

Research Article

OMIC: A Bagging-Based Ensemble Learning Framework for Large-Scale IoT Intrusion Detection

Jean Pierre Ntayahabiri ^{1,*}, Youssef Bentaleb ², Jeremie Ndikumagenge ³, and Hind El Makhtoum ⁴

¹ Doctoral School of the University of Burundi, Center for Research in Infrastructure, Environment and Technology (CRIET), University of Burundi, Bujumbura, Burundi; e-mail : jpnayahabiri@gmail.com

² Engineering Sciences Laboratory, ENSA Kenitra, Ibn Tofail University, Kenitra, Morocco; e-mail : ybentaleb@gmail.com

³ Center for Research in Infrastructure, Environment and Technology (CRIET), University of Burundi, Bujumbura, Burundi; e-mail : jeremie.ndikumagenge@ub.edu.bi

⁴ Engineering Sciences Laboratory, ENSA Kenitra, Ibn Tofail University, Kenitra, Morocco; e-mail : elmakhtoum_hind@live.fr

* Corresponding Author : Jean Pierre Ntayahabiri

Abstract: The research focuses on developing an Optimized Multiclass Intrusion Classifier (OMIC), an advanced framework for large-scale network intrusion detection in IoT environments. Traditional intrusion detection systems face significant challenges with increasing network complexity, attack sophistication, and the exponential growth of IoT devices, particularly in handling class imbalance, computational efficiency, and real-time processing of massive data volumes. OMIC introduces a novel ensemble approach combining LightGBM and XGBoost classifiers with a memory-optimized processing pipeline to address these limitations. The framework implements sophisticated data handling techniques, including dynamic chunk-based processing, adaptive sampling methods, and cost-sensitive learning to manage class imbalance. Experimental evaluation using the comprehensive CICIoT2023 dataset, comprising over 1 million records and 33 distinct attack types, demonstrates OMIC's exceptional performance with an overall accuracy of 99.26%. The framework achieves perfect precision, recall, and F1-scores for most DDoS and DoS attack categories, significantly outperforming traditional machine learning and deep learning approaches. While excelling in most attack categories, OMIC shows limitations in detecting certain web-based attacks and reconnaissance activities, suggesting areas for future enhancement. The framework's superior performance in handling large-scale data while maintaining high detection accuracy positions it as a significant advancement in IoT network security, offering practical solutions for real-world deployments.

Keywords: Ensemble Learning; Intrusion Detection Systems; Machine Learning; Memory-Optimized Processing; Multiclass Classification; Network Security; Network Traffic Analysis.

Received: January, 19th 2025

Revised: February, 16th 2025

Accepted: February, 20th 2025

Published: February, 23rd 2025

Curr. Ver.: February, 23rd 2025



Copyright: © 2025 by the authors.
Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>)

1. Introduction

Network security faces unprecedented challenges in today's digital landscape, where the explosive growth of Internet of Things (IoT) devices and network traffic volumes has fundamentally transformed the threat landscape [1]. While traditional intrusion detection systems (IDS) have historically served as the cornerstone of network defense, employing approaches from signature-based detection to basic machine learning (ML) techniques[2], they are increasingly proving inadequate for modern security demands. The sheer scale, complexity, and sophistication of contemporary cyber threats have exposed critical limitations in these conventional approaches[3], [4]. The urgency of addressing these limitations cannot be overstated. As organizations rapidly transition to IoT-centric infrastructures, they face a perfect storm of security challenges: exponentially increasing data volumes, unprecedented device heterogeneity, and increasingly sophisticated attack vectors. Traditional IDS solutions, designed for simpler, more predictable network environments, struggle to adapt to this new

reality. This critical gap between current capabilities and emerging threats creates an urgent need for more robust, scalable, and intelligent detection systems.

This research addresses this pressing challenge by developing an Optimized Multiclass Intrusion Classifier (OMIC) designed to handle modern network environments' scale and complexity while maintaining high detection accuracy across multiple attack classes. This work emerges from several critical observations in current security landscapes. First, the volume of network traffic in modern environments has grown beyond the processing capabilities of traditional IDS solutions, creating dangerous blind spots in security coverage. Second, the diversity of IoT devices has introduced unprecedented heterogeneity in network patterns, making traditional detection models increasingly unreliable. Third, modern cyber-attacks have evolved to exploit the gaps between different detection mechanisms, highlighting the need for more sophisticated, integrated approaches.

Previous research in network intrusion detection has explored various methodological approaches, each with distinct advantages and limitations. Signature-based detection methods have demonstrated excellent precision in identifying known attacks but struggle with zero-day threats and variants of existing attacks[5]. Machine learning-based approaches have shown promise in detecting novel attacks through pattern recognition, with supervised and unsupervised learning techniques applied to the problem[6], [7]. Deep learning (DL) methods have recently emerged as powerful tools for intrusion detection, offering improved feature extraction capabilities and the potential for better generalization[8], [9]. However, these advanced methods often require significant computational resources and struggle with real-time processing requirements[10]. In contrast, gradient-boosting frameworks like LightGBM and XGBoost have demonstrated remarkable success in handling large-scale data with superior computational efficiency and comparable accuracy to DL models in various domains[11], [12].

The strengths and weaknesses of existing methods reveal several critical gaps in current approaches. While signature-based systems offer quick response times and low false-positive rates for known attacks, they require constant updates and fail to detect novel threats[13]. Traditional ML approaches provide better generalization but often struggle with class imbalance and high-dimensional data [14]–[16]. DL methods offer superior feature learning capabilities but face challenges in terms of computational efficiency and interpretability[17]. The analysis of recent literature suggests that ensemble methods, particularly those leveraging gradient boosting frameworks, could address these limitations by combining high accuracy with computational efficiency and better interpretability [18]. Additionally, many existing solutions struggle to maintain performance when scaled to handle large volumes of network traffic, particularly in environments with resource constraints[7], [8], [19], [20]. Based on these observations, we hypothesize that an ensemble approach combining LightGBM and XGBoost can achieve superior performance in network intrusion detection by leveraging their complementary strengths: LightGBM's leaf-wise growth strategy for handling sparse data and XGBoost's regularization capabilities for preventing overfitting. This hypothesis is supported by recent studies demonstrating the effectiveness of gradient-boosting frameworks in handling imbalanced, high-dimensional data with resource efficiency[21], [22].

The limitations of traditional methods give rise to several fundamental research problems that need to be addressed. The primary challenge lies in developing a system that can efficiently process massive volumes of network traffic data while maintaining high detection accuracy across multiple attack classes. This is complicated by the need to handle highly imbalanced datasets, where attack patterns represent a small fraction of overall traffic. The high-dimensional nature of network data and the requirement for real-time processing create additional challenges. Furthermore, the system must adapt to evolving attack patterns while maintaining reasonable computational and memory requirements.

To address these challenges, OMIC is a novel framework combining advanced ensemble learning techniques with optimized data processing pipelines. This approach leverages the complementary strengths of multiple ML algorithms, specifically integrating LightGBM and XGBoost classifiers in a unique ensemble configuration. The implementation is coupled with a sophisticated memory-optimized processing pipeline that enables efficient large-scale data handling through dynamic chunk-based processing. The framework incorporates adaptive sampling methods and cost-sensitive learning techniques to address class imbalance issues while also employing advanced feature engineering and selection methodologies to reduce computational overhead.

The key contributions of this work span several dimensions of network security and intrusion detection. A novel memory-optimized data processing pipeline that dynamically adapts to available system resources while maintaining processing efficiency is introduced. The sophisticated ensemble learning architecture combines multiple classifiers with a dynamic weighting scheme that adjusts based on performance characteristics for different attack classes. An advanced approach to handling class imbalance is presented through a multi-level balancing strategy that combines data-level and algorithm-level techniques. Additionally, the work contributes a comprehensive feature engineering and selection methodology specifically tailored for network security applications, along with a scalable implementation strategy that enables practical deployment in real-world environments.

