Journal of Computer Science

Research Paper

An Intrusion Detection Framework for MEC-Enabled IoT Networks

Rofaida Tawfik[1], Abdelfattah Hegazy[1], Hesham Dahshan[1] and Ahmed Gaber Abuabdallah[2]

[1]*Department of Computer Engineering, Arab Academy for Science, Technology and Maritime Transport, Egypt*
[2]*Department of Computer Science, Canadian International College, Egypt*

Article history
Received: 03-12-2025
Revised: 29-01-2026
Accepted: 02-03-2026

Corresponding Author:
Ahmed Gaber Abuabdallah
Department of Computer Science,
Canadian International College,
Egypt
Email: a_gaber_abuabdallah@cic-cairo.com

Abstract: Mobile Edge Computing (MEC) has emerged as a promising paradigm for supporting latency-sensitive Internet of Things (IoT) applications by bringing computational resources closer to data sources. However, the distributed nature of MEC environments increases exposure to network-based security threats, particularly network intrusions that can impact both system reliability and task offloading efficiency. This paper proposes a security-aware framework that integrates a machine learning–based Intrusion Detection System (IDS) with the DTOME (Dynamic Task Offloading with Hybrid Energy) scheme to enhance security in MEC-enabled IoT networks. A preprocessing security layer is deployed at the edge server to detect and filter malicious traffic before offloading decisions are executed. The proposed framework is evaluated using benchmark intrusion detection datasets and a comprehensive set of performance metrics. The results demonstrate robust detection performance and stable operation under edge computing constraints. The main contributions of this work include integrating machine learning–based intrusion detection with dynamic task offloading in MEC environments, conducting a multi-dataset experimental evaluation to improve result reliability, and highlighting the practical feasibility of intrusion-aware offloading for real-world edge systems.

Keywords: Mobile Edge Computing (MEC), Internet of Things (IoT), Intrusion Detection System (IDS), Random Forest, LightGBM, XGBoost, DTOME, Security

Introduction

The rapid growth of the Internet of Things (IoT) has led to the deployment of large-scale, latency-sensitive applications that require real-time data processing and reliable communication. To address the limitations of cloud-centric architectures, Mobile Edge Computing (MEC) has emerged as a promising paradigm by bringing computational resources closer to data sources. By enabling local processing and task offloading at the network edge, MEC significantly reduces latency, alleviates bandwidth congestion, and improves overall system performance. As a result, MEC has become a key enabling technology for critical IoT applications such as smart healthcare, autonomous vehicles, industrial automation, and online gaming.

Despite these advantages, the distributed and resource-constrained nature of MEC-enabled IoT environments introduces serious security challenges. Edge nodes and IoT devices are particularly vulnerable to network-based cyberattacks, including Distributed Denial of Service (DDoS), probing, spoofing, and privilege escalation attacks such as User-to-Root (U2R) and Remote-to-Local (R2L). These attacks can disrupt service availability, manipulate task execution, and significantly degrade the quality of service.

Fig. 1 illustrates how heterogeneous IoT devices connected through edge nodes continue to face volumetric and network-level threats across the access and distribution networks.

Conventional task offloading frameworks in MEC primarily focus on optimizing performance metrics such as latency, energy consumption, and resource utilization. However, most existing approaches overlook cybersecurity considerations and assume that incoming tasks are legitimate. This assumption is increasingly unrealistic in adversarial IoT environments. Furthermore, the limited computational capacity, memory, and battery life of edge devices make it challenging to deploy complex security mechanisms using traditional rule-based or signature-based intrusion detection techniques.

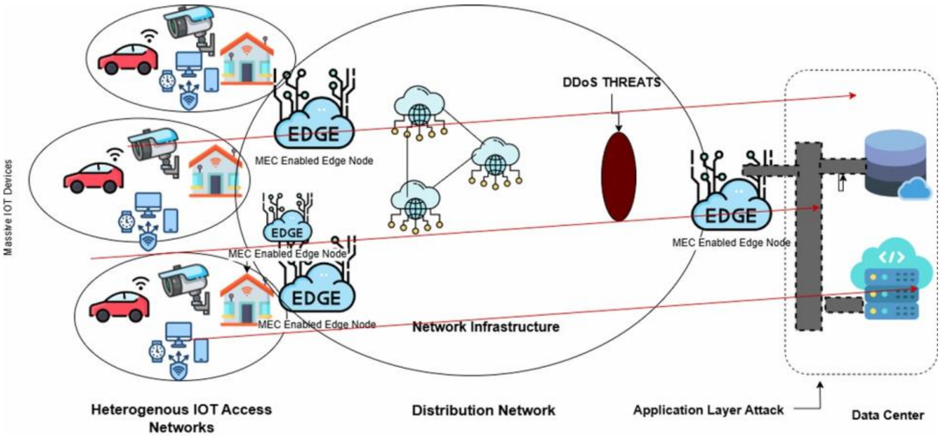

Fig. 1: Heterogenous IoT Access Networks and Threats in MEC Environment

Consequently, there is a critical need for intelligent and lightweight security solutions that can operate effectively at the network edge without compromising system efficiency.

In this paper, we propose a security-aware extension of the DTOME (Dynamic Task Offloading with Hybrid Energy Supply) framework, which minimizes system cost by jointly optimizing energy consumption and task delay under hybrid energy sources using Lyapunov optimization. The proposed approach integrates a machine learning–based Intrusion Detection System (IDS) at the edge server to proactively analyze incoming IoT traffic and classify requests as normal or malicious before executing task offloading decisions. This work focuses on securing the task admission and offloading pipeline in MEC systems rather than re-optimizing or experimentally re-evaluating the internal offloading algorithm itself.

By embedding intrusion detection into the offloading workflow, the framework enhances both security and operational reliability in MEC-enabled IoT systems.

The effectiveness of the proposed framework is evaluated using widely adopted benchmark intrusion detection datasets, including NSL-KDD, CICIDS2017, and UNSW-NB15, covering multiple attack categories such as DDoS, spoofing, U2R, and R2L. The proposed approach is further compared with recent state-of-the-art IDS solutions designed for MEC environments to demonstrate its effectiveness and practical relevance against two recent studies: The "Edge-Based Hybrid Intrusion Detection Framework for Mobile Edge Computing" by Ahmad et al. (2022); and the ensemble-based IDS proposed by Aldaej et al. (2024).

Contributions

1. Integration of security and task offloading: A unified framework that combines machine learning–based intrusion detection with dynamic task offloading in MEC-enabled IoT environments
2. Security-aware offloading mechanism: A proactive IDS layer that filters malicious traffic at the edge before offloading decisions, improving system reliability under adversarial conditions
3. Comprehensive multi-dataset evaluation: An extensive experimental study using multiple benchmark datasets and performance metrics to enhance the reliability and generalizability of the results

Background and Related Work

Internet Security

Internet Security is all about strategies and technologies that protect networks, systems, and data from unauthorized access, data breaches and cyber-attacks. As long as services transition online, the need for robust security frameworks becomes more critical. The rise of Artificial Intelligence (AI) has significantly changed this domain by analyzing large volumes of data and detecting anomalies. AI-powered tools can predict, detect, and respond to cyberattacks in real-time, reducing response time and improving system security. Traditional signature-based systems might miss to detect some attacks so, behavioral analysis is used for this moral, as they are adaptive to new, unseen attacks as AI-powered IDS.

