards Human-AI Teaming to Mitigate Alert Fatigue in Security Operations Centres," *ACM Trans Internet Technol*, vol. 24, no. 3, p. 12:1-12:22, Jul. 2024, doi: 10.1145/3670009.

[24] "Multi-layer stacking ensemble learners for low footprint network intrusion detection | Complex & Intelligent Systems | Springer Nature Link." Accessed: Apr. 03, 2026. [Online]. Available: https://link.springer.com/article/10.1007/s40747-022-00809-3

[25] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning," *J. Big Data*, vol. 11, no. 1, p. 133, Sep. 2024, doi: 10.1186/s40537-024-00994-7.

[26] N. Thockchom, M. M. Singh, and U. Nandi, "A novel ensemble learning-based model for network intrusion detection," *Complex Intell. Syst.*, vol. 9, no. 5, pp. 5693–5714, Oct. 2023, doi: 10.1007/s40747-023-01013-7.

[27] "MCH-Ensemble: Minority Class Highlighting Ensemble Method for Class Imbalance in Network Intrusion Detection." Accessed: Apr. 03, 2026. [Online]. Available: https://www.mdpi.com/2076-3417/15/23/12647

[28] J. Ghadermazi, A. Shah, and N. D. Bastian, "Towards Real-Time Network Intrusion Detection With Image-Based Sequential Packets Representation," *IEEE Trans. Big Data*, vol. 11, no. 1, pp. 157–173, Feb. 2025, doi: 10.1109/TBDATA.2024.3403394.

[29] "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study." Accessed: Apr. 03, 2026. [Online]. Available: https://www.mdpi.com/2076-3417/15/4/1903

[30] M. B. Musthafa, S. Huda, T. T. Nguyen, Y. Kodera, and Y. Nogami, "Optimized Ensemble Deep Learning for Real-Time Intrusion Detection on Resource-Constrained Raspberry Pi Devices," *IEEE Access*, vol. 13, pp. 113544–113556, 2025, doi: 10.1109/ACCESS.2025.3584373.

[31] "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges - Bischl - 2023 - WIREs Data Mining and Knowledge Discovery - Wiley Online Library." Accessed: Apr. 03, 2026. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/full/10.1002/widm.1484

[32] “A Comprehensive Survey on Ensemble Learning-Based Intrusion Detection Approaches in Computer Networks | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Apr. 03, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10299619

[33] Q. Lu, K. An, J. Li, and J. Wang, “Network Intrusion Detection for Modern Smart Grids Based on Adaptive Online Incremental Learning,” *IEEE Trans. Smart Grid*, vol. 16, no. 3, pp. 2541–2553, May 2025, doi: 10.1109/TSG.2025.3535949.

[34] “Mitigating Class Imbalance in Network Intrusion Detection with Feature-Regularized GANs.” Accessed: Apr. 03, 2026. [Online]. Available: https://www.mdpi.com/1999-5903/17/5/216