The remainder of this paper is organized as follows. Section 2 presents a comprehensive review of related work in network intrusion detection, focusing on recent advances in ML approaches, class imbalance management, ensemble learning, and large-scale data processing. Section 3 details the architecture of the OMIC framework, describing its core components and their interactions. Moreover, this section provides detailed technical specifications and optimization guidelines for implementing the framework. Section 4 discusses the specific benefits and advantages of OMIC in IoT environments. Section 5 presents the experimental evaluation, including the setup, traffic distribution analysis, and comprehensive performance results compared with existing approaches. Finally, Section 6 concludes the paper with a summary of the contributions, limitations, and directions for future research.

2. Related Work

Network intrusion detection has evolved significantly over recent decades, progressing from simple signature-based approaches to sophisticated ML solutions. This evolution has been marked by several key research directions that have shaped our current understanding and implementation of network security systems. In recent years, ML approaches in network intrusion detection have shown remarkable advancement, particularly in addressing class imbalance and feature selection challenges. Setiadi et al.[23] investigate the effectiveness of a Bidirectional Gated Recurrent Unit (BiGRU) model combined with feature selection techniques for detecting phishing websites. Their approach involves data cleaning and preprocessing, feature normalization, and feature selection to identify the most relevant attributes for classification. Their study highlights the potential of integrating BiGRU models with feature selection and cross-validation to develop highly accurate phishing detection systems, providing a reliable solution to enhance cybersecurity measures. Gul et al. [24] significantly contributed by integrating a GAN-based oversampling strategy to overcome imbalanced dataset challenges while employing the Random Forest(RF) importance algorithm for optimal feature subset selection. Their work demonstrated the effectiveness of combining multiple DL techniques, including Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks for attack classification. Building on this foundation, Kamil [25]proposed an enhanced network security framework incorporating an IDS between endpoint devices. Their framework addresses big data challenges and the complexity of updating threat databases through a supervised CNN-based approach, integrating Recursive Feature Elimination (RFE) and XGBoost for resource optimization.

The management of class imbalance in network security has emerged as a critical research focus. Ntayagabiri et al. [26] present comprehensive findings on supervised learning algorithms for IoT attack detection, with RF achieving 99.29% accuracy and leading precision at 82.30%. Their work reveals critical insights into the challenges of balancing detection accuracy with practical deployment constraints, particularly in resource-constrained environments. This research highlights the ongoing challenge of maintaining high performance across diverse attack types while managing computational resources effectively. Al-Shehari and Al-sowail [27]developed an internal data leak detection model utilizing various random sampling techniques and established ML algorithms. Babu and Rao [28] complemented this work who introduced a modified conditional generative adversarial network (MCGAN) technique to address class imbalance by generating additional samples to balance minority and majority classes. Abdelkhalek and Mashaly [29] further advanced this field by proposing a data resampling technique that combines Adaptive Synthetic (ADASYN) and Tomek Links algorithms with various DL models. Thakkar and Lohiya [30] approached the class imbalance problem through an ensemble learning perspective, implementing a Bagging classifier with a

deep neural network (DNN) as the base estimator in corporate class weights to influence the DNN training process.

Ensemble learning has emerged as a powerful approach for improving detection accuracy and robustness. Çetin et al. [31] develop a hybrid approach combining the sub-models Logistic Regression (LR), Gaussian Naive Bayes (GNB), and RF, employing both Voting and Stacking techniques for cyberattack prediction and detection. Dhahir [32] introduces a novel hybrid technique that combines Clustering-Based Local Outlier Factor (CBLOF) and XGBoost to improve the accuracy of Distributed Denial of Service (DDoS) attack detection from network traffic. However, the study notes that this model is computationally intensive and suggests real-time large-scale data analysis optimization strategies. Rane et al. [33] researched ensemble techniques comprehensively, highlighting their applications across various domains and addressing key challenges such as computational requirements and model interpretability. Their work explores potential solutions, including developing more efficient algorithms and integrating explainable AI frameworks. Alserhani and Aljared [34] extended this research by investigating the potential of ensemble models for enhancing NIDS functionalities, presenting a hybrid NIDS model that leverages ensemble ML techniques. Arukonda et al. [35] introduced an innovative Bagging approach with Teaching-Learning-Based Optimization (BA-TLBO) for ensemble optimization, proposing a novel fitness function that balances accuracy and diversity.

Large-scale network traffic data processing presents unique challenges that have been addressed through various hybrid approaches. Itasoy et al. [36] developed a hybrid detection system combining Isolation Forest for anomaly detection and XGBoost for classification, achieving high accuracy in ransomware detection with minimal false positives. Batalov et al. [37] advanced this work by integrating Isolation Forests with Long Short-Term Memory (LSTM) networks for sequence modeling, offering improved detection capabilities for ransomware activities within network traffic. Lumazine et al. [38] further contributed to this field by proposing a hybrid detection model combining CNN for feature extraction and Isolation Forest for anomaly detection, providing an adaptive approach that doesn't rely on static signatures.

Despite these advances, several crucial limitations remain unaddressed in current research. First, while existing solutions demonstrate high accuracy in controlled environments, they often struggle with real-time processing of massive data volumes, a critical requirement for modern network environments. Second, although recent works have progressed in handling class imbalance, they typically focus on either data-level or algorithm-level solutions in isolation. The lack of a comprehensive approach that integrates multiple strategies for handling imbalanced data remains a significant gap in the field.

Third, while ensemble methods have shown promise, current implementations often fail to effectively leverage different algorithms' complementary strengths, particularly in adapting to evolving attack patterns while maintaining computational efficiency. While advancing hybrid approaches, the work by Çetin et al. [31] still leaves room for improvement in handling complex, multi-dimensional attack scenarios. Fourth, despite the strong performance metrics reported in recent studies, including Ntayagabiri et al. [26] comprehensive analysis, there remains a crucial gap in developing solutions that can maintain high detection accuracy while operating within the practical constraints of real-world network environments.

These advances in the field have established solid foundations for advanced IDS while highlighting several remaining challenges. This work builds upon these advancements while addressing key limitations of existing approaches, particularly in large-scale data processing, class imbalance management, adaptive learning capabilities, and computational complexity optimization.

3. Framework Architecture and Technical Specifications

3.1. Overview and System Architecture

The OMIC framework represents a sophisticated approach to network security through its innovative architecture combining advanced data processing and intelligent classification capabilities, as illustrated in Figure 1. This framework's architecture integrates two main components that work in synergy to deliver efficient and accurate intrusion detection capabilities.