Challenges With Edge Computing

1. It’s a vital and crucial technology for processing data in real time in IOT environments. Although it plays a vital role, it poses significant challenges concerning security and privacy
2. As edge networks are distributed. It’s hard to use traditional security measures. Key challenges include device vulnerabilities, network security, and data privacy
3. By guaranteeing that data privacy is protected,

especially sensitive data, which is more challenging as data flows can be intercepted or manipulated

Recent Developments in Edge and Cloud Security

Edge AI in Security: Edge AI makes impacts efficiently in security systems for making decisions faster and using data locally at the edge instead of transmitting it to the cloud. This make it likely that data will be intercepted while it is being sent and speeds the time it takes to respond to threats. Adversarial AI: Which manipulates AI models for making incorrect decisions, which is a growing concern for the future of cybersecurity.

Interpretability: Ensuring that data made by AI model are interpretable, especially in high stakes environments where wrong decisions can have severe consequences.

AI Driven IDS in cloud and MEC: In their systematic literature review, Jada and Mayayise (2023) showed that AI significantly enhances cybersecurity by automating threat detection and response, although adversarial attacks remain a major challenge. Schmitt et al. (2021) demonstrated that machine-learning classifiers in digital ecosystems can outperform traditional techniques in real-time threat detection.

Focusing on IoT and edge settings, Wang et al. (2023) emphasized that trustworthy edge intelligence is essential to ensure security, privacy, and reliability as shown in Fig. 2.

While the aforementioned studies demonstrate the effectiveness of machine learning and deep learning techniques in detecting cyber threats in cloud and edge environments, most of them focus solely on detection accuracy without considering how intrusion detection outcomes influence resource management or task offloading decisions in MEC systems. In contrast, the proposed framework integrates intrusion detection directly into the task offloading pipeline, ensuring that security decisions actively govern whether tasks are admitted to the MEC optimization process as shown in Fig. 2.

In addition, Al-Garadi et al. (2020) highlighted that deep learning and federated learning can reduce latency and enable real-time anomaly detection in IoT environments. Privacy-preserving mechanisms for AI-assisted IoT security were further examined by Rupanetti and Kaabouch (2024) as shown in Fig. 5, showing how edge computing can help maintain data confidentiality. Finally, Jada (2022) reinforced these findings by noting that, although AI improves anomaly detection at the edge, integration with existing systems remains challenging.

Machine Learning for Cybersecurity in MEC and IoT

Bakro et al. (2024) proposed a hybrid cloud-based IDS that combines GOA-GA feature selection with a Random Forest classifier, achieving between 98 and 99% accuracy on datasets such as UNSW-NB15 and CIC-DDOS2019. In addition, Attou et al. (2023) reported 99.99% accuracy using Random Forest and feature engineering on the NSL-KDD and Bot-IoT datasets, confirming the strong capability of Random Forest for anomaly detection in cloud environments as shown in Fig. 3.

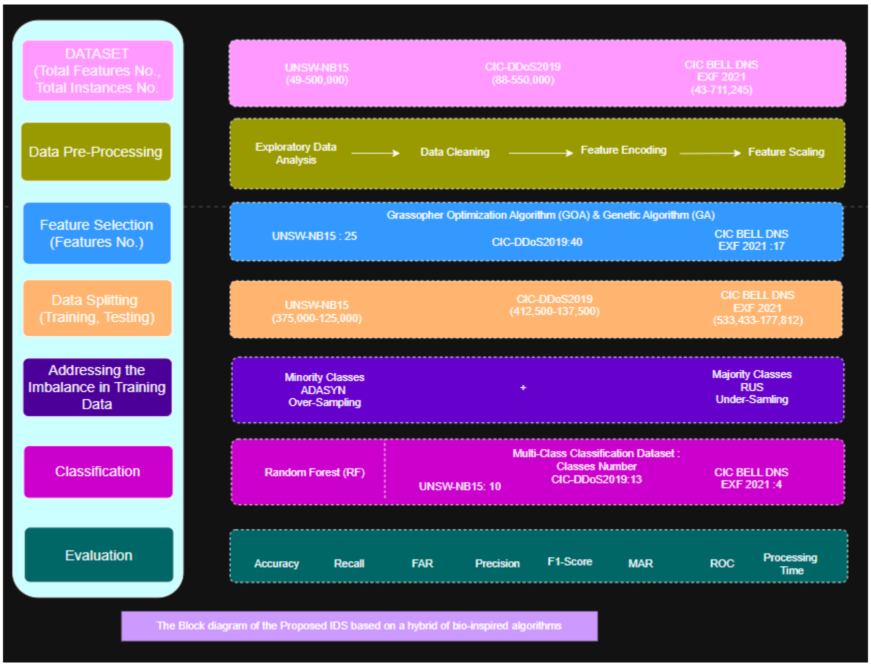

Fig. 2: AI-driven security in IoT and edge environments

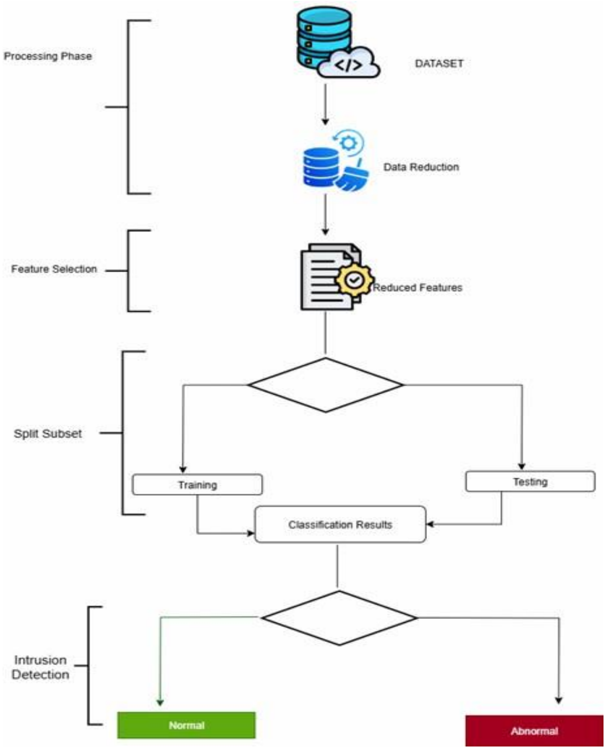

Fig. 3: AI-based anomaly detection and integration challenges

Although prior works such as Bakro et al. (2024); Attou et al. (2023) report high detection accuracy using Random Forest–based IDS models, their approaches are primarily designed for cloud or standalone detection scenarios. These studies do not address the interaction between intrusion detection and MEC-specific constraints such as latency sensitivity, edge resource limitations, or offloading dynamics. Unlike these approaches, the proposed framework is explicitly designed for MEC-enabled IoT environments, where intrusion detection operates as a gatekeeper for dynamic task offloading.

MEC Offloading and Performance-Driven Frameworks

Chen et al. (2023) proposed the DTOME algorithm, a dynamic task offloading framework for MEC with hybrid energy supply based on Lyapunov optimization. This framework addresses the stochastic nature of task arrivals and energy harvesting by constructing a "green energy grid" model that allows IoT devices to utilize harvested renewable energy (e.g., solar, wind) alongside the traditional grid. To minimize the long-term system cost while guaranteeing queue stability, the authors formulated a stochastic optimization problem and decomposed it into three deterministic subproblems to be solved in each time slot: Local task allocation, task offloading duration, and edge server task allocation. In a related study, Ahmad et al. (2022) developed an edge-based hybrid IDS for MEC environments that achieved 90.25% accuracy and a 1.1% false positive rate on the UNSW-NB15 dataset as shown in Fig. 4.

