KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

Comprehensive Characterization and Taxonomy of Evolving Distributed Denial of Service (DDoS) Threats in Heterogeneous Internet of Things (IoT) Environments

Selvi. T [*1], Research Scholar, Department of Computer and Information Science, Annamalai University, Annamalainagar.
Email Id: mcaselvipandian@gmail.com

Jayaganesh. J [#2], Assistant Professor, Department of Computer Science, Government Arts and Science College, Perumbakkam, Chennai.
Email Id: everjays@gmail.com Orcid: 0000-0001-8724-9818

Abstract:
The evolving Distributed Denial of Service (DDoS) landscape in heterogeneous Internet of Things (IoT) ecosystems presents a dynamic attack surface due to the diversity of devices, lightweight communication protocols, and inconsistent security settings. This paper provides a structured approach to classification, detection, and mitigation of IoT-based DDoS attacks by integrating a multidimensional taxonomy with simulation-based datasets, feature engineering, and advanced learning models. Using NS-3 and OMNeT++ simulations, volumetric floods, protocol exploits, reflective amplification, and stealthy low-rate attacks were emulated across IoT architectural layers. Statistical and protocol-aware features such as payload size, entropy, burstiness, and inter-arrival time were extracted and analyzed. Principal Component Analysis (PCA) retained ~79% of the variance while reducing feature dimensionality, improving computational efficiency. On controlled datasets, Random Forest and Support Vector Machine classifiers achieved an accuracy, precision, recall, and F1-score of ~95%, with an AUC of 0.95, demonstrating strong separability of benign and malicious traffic. Experiments with federated learning across low-end devices, mid-range gateways, and mixed-capability nodes showed accuracy improvements from 50% to ~67%, with communication overheads ranging from ~20 MB to ~60 MB depending on device profile. Attack vectors were mapped to IoT architectural layers to support targeted defense strategies such as cross-layer monitoring, SDN-based filtering, and blockchain-backed authentication. These results validate a scalable and adaptive framework for real-time IoT DDoS detection that combines traffic profiling, dimensionality reduction, efficient classification, and federated training. The findings underscore the need for protocol-aware, layer-specific defense mechanisms to counter the growing sophistication of adversarial strategies in IoT systems.

Keywords: IoT Security, DDoS Detection, Federated Learning, Feature Engineering, Machine Learning Classification, Cross-Layer Defense Strategies

Introduction:

The evolution of DDoS attacks in IoT highlights a significant shift in cyber threat dynamics. Initially targeting traditional IT infrastructures, DDoS attacks now exploit vulnerable IoT devices, often lacking computational power and security [1]. Massive botnets like Mirai and Mozi demonstrate the ease of hijacking unsecured IoT endpoints to launch large-scale attacks, employing sophisticated, multi-vector strategies that utilize encrypted channels and stealth techniques. The rise of DDoS-as-a-Service allows less experienced attackers to execute

significant assaults with minimal effort. This necessitates systematic classification and taxonomy of threats to effectively analyze and combat evolving DDoS tactics [2].

The evolution of DDoS attacks in IoT has drastically changed cybersecurity dynamics. Initially targeting traditional IT infrastructures, these attacks now exploit numerous IoT devices, characterized by limited resources and weak security [3]. New botnets like Mirai and Mozi exemplify how unsecured IoT endpoints can be leveraged for large-scale attacks. Modern DDoS operations are sophisticated, employing multi-vector strategies, encrypted communications, and stealth. The rise of DDoS-as-a-Service has further enabled unskilled attackers to launch large-scale assaults. Consequently, there is a need for systematic classification and taxonomy of these threats to effectively study and combat them [4].

The perception layer comprises devices and sensors with limited computational resources, making them vulnerable to unauthorized access, tampering, and firmware exploits. The network layer transmits data across various protocols like MQTT and Zigbee, exposing weaknesses that attackers can exploit [5]. The application layer, which involves software platforms and cloud services, faces risks related to authentication and data privacy. The lack of standardized security in IoT ecosystems broadens the attack surface, with common vulnerabilities including weak credentials and outdated firmware. Understanding attack techniques is crucial for developing effective detection methods against IoT-based DDoS threats [6].

UDP and TCP floods aim to saturate network bandwidth, while amplification attacks exploit protocol misconfigurations (e.g., DNS, NTP, Memcached) to generate excessive traffic. Reflective attacks redirect legitimate server responses to victims, escalating the assault [7]. Protocol-based exploits target IoT standards like MQTT, CoAP, Zigbee, and Modbus, allowing attackers to disrupt essential services with malformed requests. Attack tools like Mirai, BASHLITE, and Hajime automate device compromise and botnet creation, employing scanning and brute-forcing of default passwords. Recent trends include stealthy, low-rate attacks that mimic normal traffic and distribute across multiple sources, challenging conventional detection methods [8].

DDoS attack detection in IoT has evolved to address inter-device diversity, limited resources, and changing attack patterns. Traditional signature-based techniques detect known threats but struggle with new or hidden attacks. Anomaly-based methods using statistical analysis and machine learning can identify deviations in traffic patterns [9]. Deep learning models like CNNs, RNNs, and LSTMs effectively model complex traffic features. Federated learning enables collaborative discovery among IoT nodes while preserving data privacy. Software-defined networking (SDN) and network function virtualization (NFV) enhance detection through centralized and dynamic traffic monitoring, forming a crucial defense against DDoS attacks in IoT environments [10].