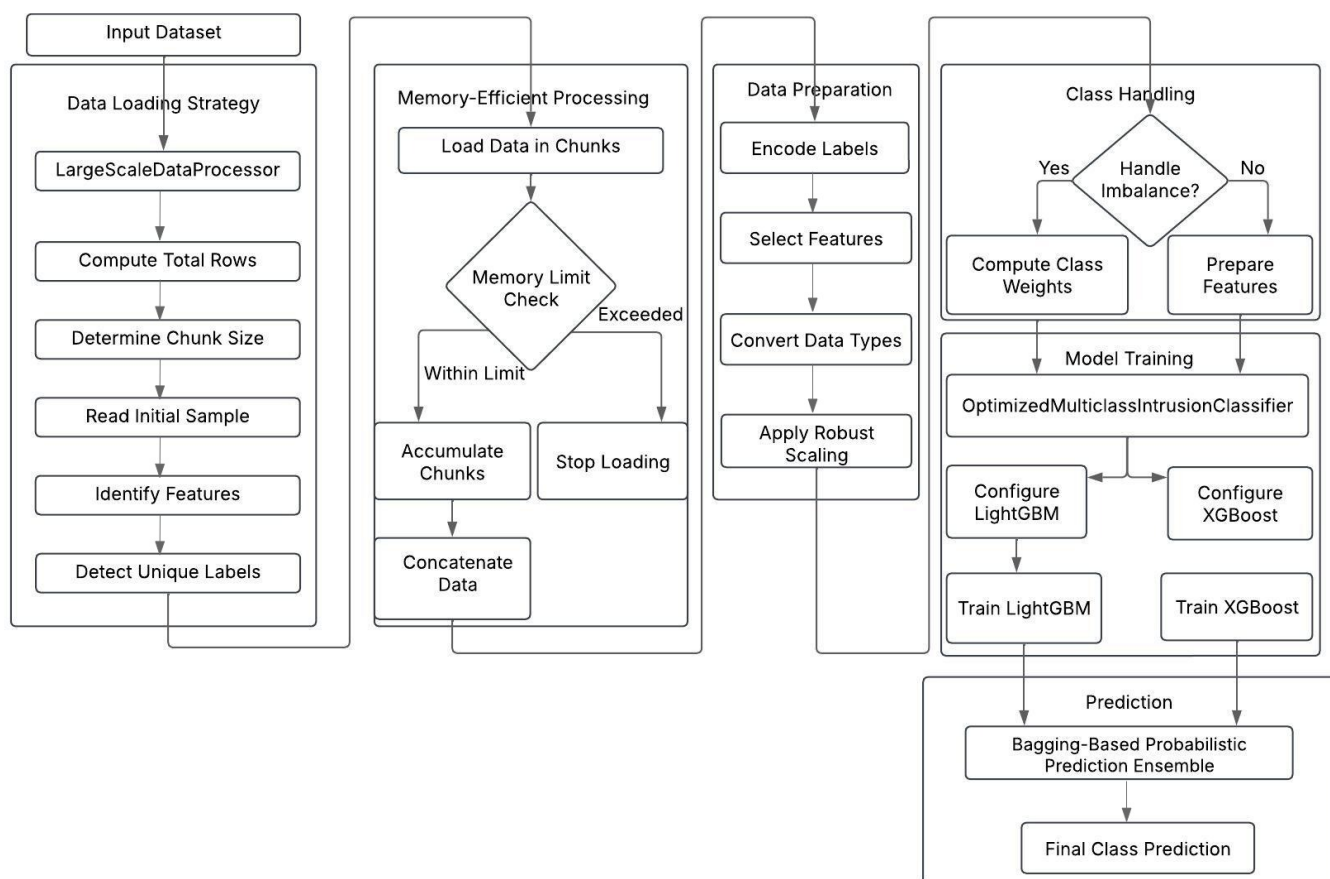


Figure 1. OMIC architecture

3.2. Data Processing Module

The data processing module forms the cornerstone of the OMIC framework, implementing a sophisticated streaming architecture specifically engineered for handling large-scale network traffic data. This module operates through a dynamic chunk-based processing system at its core, demonstrating remarkable adaptability to available system resources and data characteristics. The streaming architecture enables continuous processing of incoming network traffic while maintaining optimal performance levels through intelligent resource allocation. This approach allows the system to handle massive data volumes without compromising processing speed or accuracy, making it particularly suitable for real-time network security applications. The module's adaptive nature ensures that processing parameters are automatically adjusted based on current system conditions and data flow patterns, maximizing efficiency while preventing system overload.

3.2.1. Data Type and Preprocessing Optimization

Data type optimization within the OMIC framework represents a sophisticated approach to memory management and computational efficiency. The implementation incorporates advanced type optimization techniques that substantially reduce memory footprint while maintaining high numerical precision through carefully designed automatic type inference and conversion mechanisms. The preprocessing pipeline implements critical optimization strategies through several key components, as outlined in Table 1. The RobustScaler implementation plays a crucial role in feature scaling and normalization, proving particularly effective in managing the outliers that frequently appear in network traffic patterns. This robust scaler ensures that unusual traffic patterns, which might indicate potential security threats, are preserved while maintaining optimal data distribution for ML algorithms. The system continuously monitors data characteristics and automatically adjusts type conversions to maintain the optimal balance between precision and resource utilization. These optimization techniques operate seamlessly in the background, ensuring that data processing remains efficient without requiring manual intervention or adjustment.

Table 1. Preprocessing component.

Parameter	Configuration	Objective	Optimization
LabelEncoder	Default	Label encoding	Converts to indices
RobustScaler	copy=False	Normalization	Memory savings
Type Conversion	float64→float32, int64→int32	Memory reduction	~50% memory gain

3.2.2. Memory Management

The memory management subsystem exemplifies sophisticated resource control through comprehensive monitoring and preemptive overflow prevention. The system implements multiple complementary strategies to ensure optimal memory utilization while maintaining processing efficiency. Dynamic garbage collection triggers are intelligently implemented to release unused resources at optimal intervals, preventing memory fragmentation while minimizing processing interruptions. The temporary data caching mechanism employs sophisticated algorithms to determine which data should be kept in rapid-access memory and which can be temporarily stored in more efficient long-term storage solutions. Adaptive processing rate adjustments are continuously calculated based on current memory usage patterns, system load, and incoming data characteristics. The module's data validation and cleaning mechanisms operate through optimized single-pass operations, efficiently handling missing values, outlier detection, and data consistency validation across features. This comprehensive approach ensures that memory usage remains optimal while maintaining the integrity and accessibility of critical security-related data.

3.3. System Resource Management

The system resource management framework operates within a precisely defined 16GB memory constraint, as specified in the Base System Parameters (Table 2) while maintaining optimal performance through sophisticated resource allocation strategies. Implementing adaptive chunk sizing represents a critical advancement in efficient data processing, with the dynamic configuration adjusting to dataset characteristics while maintaining system stability. The chunk size formula, defined as $\min(\text{total\_rows}, 500000)$, as shown in Table 2, exemplifies the framework's intelligent approach to resource utilization. This adaptive methodology ensures that processing remains efficient regardless of dataset size variations, automatically adjusting chunk sizes to optimize memory usage without compromising processing speed or accuracy. The system continuously monitors resource utilization across all components, implementing preemptive measures to prevent resource exhaustion while maintaining consistent processing throughput. This sophisticated resource management approach enables the framework to handle varying workloads while maintaining optimal performance within specified hardware constraints.

Table 2. Base system parameters.

Parameter	Value	Description	Impact
memory_limit_gb	16	Memory limit in GB	Prevents OOM crashes
chunk_size	$\min(\text{total\_rows}, 500000)$	Data batch size	Optimizes loading
sample_fraction	Optional (None)	Fraction to sample	Reduces data volume

3.4. Classification Module

The classification module represents the critical intelligence component of the OMIC framework, implementing a sophisticated ensemble learning approach that combines the strengths of LightGBM and XGBoost classifiers. This orchestrated configuration maximizes the complementary advantages of both algorithms, with LightGBM's leaf-wise growth strategy excelling in handling sparse features common in network traffic data, while XGBoost's lev-el-wise growth approach ensures robust generalization capabilities. The module incorporates sophisticated weighting mechanisms that dynamically adjust classifier contributions based on performance characteristics across different attack classes. This adaptive approach ensures optimal classification performance across various network threats while maintaining

efficient resource utilization. The implementation includes comprehensive parameter optimization strategies that balance accuracy, processing speed, and memory efficiency through automated tuning mechanisms that adapt to dataset characteristics and system constraints.

3.4.1. XGBoost Configuration

The XGBoost implementation within the classification module demonstrates a carefully calibrated approach to model training and optimization. As detailed in the XGBoost Main Parameters (Table 3), the configuration utilizes one hundred estimators with a precisely tuned learning rate of 0.1, representing an optimal balance between model complexity and training efficiency. As shown in Table 4, the regularization strategy combines L1 and L2 regularization, both set at 0.1, with carefully tuned gamma and minimum child weight parameters to prevent overfitting while maintaining model responsiveness to emerging attack patterns. The implementation employs sophisticated optimization through row and column subsampling strategies configured at 0.8, as specified in Table 5, ensuring diverse feature representation while maintaining model stability.

Table 3. XGBoost main parameters.

Parameter	Value	Impact	Optimization
n_estimators	100	Number of trees	Balances accuracy/time
learning_rate	0.1	Learning speed	Prevents overfitting
max_depth	15	Tree depth	Balances complexity/generalization

Table 4. XGBoost regularization parameters.