Aldaej et al. (2024) proposed an ensemble-based intrusion detection technique for IoT-edge platforms that employs a two-stage detection mechanism to balance efficiency and accuracy. In the first stage, a binary Extra Tree (E-Tree) classifier is utilized to rapidly distinguish between intrusive and non-intrusive traffic at the edge layer. Traffic flagged as malicious is then passed to the second stage, which utilizes a robust ensemble of E-Tree, Random Forest (RF), and Deep Neural Network (DNN) classifiers. The proposed framework was rigorously validated across multiple datasets, achieving 97.97% accuracy on the CICIDS2018 dataset and 98.7% accuracy on NSL-KDD, demonstrating high stability and a low false-positive rate as illustrated in Fig. 5.

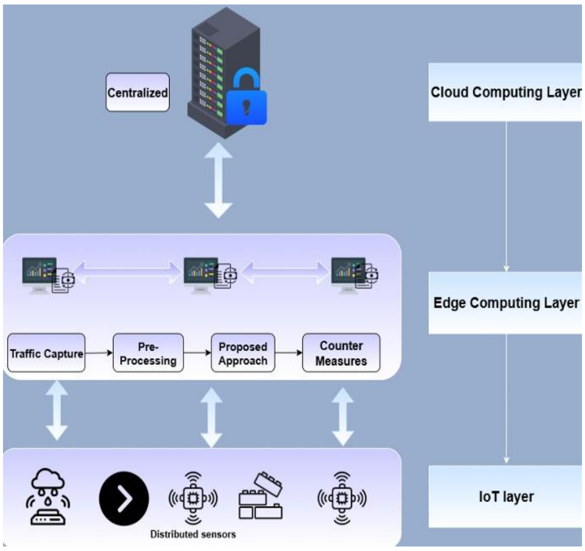

Fig. 4: EHIDF Framework Architecture

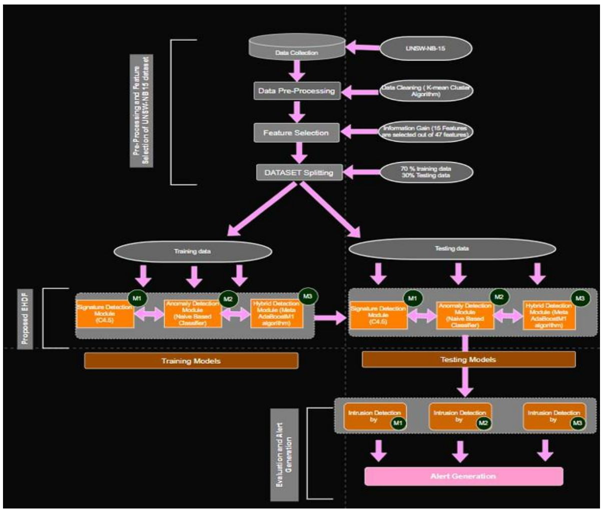
Pre-Processing and Feature Selection of UNSW-NB 15 dataset
Data Collection
UNSW-NB-15
Data Pre-Processing
Data Cleaning (K-mean Cluster Algorithm)
Feature Selection
Information Gain (15 Features are selected out of 47 features)
DATASET Splitting
70 % training data
30% Testing data
Training data
Testing data
Proposed EHDF
M1
M2
M3
Signature Detection Module (C4.5)
Anomaly Detection Module (Naive Based Classifier)
Hybrid Detection Module (Meta AdaBoostM1 algorithm)
Training Models
Testing Models
Evaluation and Alert Generation
Intrusion Detection by M1
Intrusion Detection by M2
Intrusion Detection by M3
Alert Generation

Fig. 5: Performance of the ensemble IDS for IoT-edge platforms

Existing MEC offloading frameworks, including DTOME and TOFFEE, primarily aim to optimize energy consumption, delay, and queue stability under stochastic task arrivals. However, these frameworks generally assume that all incoming tasks are legitimate and do not incorporate explicit security mechanisms. As a result, they remain vulnerable to attack-induced workload inflation and malicious task injection. Unlike these performance-driven approaches, the proposed framework introduces a security-aware extension in which machine learning–based intrusion detection precedes the offloading decision, thereby preventing malicious tasks from entering the optimization process.

Furthermore, Ghasemi et al. (2021) introduced the TOF-FEE (Task Offloading and Frequency Scaling for Energy Efficiency) framework, which addresses the challenges of dynamic task arrival and fluctuating wireless channel states in MEC. TOFFEE jointly optimizes task offloading and CPU frequency scaling to minimize energy consumption for mobile devices while strictly bounding the application queue length to ensure stability. By formulating the problem as a stochastic optimization and decoupling it into two deterministic subproblems Local Computation Allocation (LCA) and Offloaded Computation Allocation (OCA) the framework achieves close-to-optimal energy performance without requiring prior statistical information of the system as shown in Fig. 6.

Hybrid IDS in MEC in Kimmell et al. (2021) research assessed multiple machine learning algorithms for malware detection in cloud environments and found that Neural Networks (NN), specifically Convolutional Neural Networks (CNNs), were the most effective.

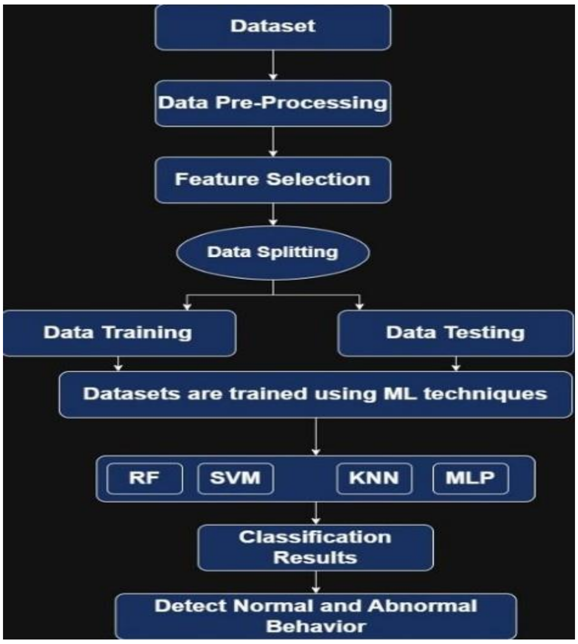

Fig. 6: Machine Learning Workflow for IDS

Their study utilized the DenseNet-121 architecture to analyze process-level performance metrics such as CPU usage and memory footprints, collected from virtual machines in a live cloud testbed. Unlike static analysis methods that rely on signatures, this online approach treats system feature sequences similarly to image data, allowing the CNN to capture complex, non-linear behavioral patterns of malware. In their comparative analysis against baseline models like Support Vector Classifiers (SVC) and Random Forests (RFC), the CNN model demonstrated superior performance, achieving an overall accuracy of over 99% and an F1-score of 91.5%, proving its robustness in distinguishing between benign and infected system states as shown in Fig. 7.

Literature Review

The rapid growth of IoT-enabled Mobile Edge Computing (MEC) environments has motivated extensive research on intrusion detection, edge intelligence, and task-offloading optimization. Existing studies have explored diverse machine learning models, feature-selection techniques, and hybrid MEC-aware offloading frameworks to enhance security and performance at the network edge. However, these works differ significantly in their objectives, datasets, methodologies, and achieved performance, making it essential to organize and compare them systematically.

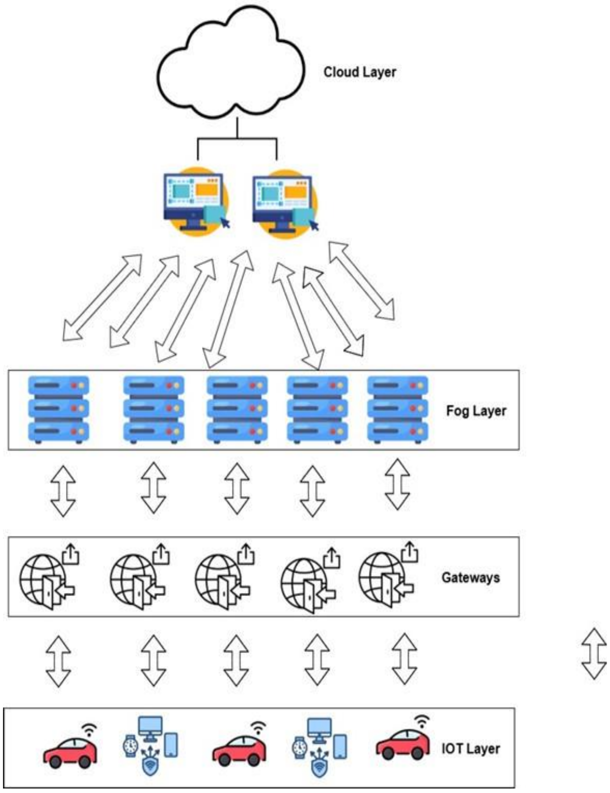

Fig. 7: Task Offloading and Caching Problems in MEC

In addition to the studies discussed below, several other works were reviewed and summarized in Table 1 (Ajala and Balogun, 2024; Hao et al., 2018; Tran and Pompili, 2019; Bebortta et al., 2023; Kushwah and Ranga, 2021; Alshammari and Aldribi, 2021). These studies provide further insights into energy-efficient task offloading, ML-based cloud intrusion detection, and optimization techniques in edge-cloud environments.

In summary, existing studies typically address either intrusion detection or task offloading optimization in isolation. Few works attempt to bridge the gap between security enforcement and offloading decision-making in MEC-enabled IoT environments. The proposed framework distinguishes itself by coupling a significant, edge-based intrusion detection mechanism with a dynamic task offloading strategy, enabling security-aware resource allocation without redesigning the underlying offloading algorithm.

Proposed Approach

The novelty of this work lies in the tight integration of a machine learning–based intrusion detection system with a dynamic MEC task offloading framework. Unlike existing studies that address intrusion detection or offloading optimization in isolation, this work introduces a security-aware offloading architecture in which intrusion detection actively governs task admission to the MEC optimization process. to ensure that only legitimate tasks are forwarded to the offloading module. In addition, the proposed framework is validated across multiple heterogeneous datasets, providing insights into robustness under diverse attack scenarios.

This research provides a robust approach for protecting and securing Mobile Edge Computing (MEC) environments within the area of the Internet of Things (IoT). The unprecedented increase in the volume of data and its rapid expansion at the network edge was massive due to the growth of IoT devices, starting from smartphones to smart sensors. This results in significant issues that must be taken in consideration like response latency, bandwidth use, and security. As a result for the increased latency and limited scalability, the traditional centralized cloud security mechanisms are no longer adequate specially when relying on external servers to perform sensitive tasks. In the context of MEC, which facilitates the decision-making process and makes task completion easier by offering low response time, it also reduces the reliance on centralized servers by allowing data processing and storage to occur closer to the data source (at the edge).

Table 1: Literature Review

Research Title	Used Method	Used Method
AI Impact on Organizational Cybersecurity	Systematic Literature Review (SLR) of 73 papers (2018-2023).	AI automates protection and enhances threat intelligence. Adversarial assaults and data quality remain hazards.
Trustworthy Edge Intelligence for IoT	Analysis of present solutions to protect AI in IoT settings.	AI is necessary for Trustworthy Edge Intelligence. It improves IoT security by detecting anomalies and reducing latency.
AI in Securing IoT and Edge Computing	Review of AI learning to identify risks in real-time.	AI-driven solutions depend heavily on security, privacy, reliability, and Quality of Service (QoS).
AI for Cybersecurity in Edge and Fog Computing	Deep Learning and testing systems for threat detection.	Machine learning classifiers in digital ecosystems can outperform traditional techniques in real-time threat detection.
Cybersecurity in Edge Computing and IoT Environments	Hybrid Feature Selection (GOA-GA) and Random Forest (Bakro et al., 2024).	AI enhances cybersecurity by making it easier to detect anomalies, though scalability in large IoT environments remains a challenge.
Cloud-Based Intrusion Detection	Random Forest and Feature Engineering on NSL-KDD and Bot-IoT.	Achieved 98% to 99% accuracy. Identified security gaps in cloud computing and suggested ML-based mitigation strategies.
Dynamic Task Offloading for MEC	DTOME algorithm using Lyapunov optimization.	Uses sub-algorithms for allocation and duration. Stochastic optimization ensures cost reduction and queue stability.
Ensemble technique of intrusion detection	Two-stage ensemble: Extra Trees (E-Tree) + Random Forest/DNN.	Proposed approach is robust and scalable, detecting intrusions with lower delay and higher stability than state-of-the-art.
Edge-Based Hybrid IDS Framework	Hybrid strategy using C4.5, Naïve Bayes, and AdaBoostM1.	Effectively discovered threats with fewer false alarms, making it a viable choice for edge deployment.
Online Malware Detection in Cloud	SVC, Random Forest, CNN, KNN.	Neural networks (CNNs) were found to be most effective, outperforming traditional models in behavior-based detection.

However, because of the extra attack surfaces as it may be difficult to distinguish between normal or abnormal tasks if no efficient and untraditional security measures are applied. So, all of these constraints highlights that edge cybersecurity remains a critical challenge.

After taking these cybersecurity challenges, a proposed ML-based Intrusion Detection System (IDS) is integrated with the MEC framework. After addressing these cybersecurity challenges, a machine learning–based IDS is integrated with the MEC framework to detect and block malicious traffic in real time before task offloading decisions are executed.

Framework

The proposed framework in Fig. 8 illustrates a machine learning-based intrusion detection and offloading decision architecture tailored for IoT-enabled Mobile Edge Computing (MEC) environments. The workflow consists of several interconnected components that collaborate to ensure secure and efficient task processing at the network edge.

IoT Devices and Task Generation: IoT devices (including smart cameras, home sensors, cars, etc.) continuously generate variant data and computational jobs. These tasks may include image processing, video analysis, or sensor reporting, that usually needs real-time processing due to sensitivity to delay.

- Goal: generate tasks and data streams from variant IoT sources
- Challenge: These devices are vulnerable to cyberattacks including DDoS, Spoofing and other attacks due to inappropriate security mechanisms

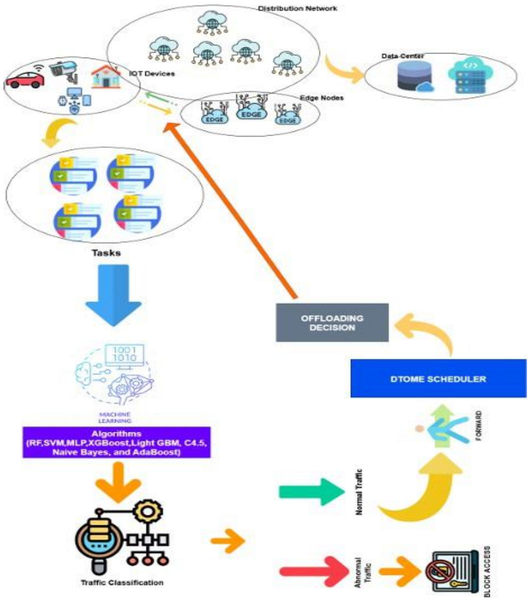

Fig. 8: Proposed Framework

Edge Nodes within the MEC Environment: The generated tasks are transmitted through a distribution network to nearby edge nodes, which serve as intermediate computing and storage layers closer to the data source than traditional cloud data centers. Edge nodes host the security and offloading logic.