Parameter	Value	Objective	Impact
min_child_weight	1	Controls overfitting	Model stability
gamma	0.1	Pruning	Complexity reduction
reg_alpha	0.1	L1 regularization	Sparsity
reg_lambda	0.1	L2 regularization	Stability

Table 5. XGBoost performance parameters.

Parameter	Value	Objective	Impact
subsample	0.8	Row subsampling	Reduces overfitting
colsample_bytree	0.8	Column subsampling	Tree diversity
tree_method	'hist'	Construction method	Calculation acceleration
n_jobs	-1	Parallelization	Uses all CPUs

3.4.2. LightGBM Configuration

The LightGBM configuration mirrors several core parameters from XGBoost while implementing unique optimizations specific to the LightGBM architecture. The LightGBM Main Parameters (Table 6) show that the implementation maintains identical estimator count and learning rate settings to enable direct performance comparisons. The feature fraction and bagging fraction parameters are precisely tuned at 0.8, with a bagging frequency of 5, as detailed in Table 7, ensuring regular pattern sampling and optimal model diversity. The Performance Optimization parameters (Table 8) demonstrate the implementation's efficiency focus, matching regularization parameters and full CPU utilization.

Table 6. LightGBM main parameters.

Parameter	Value	Impact	Optimization
n_estimators	100	Number of trees	Balances accuracy/time
learning_rate	0.1	Learning speed	Prevents overfitting
max_depth	15	Tree depth	Balances complexity/generalization
num_leaves	31	Tree structure	Controls complexity

Table 7. LightGBM regularization parameters.

Parameter	Value	Objective	Impact
feature_fraction	0.8	Feature subsampling	Reduces overfitting
bagging_fraction	0.8	Data subsampling	Model diversity
bagging_freq	5	Bagging frequency	Learning stability
min_child_samples	20	Min leaf size	Robustness

Table 8. LightGBM performance parameters.

Parameter	Value	Objective	Impact
reg_alpha	0.1	L1 regularization	Sparsity
reg_lambda	0.1	L2 regularization	Stability
n_jobs	-1	Parallelization	Uses all CPUs
random_state	42	Reproducibility	Consistent results

3.5. Bagging Ensemble Methodology in OMIC

The ensemble methodology in OMIC implements a sophisticated bagging-based fusion strategy that leverages the strengths of both LightGBM and XGBoost classifiers. Each base classifier employs bagging techniques at multiple levels: at the observation level through bagging_fraction and subsample parameters and at the feature level through feature_fraction and colsample_bytree parameters. These bagged predictions are then combined through a probability averaging mechanism, where the final prediction probabilities are computed as the arithmetic mean of the probabilities produced by each bagged classifier.

The use of bagging in this architecture provides several crucial advantages. It helps reduce overfitting by introducing randomization at both the data and feature levels. The consistent bagging parameters across both models (0.8 for both data and feature sampling) ensure a balanced approach to variance reduction. Combining bagging with probability averaging creates a particularly robust ensemble that benefits from variance reduction through random sampling and uncertainty quantification through probability estimates.

The decision to implement bagging rather than ensemble methods like stacking or simple voting was driven by its particular advantages in the network security context. Bagging protects against outliers and noisy data points that are common in network traffic. Random sampling of observations and features helps the system maintain robustness against various types of network attacks while reducing the impact of any anomalous pattern.

This bagging ensemble architecture demonstrates superior characteristics in handling concept drift, as each bagged model provides an independent view of the data, making the ensemble more robust to changes in attack patterns. The system's ability to combine bagged predictions through probability averaging further enhances its capability to detect novel threats, as it can leverage the diverse perspectives provided by different random subsets of the data and features.

3.6. Monitoring System Integration

The monitoring system integration provides comprehensive oversight of framework operations through sophisticated real-time tracking mechanisms. As the Monitoring System specifications (Table 9) outline, implementing tqdm progress bars enables detailed visibility into processing status, while comprehensive class distribution reporting ensures balanced learning across all attack categories. The memory alert system, detailed in Table 9, implements sophisticated resource monitoring that provides proactive protection against resource exhaustion, maintaining system stability during extended processing operations. This monitoring framework continuously tracks system performance metrics, providing detailed insights into resource utilization and processing efficiency. The system implements sophisticated alerting mechanisms that provide early warning of potential resource constraints or processing bottlenecks, enabling preemptive optimization of system parameters to maintain optimal performance.

Table 9. Monitoring system.

Metric	Type	Objective	Usage
Progress bars	tqdm	Real-time tracking	Progress monitoring
Class distribution	Report	Class balance	Dataset diagnostics
Memory alerts	Warning	Memory monitoring	Crash prevention

4. Benefits of the OMIC Model for the Internet of Things

Implementing the OMIC model in the Internet of Things (IoT) context presents significant advantages, particularly considering the inherent constraints in this domain. The presented algorithm, with its `LargeScaleDataProcessor` and `OptimizedMulticlassIntrusion-Classifer` classes, offers an approach well-suited to IoT environments thanks to several key characteristics. First, optimized memory management through chunk processing and a garbage collector (gc) allows for efficient operation even with IoT devices with limited memory resources. This progressive data-loading approach prevents memory saturation, a crucial aspect for IoT devices with restricted storage capabilities.

As implemented in this algorithm, the combination of XGBoost and LightGBM offers an interesting balance between performance and resource consumption. LightGBM, with its leaf-wise approach in tree construction, allows for more efficient memory usage and faster processing time, which is particularly beneficial for battery-powered IoT devices. In parallel, XGBoost brings its robustness and precision, while the hybridization of both models allows for compensating their respective weaknesses while maintaining a reasonable memory footprint thanks to the implemented optimizations.

The scalability aspect is also handled remarkably well in this implementation. Data processing by chunks, combined with automatic data type conversion to save memory (conversion from float64 to float32 and int64 to int32), allows for intelligent adaptation to IoT devices' hardware constraints. This approach is particularly relevant in IoT networks where resources are heterogeneous, and energy efficiency is paramount.

The management of class imbalance, often present in IoT data (for example, anomaly detection), is efficiently handled thanks to the integrated class weighting system. This feature is particularly important in IoT applications where certain rare but critical events must be correctly detected, such as equipment failures or abnormal behaviors in a sensor network.

The integrated parallelism in both models (with `n_jobs=-1`) also allows for optimal use of available computing resources, thus automatically adapting to the IoT device's capabilities. Furthermore, using the 'hist' method in XGBoost and LightGBM's optimized parameters reduces energy consumption while maintaining high performance, a crucial aspect for battery-powered IoT devices.

The OMIC approach also allows for better robustness against variations in input data, a particularly important aspect in IoT where environmental conditions can affect the quality of collected data. The averaging of predictions from both models offers a form of natural regularization, thus reducing the risk of overfitting and increasing the reliability of predictions in real-world usage conditions.

5. Performance Analysis

5.1. Experimental Setup

The experimental evaluation of OMIC was conducted using the comprehensive CIIoT2023 dataset, which represents a significant advancement in IoT security research. This dataset, developed by the Canadian Institute of Cybersecurity at the University of New Brunswick, presents unique challenges that distinguish it from existing IoT security datasets. It simulates a complex real-world IoT environment with an extensive topology of 105 IoT devices, offering a more realistic and challenging testing ground for IDS.

The dataset's complexity is particularly noteworthy in several aspects. First, it encompasses 1,048,575 records covering 33 distinct types of attacks, classified into seven sophisticated categories: DDoS, Brute Force, Spoofing, DoS, Recon, Web-based, and Mirai. This extensive attack diversity poses a significant challenge for detection models, including both traditional network attacks and IoT-specific threats. Many of these attacks, particularly in the

Web-based and Recon categories, represent sophisticated intrusion techniques that are notoriously difficult to detect due to their subtle nature and similarity to legitimate traffic patterns.