- Advantages: Low latency, real-time responses, reduced network congestion

Task Offloading Analysis and ML-Classification: When the task arrives at the edge layer, the system examines and classifies the tasks whether normal or suspicious using a trained ML-based classifier which act as the ML-based intrusion detection system.

Traffic Classification: The classifier output is fed into a Traffic classification module, which distinguish traffic where it is:

(1) Normal Traffic: Authorized activities with no malicious behavior
(2) Abnormal Traffic: Tasks that exhibit suspicious or malicious traits. This is an important stage to determine system response strategy

DTOME Scheduler and Offloading Decisions

In this framework, DTOME is adopted as an existing baseline offloading mechanism to represent energy-aware task scheduling in MEC environments. The primary contribution of this work lies in introducing a security layer that filters malicious traffic before it enters the offloading decision process, rather than modifying or re-evaluating the internal optimization behavior of DTOME under adversarial conditions.

When the task is identified as Normal one, it is forwarded to the DTOME Scheduler (Dynamic Task Offloading for Mobile Edge Computing with hybrid energy supply). This scheduler is responsible for making offloading decisions, determining whether a task should be:

1. Executed locally on the current edge node
2. Transferred to the neighboring edge node due to load balancing
3. Offloaded to the cloud/ data center for processing (in case of the frequency scaling and response time exceed the local capabilities)

DTOME Objective: Enhance task allocation while preserving quality of service and resource efficiency, even under dynamic or partial threat scenarios.

Strengths:

1. Real-time Detection: ensures that the identification of threats is important on the cloud infrastructure
2. Edge-Centric Processing: reduce latency and impacts in fast response time

3. Integrated security and Scheduling: combines detection and resource allocation into a single framework
4. Scalable and Adaptive: the ability to respond to unstable network conditions and threats

Materials and Methods

Datasets

Three publicly available benchmark datasets were used to evaluate the proposed Intrusion Detection System (IDS):

1. NSL-KDD: an improved version of KDD'99 that removes redundancy. Also, it Contains normal traffic and four attack categories: DoS, Probe, R2L, U2R
2. CICIDS2017: a modern and realistic dataset generated by the Canadian Institute for Cybersecurity.it Includes DDoS, PortScan, Brute Force, Botnet, Web attacks, and more. Beside that it provides bidirectional flows with 80+ statistical features
3. UNSW-NB15: its generated using the IXIA Perfect-Storm testbed. It includes nine contemporary attacks such as Exploits, Fuzzers, Worms, DoS, Reconnaissance, Backdoor, Generic, Shellcode. Analysis. And it's considered more difficult due to overlapping distributions

Cross-dataset generalization experiments were not conducted in this study, as the primary objective was to evaluate intrusion detection performance within each dataset under consistent preprocessing and experimental conditions. Differences in feature spaces, traffic distributions, and labeling schemes across NSL-KDD, CICIDS2017, and UNSW-NB15 make direct cross-dataset evaluation non-trivial and beyond the scope of the current work.

Computational Environment

All experiments were executed on:

- CPU: 12th Gen Intel Core i7-12700H @2.30 GHz
- RAM: 16 GB
- Operating System: Windows 11 (64-bit)
- Programming Language: Python. Libraries: Scikit-learn (Random Forest, SVM, MLP, AdaBoost, C4.5), XGBoost, LightGBM, Imbalanced-learn (SMOTE), Pandas, NumPy for data handling

Data Preprocessing

To ensure consistency across datasets, a unified preprocessing pipeline was applied.

Data Cleaning: Removal of missing, corrupted, or duplicated samples. Replacement of infinite values (∞) and NaN with valid numeric values.

Feature Encoding: Label Encoding was used for categorical fields (e.g., protocol, service). One-hot encoding was applied where necessary for high-cardinality features.

A stratified train–test split of 70% for training and 30% for testing was adopted for all datasets. To address class imbalance, SMOTE was applied exclusively to the training data, while the test sets remained untouched. Feature scaling was performed using the StandardScaler technique. All experiments were implemented in Python using standard machine learning libraries such as Scikit-learn.

Feature Scaling: Because ML algorithms such as SVM and MLP are sensitive to scale, all continuous features were standardized using as shown in Equation 1:

$$x' = \frac{x-\mu}{\sigma} \tag{1}$$

The application of SMOTE in this study was motivated by the severe class imbalance observed in all evaluated intrusion detection datasets, particularly for minority attack classes such as U2R and R2L. To avoid biased learning toward majority classes, SMOTE was applied exclusively to the training set, while the test set was kept unchanged to preserve realistic evaluation conditions. Although this work does not include an explicit ablation study isolating the effect of SMOTE, prior studies have consistently shown its effectiveness in improving minority-class detection in IDS scenarios.

Train-Test Split: A stratified split was used (e.g., 70/30), ensuring all attack types are replicated proportionally in both sets.

Machine Learning Models

Eight machine learning algorithms were deployed to evaluate system's effectiveness and efficiency: Random Forest (RF), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Support Vector Machine (SVM), and Multi-Layer Perceptron (MLP).

Random Forest (RF): An ensemble of decision trees trained with bootstrap sampling. Random Forest resists noise and overfitting by the way, also, it has the ability to detect non-linear relations. Beside the mentioned advantages it has a strong advantage which is generalization. However, it's performance might not be as good when the attack patterns are very uneven.

XGBoost: Is a boosting method that fixes problems that earlier weak learners had. It handles complex decision boundaries and unbalanced datasets very well. Which is why its accuracy and F1-score keep going up.

LightGBM: Similar to XGBoost, but better at using memory and speed to make things run faster. It builds trees based on leaves instead of levels, which speeds up convergence. It did very well on NSL-KDD, getting the best results with very little computing power.

Support Vector Machine: A type of classifier that works well in high dimensional spaces and focuses on margins. It gets high accuracy and recall on smaller datasets, but it costs a-lot of computing power to work with large datasets like CICIDS 2017. Sampling was required to reduce training overhead.

MLP (Neural Network): A feed-forward neural network that can model non-linear relationships. It achieved excellent recall and F1 on CICIDS2017, showing strength in recognizing subtle, complex attack patterns. But it took longer to train and needed careful tuning of the parameters. C4.5 algorithm: is a decision tree classifier that uses entropy and the information gain ratio to break the data down into smaller pieces over and over again. This makes models that are easy to understand and good for tasks that need to be made clearer.

Naïve Bayes: is a type of classifier that uses Bayes' theorem and assumes that the features are not connected to each other. Even though it makes some assumptions that are too simple, this makes it fast and useful for datasets with a lot of dimensions.

AdaBoost (Adaptive Boosting): is a method for making a stronger classifier by changing the weights of instances that were wrongly classified over and over again. This is done by combining several weak learners, usually decision stumps. This makes the model more accurate and less likely to be biassed.

Integration With MEC Offloading (DTOME)

The IDS was integrated with the Dynamic Task Offloading with Hybrid Energy Supply (DTOME) framework. The integration follows these stages:

1. Incoming IoT tasks reach the edge node
2. A pre-processing security layer applies the trained ML classifier
3. If a task is normal, DTOME optimizes: Local execution, offloading duration, Edge server allocation using Lyapunov-based dynamic optimization
4. If a task is malicious, it is blocked immediately
5. This ensures security-aware, latency-optimized MEC operation

Evaluation Metrics

To ensure comprehensive evaluation, multiple metrics were used. Each focus on different aspects of Intrusion Detection Performance.

Accuracy: The ratio of properly categorized samples to all classifications. However, in cases of imbalance, accuracy by itself maybe deceptive as shown in Equation 2:

$$Accuracy=\frac{TP+TN}{TP+TN+FP\,FN} \tag{2}$$

Precision: Predicted attacks that were truly attacks. High precision means fewer false alarms as shown in Equation 3:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3}$$

Recall (Detection Rate): actual attacks that were correctly identified. High recall signifies resilience against overlocked threats as shown in Equation 4:

$$\text{Recall}=\frac{TP}{(TP+FN)} \tag{4}$$

F1-score: Balances precision and recall, essential in Intrusion Detection Systems since both false positives and undetected threats incur significant costs as shown in Equation 5.

$$F1 = 2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{5}$$

False Positive Rate (FPR)

Proportion of benign traffic wrongly flagged as attacks. Lower FPR is essential for MEC systems to avoid unnecessary blocking of legitimate services as shown in Equation 6:

$$FPR=\frac{FP}{FP+TN} \tag{6}$$

Latency: Measures per-sample prediction delay, which is crucial for real-time operation in MEC where decisions must be made within milliseconds as shown in Equation 7:

$$\text{Latency}=\frac{\text{PredictionTime}}{\text{No. of Samples}} \tag{7}$$

These metrics jointly assess correctness (accuracy, F1), safety (precision, FPR), and responsiveness (latency). Latency is crucial in MEC scenarios where real-time detection directly affects task offloading stability.

Results and Discussion

This section provides a comprehensive interpretation of the experimental results obtained from applying multiple machine learning classifiers on the NSL-KDD, CICIDS2017, and UNSW-NB15 datasets. The analysis focuses on identifying performance trends, comparing the proposed framework with existing studies, explaining discrepancies across datasets and algorithms, and discussing their implications for MEC-enabled IoT intrusion detection. The evaluation considers accuracy, precision, recall, Fl-score, false alarm rate (FAR), and latency all of which play a critical role in real-time edge security.

Overall Performance Trends Across Datasets

The results demonstrate that tree-based ensemble models particularly XGBoost, LightGBM, and Random Forest consistently outperform classical models such as SVM, Naïve Bayes, and C4.5 across all datasets.

CICIDS2017

- XGBoost: 99.92% accuracy, 99.97% recall, 0.10% FAR, latency ≈ 2.41×10^{-7} s/sample
- LightGBM: 99.90% accuracy, 99.97% recall, 0.11% FAR, latency ≈ 4.98×10^{-7} s/sample
- Random Forest: 99.85% accuracy, 99.81% recall

These values significantly exceed classical baselines reported in literature such as: DNN: 95.55% accuracy, RF baseline: 95.55%, ET baseline: 96.75%. The improvements of ~3-4% absolute accuracy and dramatically lower FAR (~0.1%) indicate that boosting + consistent preprocessing + SMOTE balancing provide almost perfect separability on CICIDS traffic.

NSL-KDD

- LightGBM: 99.63% accuracy, 99.71% precision
- 99.53% recall, 0.27% FAR
- XGBoost: 99.62% accuracy, 99.67% precision
- 99.53% recall, 0.31% FAR
- Random Forest: 99.58% accuracy, 99.73% precision, 99.40% recall, 0.25% FAR

These numbers far outperform prior studies, including the ensemble in Aldaej et al. (2024), which achieved 98.7% accuracy.

UNSW-NB15 (The Most Difficult Dataset)

- LightGBM: 90.95% accuracy, 97.90% precision: 88.60% recall, 4.05% FAR
- XGBoost:90.67% accuracy, 97.94% precision
- Random Forest: 90.80% accuracy, 98.05% precision, 88.23% recall, 3.74% FAR

Compared with the well-known EHIDF model proposed by Ahmad et al. (2022): EHIDF baseline: 90.25%. The proposed framework shows a clear improvement of 0.5-0.7%, even though UNSW contains overlapping distributions and harder attack families.

The very high accuracy values observed on NSL-KDD and CICIDS2017, particularly for gradient-boosted models, should be interpreted in the context of dataset characteristics. NSL-KDD is a cleaned and relatively well-structured dataset with limited feature overlap between normal and attack classes, which naturally facilitates near-perfect separability for ensemble classifiers. Therefore, high accuracy on NSL-KDD does not necessarily indicate overfitting, but rather reflects the maturity and simplicity of the dataset compared to modern traffic scenarios.

Behavior of Classical Machine Learning Models

Support Vector Machine (SVM): SVM achieved strong results on NSL-KDD (98.56% accuracy) but struggled heavily on CICIDS: CICIDS2017: 93.83% accuracy, 77.43% precision, 6.88% FAR, Latency: 3.77×10^{-4} s (very high). SVM is computationally expensive for large datasets. Highly sensitive to imbalance → high false positives. Latency too high for MEC real-time use.

MLP (Neural Network): Performed well: CICIDS: 98.67% accuracy, 96.69% F1, FAR: 1.50%.

Why Tree-Based Ensembles Consistently Outperform Other Models

The superior performance of XGBoost, LightGBM, and Random Forest is attributed to: 1. Ability to capture complex, non-linear patterns, especially in CICIDS and UNSW.

Sensitivity to preprocessing: SMOTE improves minority attack recall. Scaling eliminates distribution imbalance. Handling NaN/ improves numerical stability.

Boosted models (XGB/LGBM) reduce bias and overfitting simultaneously.

Feature selection (SelectKBest for UNSW) helps remove noisy attributes. This explains why the proposed framework achieves >99% on NSL-KDD and CI-CIDS, and competitive 91% on UNSW. But: Higher latency than ensembles. Requires careful tuning. Slower training time.

Naïve Bayes: For UNSW: Accuracy: 73.43%, Recall: 64.27%, FAR: 7.06%. Its weakness arises because: NB assumes feature independence (not true for UNSW/CICIDS). SMOTE generates synthetic samples that break NB statistical assumptions. This explains why NB in other papers sometimes performs better if they discretize features or skip scaling.

Decision Tree (C4.5): C4.5: 90.20% accuracy. Other studies: 86.04%. The improvement is due to: Better pruning, Balanced training, Feature selection, clean preprocessing.

Dataset-Specific Insights

Compared to recent MEC-based intrusion detection frameworks such as EHIDF and ensemble-based IDS approaches, the proposed framework demonstrates consistently higher detection accuracy and lower false alarm rates across all evaluated datasets, while maintaining suitability for edge deployment.

CICIDS2017: As shown in Table 2 it exhibits stronger feature separability for several attack categories in our setting, which contributes to higher ensemble performance: Ensembles reach ≈ 99.9%. Low FAR (<0.15%). Suitable for real-time MEC IDS.

NSL-KDD: as shown in Table 3, clean and balanced structure. Ensembles produce very high accuracy. XG-

Boost and LightGBM achieve >99.6% easily.

To mitigate this limitation, the proposed framework was also evaluated on more recent and challenging datasets, namely CICIDS2017 and UNSW-NB15, which exhibit higher feature dimensionality, stronger class imbalance, and overlapping attack distributions. The comparatively lower performance observed on UNSW-NB15 (≈91% accuracy) confirms that the proposed models do not trivially overfit, but instead face realistic detection challenges similar to real-world MEC environments.