A distinctive characteristic of the CICIoT2023 dataset is its realistic representation of modern IoT security challenges. All attacks are executed by malicious IoT devices targeting other IoT devices, accurately reflecting real-world attack scenarios in IoT environments. This aspect is crucial as it captures the unique network behavior and attack patterns specific to IoT ecosystems, which differ significantly from traditional network attacks. Furthermore, the dataset presents a significant class imbalance challenge, with attack traffic comprising 97.67% of the total volume, unevenly distributed among different attack types. This imbalance reflects real-world scenarios where certain types of attacks are more prevalent than others, making it particularly challenging for detection models to maintain high accuracy across all attack categories.

To ensure rigorous evaluation, the experiments were conducted on a computer with an Intel(R) Core(TM) i7-6500U CPU clocked at 2.50GHz, featuring a 1024GB hard drive and 16GB of RAM. The machine ran Windows 10 Professional with a base speed of 2.60GHz, having two cores and four logical processors.

The dataset's inclusion of multiple attacks not available in other IoT datasets, combined with its complex topology and diverse device types, makes it particularly valuable for developing and testing advanced security analytics solutions. This comprehensive nature of the dataset allows for a more thorough evaluation of detection models' capabilities in handling sophisticated and emerging IoT security threats, providing a more realistic assessment of their practical applicability in modern IoT environments.

5.2. Traffic Distribution

The frequency histogram provides an overview of the distribution of different types of attacks in the dataset. This visual representation helps identify the most common and the least frequent attacks. As shown in Figure 2, specific attacks, such as DDoS ICMP Flood, DoS UDP Flood, DoS TCP Flood, and DDoS PSHACK Flood, occur significantly more often, making up a substantial portion of the dataset. Their high prevalence highlights their importance in intrusion scenarios.

While less frequent, other attacks are still sufficiently represented to be noteworthy. This includes, for example, DDoS SYN Flood and Mirai udpplain. Conversely, some attacks, such as Uploading Attack, Recon PingSweep, and XSS, are particularly rare in the dataset.

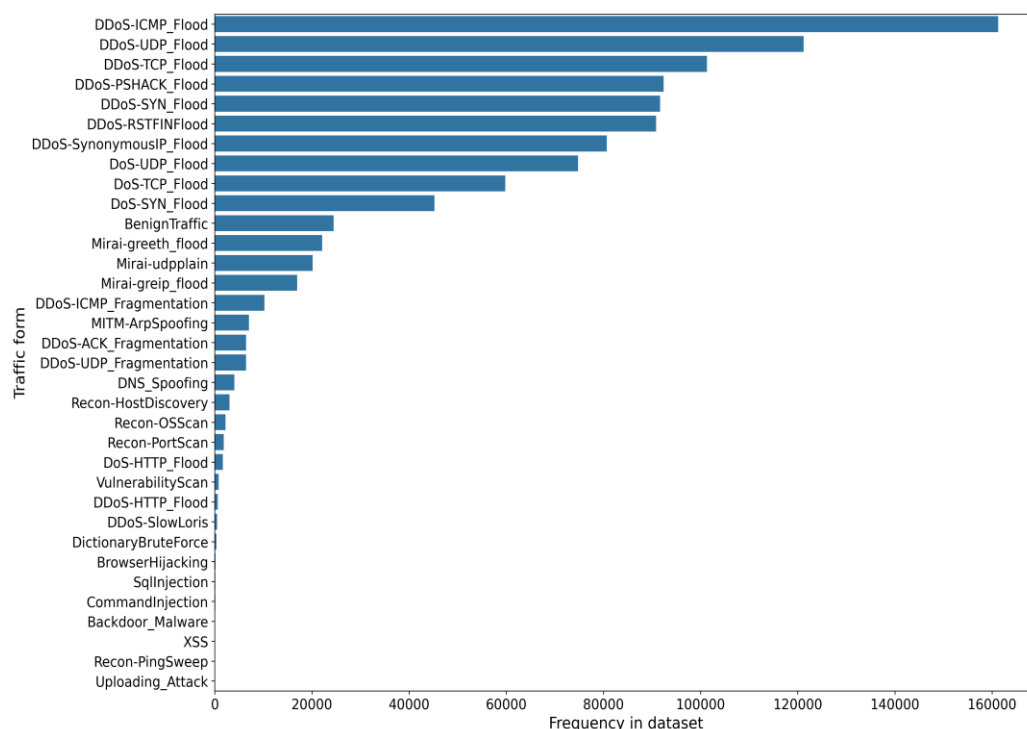


Figure 2. Traffic distribution

5.3. Results and Discussion

The framework OMIC demonstrated exceptional performance across multiple evaluation dimensions, achieving an overall accuracy of 99.26% across all attack classes and significantly outperforming baseline methods. The most notable improvements were observed in detecting minority attack classes, where OMIC showed significant F1-score improvement compared to the best-performing baseline method.

Examining the confusion matrix in Figure 3, we observe an excellent ability to discriminate between different attack classes. The concentration of optimal values (1.00) on the main diagonal indicates near-perfect precision in identifying the majority of attack types, while the majority presence of null values outside the main diagonal demonstrates a very low rate of false positives and false negatives. Some values between 0.5 and 1.00 on the main diagonal indicate satisfactory but slightly lower performance for specific classes. This can be attributed to the inherent complexity of certain attack types sharing common characteristics.

The analysis of ROC-AUC curves, illustrated in Figure 4, further reinforces the model's robust performance. The achievement of a 1.00 AUC value for almost all classes, with a single class at 0.99, represents a significant accomplishment in classification. The proximity of ROC curves to the upper left corner of the graph confirms the model's excellent balance between sensitivity and specificity.

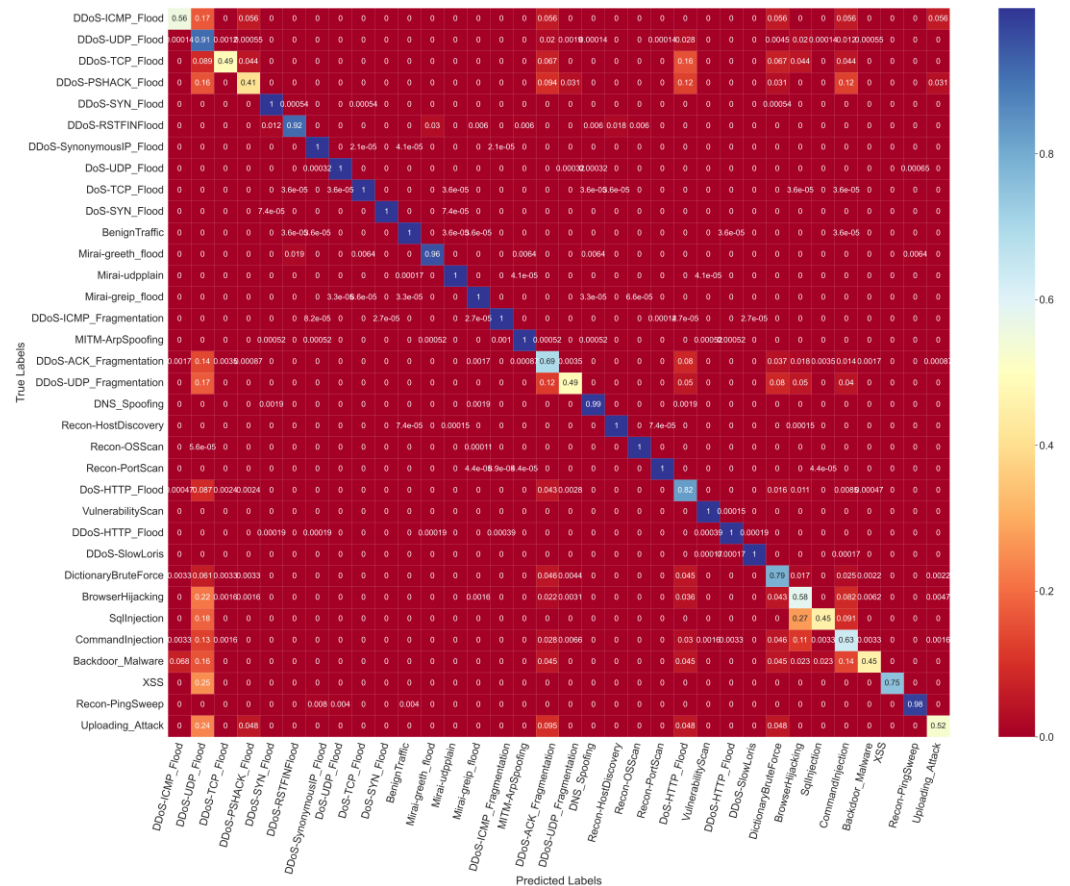


Figure 3. Confusion Matrix With OMIC

In terms of specific attack categories, the system demonstrated particularly strong performance in detecting volumetric attacks. For DDoS attacks, the framework achieved perfect precision, recall, and F1-scores of 1.00 for multiple subtypes, including ACK Fragmentation, UDP Flood, ICMP Flood, RSTFIN Flood, and others. Only SlowLoris and HTTP Flood showed slightly lower performance with F1-scores of 0.96 and 0.94, respectively, due to their sophisticated traffic patterns mimicking legitimate requests. Similarly, DoS attacks demonstrated excellent detection rates, with perfect scores for TCP Flood, SYN Flood, and UDP Flood, while HTTP Flood maintained a high F1-score of 0.99. Mirai-based attacks, including GREIP Flood, Greeth Flood, and UDPPlain, all achieved perfect scores of 1.00.

However, detailed analysis, as shown in Figure 5, reveals specific areas with notable limitations. In the reconnaissance category, Recon_Ping_Sweep attacks achieved only a 0.40 F1-score, with low precision (0.36) and recall (0.45), indicating fundamental challenges in distinguishing legitimate ping activities from malicious scanning attempts. Recon_Port_Scan and Recon_OSScan detection showed suboptimal performance with F1-scores of 0.64 and 0.57, respectively.

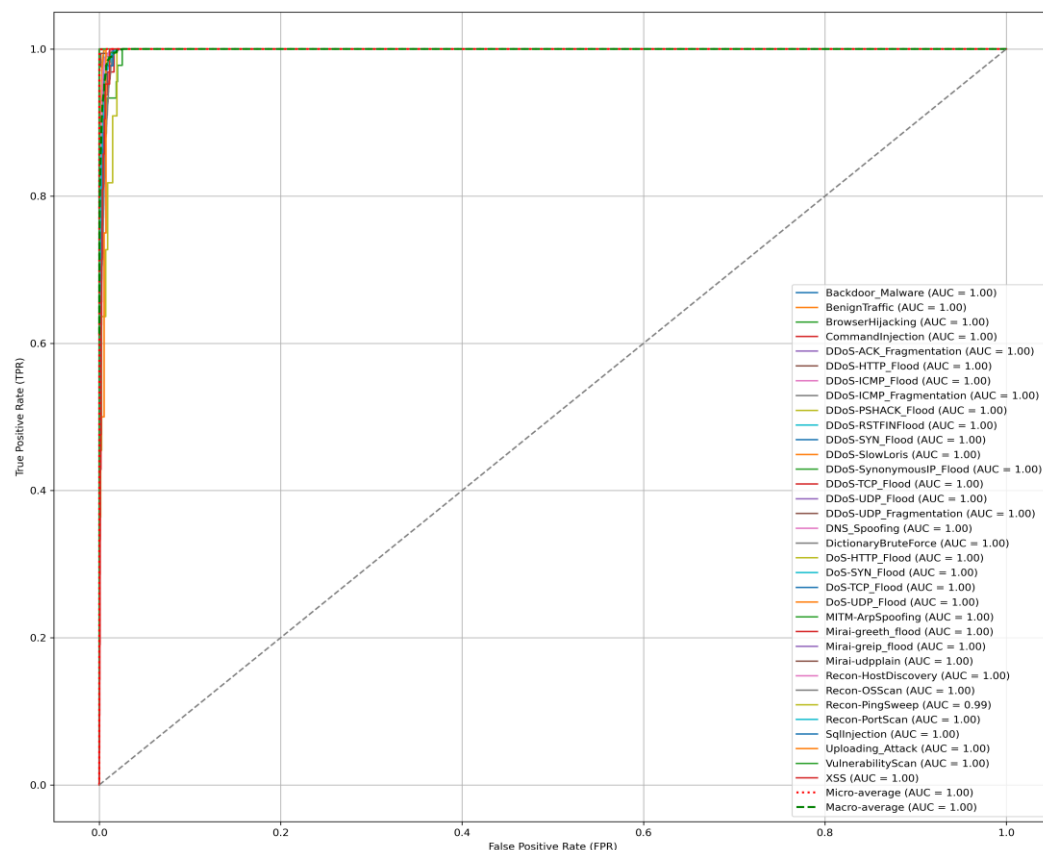


Figure 4. Receiver operating characteristic for OMIC

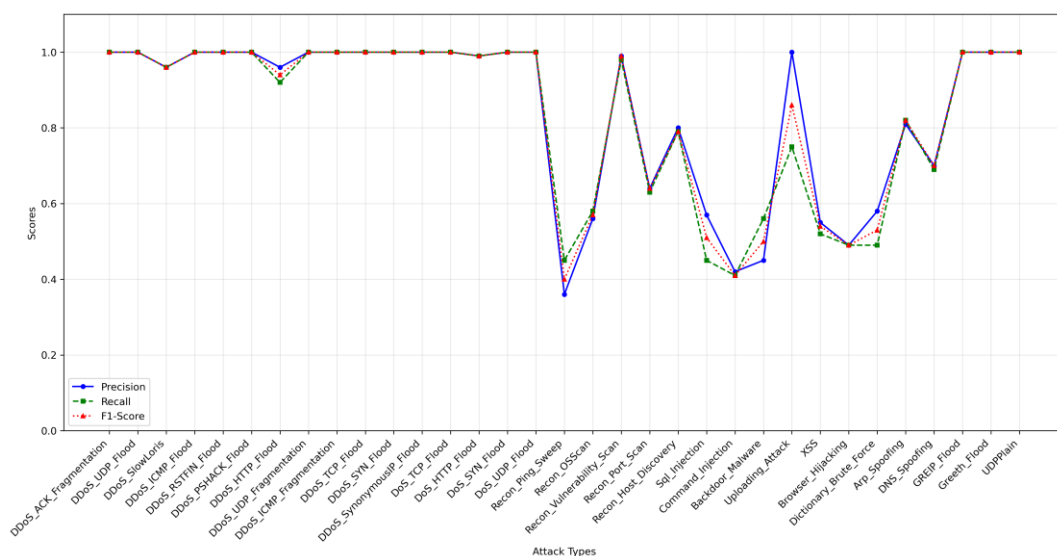


Figure 5. Evaluation of OMIC model

Web-based attacks presented another challenging area. Command Injection attacks showed poor detection rates with an F1-score of 0.41, while SQL Injection and XSS attacks

performed marginally better with F1-scores of 0.51 and 0.54, respectively. Browser-based and malware attacks also demonstrated detection challenges, with Browser Hijacking and Backdoor Malware detection achieving F1-scores of 0.49 and 0.50, respectively. The Dictionary Brute Force attacks showed concerning performance levels with an F1-score of 0.53, although Uploading Attack detection performed better with an F1-score of 0.86.

The analysis of spoofing attacks revealed varying detection capabilities, with Arp Spoofing achieving an F1-score of 0.82 and DNS Spoofing showing more limitations at 0.70, suggesting better handling of link-layer attacks than application-layer spoofing attempts.

These performance variations can be attributed to several factors: the sophisticated nature of modern attacks, often involving patterns resembling legitimate traffic, the relative scarcity of certain attack types in the training data, and the dynamic, evolving nature of these attacks, particularly in web-based scenarios.

Despite these limitations, OMIC's comprehensive performance metrics demonstrate its capability to handle large-scale network security analysis while maintaining high accuracy and efficiency across various operational conditions. The framework shows particular strength in detecting volumetric attacks while identifying areas for future improvement in more sophisticated attack detection mechanisms.