Moreover, several design choices were intentionally adopted to reduce overfitting risk, including strict train–test separation, SMOTE application limited exclusively to the training set, feature scaling, and evaluation across three heterogeneous datasets. The consistency of performance trends across datasets further suggests that the reported results reflect genuine learning behavior rather than dataset memorization.

UNSW-NB15: It is the hardest dataset because of the attack classes overlap, minority classes (e.g., Shellcode,

Analysis) are very small and false alarms slightly higher as shown in Table 4. Gradient boosting still best option (~90.9%).

All framework results are shown in Table 5.

Key conclusions for MEC: XGBoost and LightGBM are the best for deployment: Low latency + high accuracy, RF is excellent for stability, SVM is unsuitable due to very high inference cost, NB is good only for coarse filtering, and MLP is useful but needs hardware acceleration.

Most Important Findings: The proposed framework consistently outperforms baselines by 3-5% on CICIDS and 1-2% on NSL-KDD. Significant improvement over EHIDF proposed by Ahmad et al. (2022) on UNSW (90.95% vs 90.25%). Gradient boosting models are the most robust across conditions. False alarm rates remain extremely low (<0.3%) for NSL/CICIDS. XGBoost & LightGBM are the best choices for MEC due to low latency. And finally classical models are either too slow (SVM) or too fragile (NB) for MEC deployment.

Since malicious tasks are blocked prior to offloading, the optimization process operates on clean traffic, thereby avoiding attack-induced workload inflation rather than reacting to it.

Table 2: Comparative Performance Analysis of the Proposed Framework and Previous Studies on the CICIDS2017 Dataset

Method CICIDS	Accuracy (%)	Balanced Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	FAR (%)	Delay / Latency (s)
DNN	95.55	95.54	93.53	–	–	–	24.15
RF	95.95	88.65	94.13	–	–	–	44.2
ET	96.75	98.25	98.65	–	–	–	55.23
KNN	94.65	96.26	94.15	–	–	–	45.99
Naive Bayes	93.14	94.56	95.14	–	–	–	94.26
Proposed	97.97	97.79	98.14	–	–	–	3.23
Random Forest (proposed framework)	99.85	–	99.43	99.81	99.62	0.14	0.000000982
XGBoost (proposed framework)	99.92	–	99.6	99.97	99.79	0.1	0.000000241
LightGBM (proposed framework)	99.9	–	99.55	99.97	99.76	0.11	0.000000498
SVM (proposed framework)	93.83	–	77.43	96.77	86.03	6.88	0.000139
MLP (proposed framework)	98.67	–	94.17	99.35	96.69	1.5	0.000001

Table 3: Comparative Performance Analysis of the Proposed Framework and Previous Studies on the NSL-KDD Dataset

Method NSL-KDD	Accuracy (%)	Balanced Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	FAR (%)	Delay / Latency (s)
DNN	92.35	93.56	94.46	–	–	–	0.36
RF	92.83	92.44	94.15	–	–	–	5.15
ET	92.25	92.75	93.35	–	–	–	4.23
KNN	95.65	94.46	94.35	–	–	–	1.19
Naive Bayes	95.57	95.26	97.14	–	–	–	0.56
Proposed	98.7	98.88	98.04	–	–	–	1.13
Random Forest (proposed framework)	99.58	–	99.73	99.4	99.57	0.25	0.00000272
XGBoost (proposed framework)	99.62	–	99.67	99.53	99.6	0.31	0.00000132
LightGBM (proposed framework)	99.63	–	99.71	99.53	99.62	0.27	0.00000217
SVM (proposed framework)	98.56	–	98.39	98.63	98.51	1.5	0.000377
MLP (proposed framework)	99.15	–	99.22	99.02	99.12	0.72	0.00000106

Table 4: Comparative Performance Analysis of the Proposed Framework and Previous Studies on the UNSW-NB15 Dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 / ADR (%)	FAR (%)	Train Time (s)	Latency (s/sample)
C4.5	86.04	–	–	71.25	8.4	–	–
Naive Bayes	86.94	–	–	72.49	2.1	–	–
EHIDF	90.25	–	–	74.35	1.1	–	–
Random Forest (proposed framework)	90.8	98.05	88.23	88.23	3.74	41.95	0.0000165
XGBoost (proposed framework)	90.67	97.94	88.15	88.15	3.96	8.12	0.00000555
LightGBM (proposed framework)	90.95	97.9	88.6	88.6	4.05	9.73	0.0000122
SVM (proposed framework)	86.2	98.34	81.1	81.1	2.92	391.01	0.00967
MLP (proposed framework)	89.29	98.36	85.7	85.7	3.04	238.27	0.00000595
Naive Bayes (proposed framework)	73.43	95.1	64.27	64.27	7.06	0.11	0.00000161
Decision Tree C4.5 (proposed framework)	90.2	97.12	88.22	88.22	5.57	5.52	0.000000358
AdaBoost (proposed framework)	90.79	98.26	88.02	88.02	3.32	399.25	0.0000766

Table 5: Overall Comparative Results of the Proposed Security Framework across NSL-KDD, CICIDS2017, and UNSW- NB15 Datasets

Dataset	Method	Accuracy (%)	Balanced Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	FAR (%)	Delay / Latency (s)
CICIDS	Random Forest (proposed framework)	99.85	–	99.43	99.81	99.62	0.14	0.000000982
CICIDS	XGBoost (proposed framework)	99.92	–	99.6	99.97	99.79	0.1	0.000000241
CICIDS	LightGBM (proposed framework)	99.9	–	99.55	99.97	99.76	0.11	0.000000498
CICIDS	SVM (proposed framework)	93.83	–	77.43	96.77	86.03	6.88	0.000139
CICIDS	MLP (proposed framework)	98.67	–	94.17	99.35	96.69	1.5	0.000001
NSL-KDD	Random Forest (proposed framework)	99.58	–	99.73	99.4	99.57	0.25	0.00000272
NSL-KDD	XGBoost (proposed framework)	99.62	–	99.67	99.53	99.6	0.31	0.00000132
NSL-KDD	LightGBM (proposed framework)	99.63	–	99.71	99.53	99.62	0.27	0.00000217
NSL-KDD	SVM (proposed framework)	98.56	–	98.39	98.63	98.51	1.5	0.000377
NSL-KDD	MLP (proposed framework)	99.15	–	99.22	99.02	99.12	0.72	0.00000106
UNSW-NB15	Random Forest (proposed framework)	90.8	98.05	88.23	88.23	3.74	41.95	0.0000165
UNSW-NB15	XGBoost (proposed framework)	90.67	97.94	88.15	88.15	3.96	8.12	0.00000555
UNSW-NB15	LightGBM (proposed framework)	90.95	97.9	88.6	88.6	4.05	9.73	0.0000122
UNSW-NB15	SVM (proposed framework)	86.2	98.34	81.1	81.1	2.92	391.01	0.00967
UNSW-NB15	MLP (proposed framework)	89.29	98.36	85.7	85.7	3.04	238.27	0.00000595
UNSW-NB15	Naive Bayes (proposed framework)	73.43	95.1	64.27	64.27	7.06	0.11	0.00000161
UNSW-NB15	Decision Tree C4.5 (proposed framework)	90.2	97.12	88.22	88.22	5.57	5.52	0.000000358
UNSW-NB15	AdaBoost (proposed framework)	90.79	98.26	88.02	88.02	3.32	399.25	0.0000766

Conclusion

This study presented a security-aware intrusion detection and task offloading framework for MEC-enabled IoT environments by integrating a machine learning–based IDS with the DTOME offloading strategy. Extensive experiments on three benchmark datasets, NSL-KDD, CICIDS2017, and UNSW-NB15, demonstrated that gradient-boosted ensemble models, particularly XGBoost and LightGBM, achieve strong detection performance with low false alarm rates while maintaining suitability for edge deployment. The results confirm that filtering malicious tasks prior to offloading can effectively protect MEC resource optimization processes from attack-induced workload inflation.