5.3. Comparative Analysis of OMIC Model Performance

The comparative analysis of performance across different classification models found in the literature, using the same performance metric aggregation method (Macro Average), reveals particularly interesting results regarding the OMIC model (Table 10).

Table 10. Performance comparison of the OMIC Model with other classification models.

Ref	Models	Accuracy	Recall	Precision	F1-score
[39]	Perceptron	0.8195	0.5075	0.4546	0.4472
	Adaboost	0.6078	0.6076	0.4796	0.4734
	Deep Neural Network	0.9861	0.7318	0.6652	0.6723
[26]	Naive Bayes (Gaussian)	0.5896	0.3996	0.4387	0.3069
	ANN	0.9803	0.6310	0.7006	0.6469
	Logistic Regression	0.8022	0.4832	0.6218	0.4887
	k-NN	0.9466	0.6288	0.6826	0.6395
	XGBoost	0.9926	0.7169	0.7960	0.7291
	Random Forest	0.9929	0.7219	0.8230	0.7355
	LightGBM	0.3601	0.1103	0.1206	0.0824
	GRU	0.9687	0.6001	0.6166	0.5926
	LSTM	0.9760	0.6179	0.6237	0.6116
Proposed	CNN	0.9833	0.6472	0.7118	0.6520
	OMIC	0.9926	0.8359	0.8429	0.8386

Upon thoroughly examining the results, the OMIC model demonstrates exceptional performance with an accuracy of 99.26%, matching XGBoost and showing comparable performance to RF (99.29%). This excellence is particularly notable in the model's balanced performance metrics, with a recall of 83.59% and precision of 84.29%, demonstrating a superior ability to correctly identify positive cases while maintaining a low false positive rate.

The comparison with other approaches reveals significant performance gaps. Traditional models like Perceptron and LR achieve accuracies of 81.95% and 80.22%, respectively, showing considerably lower performance across all metrics. The OMIC model's recall (83.59%) substantially outperforms these approaches, with improvements of 32.84 and 35.27 percentage points, respectively, compared to Perceptron (50.75%) and LR (48.32%).

Compared to more sophisticated DL approaches, the OMIC model maintains its competitive edge. While Deep Neural Networks achieve a high accuracy of 98.61%, their recall (73.18%) and precision (66.52%) fall notably short of OMIC's balanced performance. Similarly, modern architectures like GRU, LSTM, and CNN, despite their high accuracy (96.87%, 97.60%, and 98.33%, respectively), demonstrate lower performance in recall and precision metrics.

A particularly noteworthy aspect is the internal consistency of OMIC's performance metrics. The F1-score of 83.86% reflects an excellent balance between precision (84.29%) and recall (83.59%), suggesting highly stable and reliable performance. This harmonious balance between metrics distinguishes OMIC from other high-accuracy models like RF and XGBoost, which show slightly lower F1-scores of 73.55% and 72.91%, respectively, despite similar accuracy levels.

Some models, such as LightGBM, demonstrate significantly lower performance across all metrics (accuracy: 36.01%, recall: 11.03%), highlighting the robustness of the OMIC approach. Traditional methods like Naïve Bayes also show limited performance (accuracy: 58.96%, F1-score: 30.69%), further emphasizing the advantages of the OMIC architecture.

In conclusion, the OMIC model demonstrates strong performance across all evaluated metrics, particularly in maintaining the balance between precision and recall. While sharing top-tier accuracy with models like RF and XGBoost, it achieves superior balanced performance in other metrics. This comprehensive strong performance across all metrics positions OMIC as a significant advancement in classification techniques, particularly in scenarios requiring both high accuracy and balanced prediction capabilities.

6. Conclusions

This research introduces OMIC as a significant advancement in network intrusion detection for IoT environments. The experimental results demonstrate remarkable achievements in several key areas. First, OMIC's performance surpasses traditional ML and DL approaches across most attack categories, particularly excelling in DDoS and DoS detection with perfect F1-scores. The framework's ability to handle class imbalance and maintain high detection rates for minority attack classes represents a substantial improvement over existing solutions.

The findings directly address our initial research objectives of developing a more robust and efficient approach to network intrusion detection. OMIC's memory-optimized pipeline and ensemble architecture successfully overcome the challenges of processing large-scale network traffic while maintaining high detection accuracy. Compared to baseline models, the framework's superior performance in detecting complex attacks like DNS Spoofing and Reconnaissance validates our architectural decisions and optimization strategies.

From a practical perspective, OMIC's contributions impact academic research and industrial applications in network security. The framework's ability to process large-scale data efficiently while maintaining high accuracy makes it particularly valuable for real-world IoT deployments. Its modular architecture provides a foundation for future enhancements and adaptations to emerging threat landscapes.

However, several limitations warrant attention. The framework shows relatively modest performance detecting certain web-based attacks, particularly SQL injections and XSS attacks. Some reconnaissance attacks, notably Ping Sweep, remain challenging to detect. Future research should focus on enhancing detection capabilities for these attack types, possibly through specialized feature engineering or hybrid detection approaches. Investigating real-time adaptation mechanisms and response strategies would also strengthen the framework's practical utility in dynamic threat environments.

Author Contributions: Conceptualization: Jean Pierre Ntayagabiri; methodology, Jean Pierre Ntayagabiri; software: Jean Pierre Ntayagabiri.; validation: Jean Pierre Ntayagabiri, Hind El Makhtoum, Youssef Bentaleb and Jeremie Ndikumagenge; formal analysis: Jean Pierre Ntayagabiri; investigation: Jean Pierre Ntayagabiri and Hind El Makhtoum.; writing original draft preparation: Jean Pierre Ntayagabiri.; writing review and editing: Jean Pierre Ntayagabiri.; supervision: Youssef Bentaleb and Jeremie Ndikumagenge. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset used is a public dataset that was published by [39]. Direct access to its official repository is available at: <https://www.unb.ca/cic/datasets/index.html>.

Conflicts of Interest: The authors declare no conflict of interest.

References

- [1] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. EL Makhtoum, "A Comprehensive Approach to Protocols and Security in Internet of Things Technology," *J. Comput. Theor. Appl.*, vol. 2, no. 3, pp. 324–341, Feb. 2024, doi: 10.62411/jcta.11660.
- [2] A. K. Al Hwaitat *et al.*, "Overview of Mobile Attack Detection and Prevention Techniques Using Machine Learning," *Int. J. Interact. Mob. Technol.*, vol. 18, no. 10, Jan. 2024, [Online]. Available: <https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=18657923&AN=177404208&h=FMA26sPUnglaFNBoZDo0VKVC1B1UcyIlgk5vY%2BfgVbKNTjU37%2Bo6BoImIcmftFsHPv%2BCsGxEo53CjCAuFLAAhaA%3D%3D&crl=c>
- [3] J. Mao, X. Yang, B. Hu, Y. Lu, and G. Yin, "Intrusion Detection System Based on Multi-Level Feature Extraction and Inductive Network," *Electronics*, vol. 14, no. 1, p. 189, Jan. 2025, doi: 10.3390/electronics14010189.
- [4] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, p. 100306, Jan. 2023, doi: 10.1016/j.array.2023.100306.
- [5] N. Jeffrey, Q. Tan, and J. R. Villar, "A Review of Anomaly Detection Strategies to Detect Threats to Cyber-Physical Systems," *Electronics*, vol. 12, no. 15, p. 3283, Jan. 2023, doi: 10.3390/electronics12153283.
- [6] 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.
- [7] B. R. Kikissagbe and M. Adda, "Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review," *Electronics*, vol. 13, no. 18, p. 3601, Jan. 2024, doi: 10.3390/electronics13183601.
- [8] R. Chinnasamy, M. Subramanian, S. V. Easwaramoorthy, and J. Cho, "Deep Learning-driven Methods for Network-based Intrusion Detection Systems: A Systematic Review," *ICT Express*, Jan. 2025, doi: 10.1016/j.icte.2025.01.005.
- [9] Y. Wu, B. Zou, and Y. Cao, "Current Status and Challenges and Future Trends of Deep Learning-Based Intrusion Detection Models," *J. Imaging*, vol. 10, no. 10, p. 254, Jan. 2024, doi: 10.3390/jimaging10100254.
- [10] D. Patil, N. L. Rane, P. Desai, and J. Rane, "Machine learning and deep learning: Methods, techniques, applications, challenges, and future research opportunities," in *Trustworthy Artificial Intelligence in Industry and Society*, Deep Science Publishing, 2024. doi: 10.70593/978-81-981367-4-9_2.
- [11] O. Alshboul, G. Almasabha, A. Shehadeh, and K. Al-Shboul, "A comparative study of LightGBM, XGBoost, and GEP models in shear strength management of SFRC-SBWS," *Structures*, vol. 61, p. 106009, Feb. 2024, doi: 10.1016/j.jistruc.2024.106009.
- [12] H. Hosamo and S. Mazzetto, "Performance Evaluation of Machine Learning Models for Predicting Energy Consumption and Occupant Dissatisfaction in Buildings," *Buildings*, vol. 15, no. 1, p. 39, Feb. 2025, doi: 10.3390/buildings15010039.
- [13] M. Markevych and M. Dawson, "A Review of Enhancing Intrusion Detection Systems for Cybersecurity Using Artificial Intelligence (AI)," *Int. Conf. KNOWLEDGE-BASED Organ.*, vol. 29, no. 3, pp. 30–37, Jan. 2023, doi: 10.2478/kbo-2023-0072.
- [14] M. M. Ahsan, M. S. Ali, and Z. Siddique, "Enhancing and improving the performance of imbalanced class data using novel GBO and SSG: A comparative analysis," *Neural Networks*, vol. 173, p. 106157, May 2024, doi: 10.1016/j.neunet.2024.106157.
- [15] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artif. Intell. Rev.*, vol. 57, no. 6, p. 137, Jan. 2024, doi: 10.1007/s10462-024-10759-6.
- [16] B. Pes, "Learning from High-Dimensional and Class-Imbalanced Datasets Using Random Forests," *Information*, vol. 12, no. 8, p. 286, Jan. 2021, doi: 10.3390/info12080286.
- [17] B. Xu and G. Yang, "Interpretability research of deep learning: A literature survey," *Inf. Fusion*, vol. 115, p. 102721, Jan. 2025, doi: 10.1016/j.inffus.2024.102721.
- [18] M. Ibrahim and A. Al-Wadi, "Enhancing IoMT network security using ensemble learning-based intrusion detection systems," *J. Eng. Res.*, Feb. 2024, doi: 10.1016/j.jer.2024.12.003.
- [19] V. Ravi, T. D. Pham, and M. Alazab, "Deep learning-based network intrusion detection system for Internet of medical things," *IEEE internet things Mag.*, vol. 6, no. 2, pp. 50–54, Jan. 2023, [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10145040/>
- [20] R. Devendiran and A. V. Turukmane, "Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy," *Expert Syst. Appl.*, vol. 245, p. 123027, Jul. 2024, doi: 10.1016/j.eswa.2023.123027.
- [21] F. Sun *et al.*, "Efficiency of Extreme Gradient Boosting for Imbalanced Land Cover Classification Using an Extended Margin and Disagreement Performance," *ISPRS Int. J. Geo-Information*, vol. 8, no. 7, p. 315, Jul. 2019, doi: 10.3390/ijgi8070315.
- [22] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," *Expert Syst. Appl.*, vol. 244, p. 122778, Feb. 2024, doi: 10.1016/j.eswa.2023.122778.
- [23] D. R. I. M. Setiadi, S. Widiono, A. N. Safriandono, and S. Budi, "Phishing Website Detection Using Bidirectional Gated Recurrent Unit Model and Feature Selection," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 75–83, Jul. 2024, doi: 10.62411/faith.2024-15.
- [24] S. Gul, S. Arshad, S. M. U. Saeed, A. Akram, and M. A. Azam, "WGAN-DL-IDS: An Efficient Framework for Intrusion Detection System Using WGAN, Random Forest, and Deep Learning Approaches," *Computers*, vol. 14, no. 1, p. 4, Jan. 2024, [Online]. Available: <https://www.mdpi.com/2073-431X/14/1/4>
- [25] W. F. Kamil, "Adapted CNN-SMOTE-BGMM deep learning framework for network intrusion detection using unbalanced dataset," *Iraqi J. Sci.*, pp. 4846–4864, Jan. 2023, [Online]. Available: <https://www.iasj.net/iasj/download/8ce08ee0ce878309>
- [26] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. El Makhtoum, "A Comparative Analysis of Supervised Machine Learning Algorithms for IoT Attack Detection and Classification," *J. Comput. Theor. Appl.*, vol. 2, no. 3, pp. 395–409, Feb. 2025, doi: 10.62411/jcta.11901.

- [27] T. Al-Shehari and R. A. Alsowail, "Random resampling algorithms for addressing the imbalanced dataset classes in insider threat detection," *Int. J. Inf. Secur.*, vol. 22, no. 3, pp. 611–629, Jan. 2023, doi: 10.1007/s10207-022-00651-1.
- [28] K. S. Babu and Y. N. Rao, "MCGAN: modified conditional generative adversarial network (MCGAN) for class imbalance problems in network intrusion detection system," *Appl. Sci.*, vol. 13, no. 4, p. 2576, Jan. 2023, [Online]. Available: <https://www.mdpi.com/2076-3417/13/4/2576>
- [29] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *J. Supercomput.*, vol. 79, no. 10, pp. 10611–10644, Jan. 2023, doi: 10.1007/s11227-023-05073-x.
- [30] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for IoT network using ensemble-learning-based deep neural network," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11888–11895, Jan. 2023, [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10044208/>
- [31] A. Çetin and S. Öztürk, "Comprehensive Exploration of Ensemble Machine Learning Techniques for IoT Cybersecurity Across Multi-Class and Binary Classification Tasks," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 4, pp. 371–384, Feb. 2025, doi: 10.62411/faith.3048-3719-51.
- [32] Z. S. Dhahir, "A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLOF-Based Feature Engineering and XGBoost," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 174–190, Sep. 2024, doi: 10.62411/faith.2024-33.
- [33] N. Rane, S. P. Choudhary, and J. Rane, "Ensemble deep learning and machine learning: applications, opportunities, challenges, and future directions," *Stud. Med. Heal. Sci.*, vol. 1, no. 2, pp. 18–41, Jan. 2024, [Online]. Available: <https://sabapub.com/index.php/SMHS/article/view/1225>
- [34] F. Alserhani and A. Aljared, "Evaluating Ensemble Learning Mechanisms for Predicting Advanced Cyber Attacks," *Appl. Sci.*, vol. 13, no. 24, p. 13310, Dec. 2023, doi: 10.3390/app132413310.
- [35] S. Arukonda, R. Cheruku, and V. Boddu, "Enhancing disease diagnosis accuracy and diversity through BA-TLBO optimized ensemble learning," *Biomed. Signal Process. Control*, vol. 96, p. 106507, Jan. 2024, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809424005652>
- [36] E. Itasoy, V. Rosenberg, N. Stavrakis, A. Dietrich, and C. Montanari, "Ransomware detection on windows using file system activity monitoring and a hybrid isolation forest-xgboost model," Jan. 2024, [Online]. Available: <https://www.researchsquare.com/article/rs-5257558/latest>
- [37] E. Batalov, P. Haverstock, R. Anderson, W. Thompson, and R. Wolverton, "Ransomware detection via network traffic analysis using isolation forest and lstm neural networks," Jan. 2024, [Online]. Available: <https://www.authorea.com/doi/full/10.22541/au.172928576.69686584>
- [38] A. Lumazine *et al.*, "Ransomware detection in network traffic using a hybrid cnn and isolation forest approach." Jan. 18, 2024. [Online]. Available: <https://www.authorea.com/doi/full/10.22541/au.172901014.44599790>
- [39] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023, doi: 10.3390/s23135941.