Nevertheless, the limitations of the proposed approach should be acknowledged. First, the experimental evaluation was conducted in a simulated MEC environment using offline benchmark datasets, which may not fully reflect the dynamic and adversarial nature of real-world edge deployments. Second, the DTOME offloading mechanism was adopted as a baseline and was not re-evaluated under active attack conditions in terms of energy consumption, latency variation, or queue stability. Finally, like most supervised learning–based IDS solutions, the proposed framework may be susceptible to adversarial machine learning attacks and concept drift in evolving traffic patterns.

Future Work

Future research will focus on extending this framework toward real-world MEC testbeds, enabling online evaluation of task offloading behavior under live attack scenarios. In particular, joint simulations will be conducted to quantify the impact of detected intrusions on energy consumption, latency, and queue stability within DTOME-driven MEC systems.

Additionally, future work will investigate adversarially robust learning techniques, such as adversarial training and concept drift adaptation, to improve IDS resilience against evolving and stealthy attacks. Lightweight online and incremental learning strategies will also be explored to reduce retraining overhead and support continuous deployment at the edge. Finally, integrating cross-dataset generalization analysis and federated learning mechanisms represents a promising direction to enhance scalability, privacy preservation, and robustness in large-scale IoT-enabled MEC environments.

Acknowledgment

All praise is due to ALLAH, the Most Gracious and the Merciful. The authors would like to express their sincere gratitude to their supervisors, Prof. Dr. Abdelfattah Hegazy, Assoc. Prof. Hesham Dahshan, and Dr. Ahmed Gaber, for their valuable guidance, constructive feedback, and continuous support throughout this research journey. The authors also extend their appreciation to the Arab Academy for Science, Technology and Maritime Transport (AASTMT) for providing the facilities and academic environment that enabled the successful completion of this work.

Funding Information

This research received no external funding. The work was conducted as part of the academic requirements of the Master’s program at the Arab Academy for Science, Technology & Maritime Transport (AASTMT), Egypt. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Authors’ Contributions

Rofaida Tawfik: Conceptualization, data preprocessing, model development, experimentation, analysis, and manuscript drafting.

Abdelfattah Hegazy and Hesham Dahshan: Supervision and academic oversight.

Ahmed Gaber Abuabdallah: Methodology design, Project supervision, experimental direction, structural review.

Ethics

This study does not involve human participants, animals, or sensitive personal data. All datasets used (NSL-KDD, CICIDS2017, and UNSW-NB15) are publicly available benchmark datasets, and all experiments were conducted in accordance with the ethical standards of research and data usage policies. No ethical approval was required for this research.

References

Ahmad, S., Chatterjee, K., & Satapathy, S. C. (2022). An edge-based hybrid intrusion detection framework for mobile edge computing. *Neural Computing and Applications*, *34*(2), 1–17. https://doi.org/10.1007/s40747-021-00498-4

Ajala, O. A., & Balogun, O. A. (2024). Leveraging AI/ML for anomaly detection, threat prediction, and automated response. *World Journal of Advanced Research and Reviews*, *21*(1), 2584–2598. https://doi.org/10.30574/wjarr.2024.21.1.0287

Aldaej, A., Ullah, I., Ahanger, T. A., & Atiquzzaman, M. (2024). Ensemble technique of intrusion detection for IoT-edge platform. *Scientific Reports*, *14*(1), 11703. https://doi.org/10.1038/s41598-024-62435-y

Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., & Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys & Tutorials*, *22*(3), 1646–1685. https://doi.org/10.1109/comst.2020.2988293

Alshammari, A., & Aldribi, A. (2021). Apply machine learning techniques to detect malicious network traffic in cloud computing. *Journal of Big Data*, *8*(1), 90. https://doi.org/10.1186/s40537-021-00475-1

Attou, H., Guezzaz, A., Benkirane, S., Azrour, M., & Farhaoui, Y. (2023). Cloud-Based Intrusion Detection Approach Using Machine Learning Techniques. *Big Data Mining and Analytics*, *6*(3), 311–320. https://doi.org/10.26599/bdma.2022.9020038

Bakro, M., Kumar, R. R., Husain, M., Ashraf, Z., Ali, A., Yaqoob, S. I., Ahmed, M. N., & Parveen, N. (2024). Building a Cloud-IDS by Hybrid Bio-Inspired Feature Selection Algorithms Along With Random Forest Model. *IEEE Access*, *12*, 8846–8874. https://doi.org/10.1109/access.2024.3353055

Bebortta, S., Tripathy, S. S., Modibbo, U. M., & Ali, I. (2023). An optimal fog-cloud offloading framework for big data optimization in heterogeneous IoT networks. *Decision Analytics Journal*, *8*, 100295. https://doi.org/10.1016/j.dajour.2023.100295

Ghasemi, A., Chen, Y., Zhang, N., Zhang, Y., Chen, X., Wu, W., & Shen, X. S. (2021). TOFFEE: Task Offloading and Frequency Scaling for Energy Efficiency of Mobile Devices in Mobile Edge Computing. *IEEE Transactions on Cloud Computing*, *9*(4), 1634–1644. https://doi.org/10.1109/tcc.2019.2923692

Chen, Z., Hu, J., Min, G., & Zomaya, A. Y. (2023). Dynamic task offloading for MEC with hybrid energy supply. *Tsinghua Science and Technology*, *28*(3), 421–432.

Hao, Y., Chen, M., Hu, L., Hossain, M. S., & Ghoneim, A. (2018). Energy Efficient Task Caching and Offloading for Mobile Edge Computing. *IEEE Access*, *6*, 11365–11373. https://doi.org/10.1109/access.2018.2805798

Jada, A. (2022). Cybersecurity in edge computing and IoT environments: A systematic literature review. *Electronics*, *11*(11), 1724.

Jada, A., & Mayayise, T. (2023). AI impact on organizational cybersecurity: A systematic literature review (2018-2023. *Digital Business*, *3*(1), 100052.

Kushwah, G. S., & Ranga, V. (2021). Optimized Extreme Learning Machine for Detecting Ddos Attacks in Cloud Computing. *Computers & Security*, *105*, 102260. https://doi.org/10.1016/j.cose.2021.102260

Kimmell, J. C., Abdelsalam, M., & Gupta, M. (2021). Analyzing Machine Learning Approaches for Online Malware Detection in Cloud. *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, 189–196. https://doi.org/10.1109/smartcomp52413.2021.00046

Rupanetti, D., & Kaabouch, N. (2024). Combining Edge Computing-Assisted Internet of Things Security with Artificial Intelligence: Applications, Challenges, and Opportunities. *Applied Sciences*, *14*(16), 7104. https://doi.org/10.3390/app14167104

Schmitt, C. (2021). *AI for cybersecurity in edge and fog computing.*

Tran, T. X., & Pompili, D. (2019). Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks. *IEEE Transactions on Vehicular Technology*, *68*(1), 856–868. https://doi.org/10.1109/tvt.2018.2881191

Wang, X., Liu, Y., Chen, Z., Li, K., & Rahman, M. (2023). Trustworthy edge intelligence for IoT. *IEEE Communications Surveys & Tutorials*, *27*(1), 1–45.