Defense strategies against DDoS attacks in IoT have evolved to address increasing attack complexity. Reactive measures like rate limiting, IP blacklisting, and traffic filtering provide quick, albeit limited, protection for smaller attacks. Proactive techniques utilize anomaly detection and automated responses to prevent significant damage [11]. The integration of Software-Defined Networking (SDN) and Network Function Virtualization (NFV) enables real-time control, while fog and edge computing enhance local defenses to reduce latency.

Collaborative architectures, including blockchain and federated learning, improve trust and coordination among diverse devices [12].

Quality data and benchmarking tools are crucial for developing methods to detect and mitigate DDoS threats in IoT environments. Existing datasets like Bot-IoT, TON IoT, and CICDDoS2019 provide useful traffic samples but often lack heterogeneity and realism. Synthetic datasets and simulated environments have addressed some gaps, but challenges remain in capturing diverse device behaviors and protocol interactions. Real-time, scalable detection systems face constraints from IoT device resources and high data volumes [13]. Practical deployment in edge and fog computing offers benefits like localized processing and low latency, but also introduces interoperability, energy consumption, and dynamic network topology issues. Effective algorithms that are accurate without excessive computational overhead are necessary for scalability. Addressing current dataset deficiencies and enabling real-time, scalable implementations will guide future IoT DDoS defense strategies [14].

There also exists a lack of sufficient, labeled data that was representative of actual and dynamic IoT systems settings, and thereby an artificial and heterogeneous data source has to be generated and then labeled [15]. Full-scale lightweight machines and federated learning models can play a critical role in allowing detection to be carried out on-device in a privacy-preserving and communication-optimized fashion. Multi-protocol and cross-layer detection methods are becoming significant that combine the information at network, transport, and application levels to enhance their accuracies against complex attacks [16]. Also, a combination of new technologies like blockchain technology to allow sharing of information safely and quantum-resistant cryptography, which allows devices to be authenticated, contains some potential [17]. The importance of standardization in order to harmonize security procedures and interoperability in different IoT platforms was also important in harmonizing defense measures.

Research Gap:

Existing studies of DDoS threats in heterogeneous IoT environments encounter a number of issues. Unexpectedly, there was a lack of holistic, realistic data that has a good representation of various IoT devices and protocols, which does not allow proper training and testing of models. Current taxonomies of attacks normally do not consider the additional threats surrounding the IoT, like the device heterogeneity and the multi-protocol weaknesses. The accuracy achieved by detection methods fails to maintain the balance between the required resources of IoT devices and real-time implementation and scalability. The defense mechanisms do not provide adaptive multi-layered protection embracing the new technologies such as blockchain. The absence of standardized security practices also hinders the interoperability between different types of IoT and protection on multiple platforms.

Research Methodology:

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

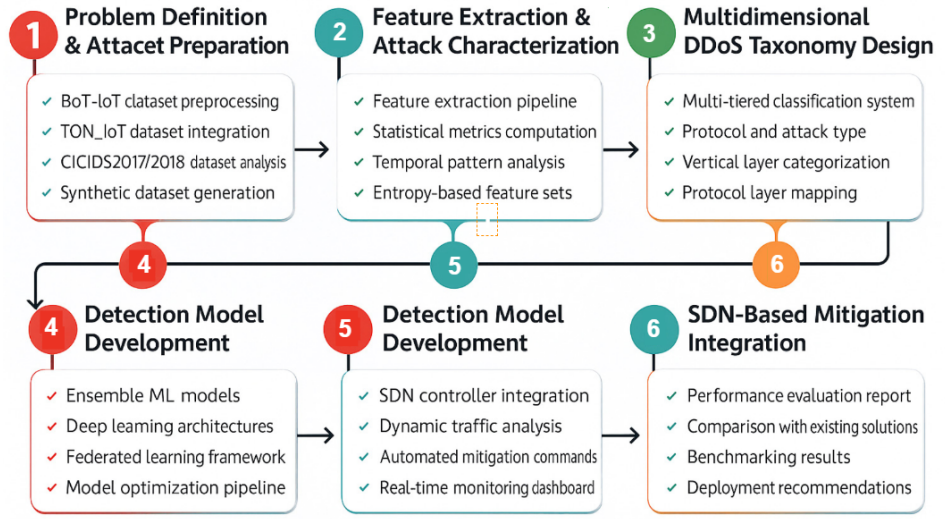


Figure 1. Research Methodology

Problem Definition and Dataset Preparation

Defending against DDoS attacks in IoT is challenging due to the diverse devices and protocols involved, which create multiple vulnerabilities and complicate traditional mitigation techniques. Many IoT nodes have limited resources, making heavy security measures impractical. Therefore, scalable, lightweight, and context-aware DDoS detection frameworks are needed for real-time operation in resource-constrained environments. Reliable detection systems require datasets that accurately represent IoT traffic, including normal and malicious patterns. While datasets like Bot-IoT, TON_IoT, and CICDDoS2019 provide attack samples, they often lack the diversity found in actual IoT environments, risking poor model performance in real-world conditions.

To address these gaps, synthetic datasets were generated using NS-3 and OMNeT++ simulators to emulate diverse IoT topologies and attack scenarios. These environments included various device profiles and communication protocols, incorporating realistic benign traffic and multiple DDoS attack classes for comprehensive threat coverage. This hybrid data approach captures the behavior of heterogeneous IoT networks while ensuring reproducibility.

All collected data underwent structured preprocessing using Python libraries such as Pandas and NumPy for quality and consistency. Raw logs were cleaned, normalized, and labeled with metadata including attack type and affected device class. Techniques like random undersampling and SMOTE oversampling addressed class imbalance, resulting in balanced datasets that supported supervised learning for DDoS detection in heterogeneous IoT settings.

IoT Network Simulation and Attack Emulation

A heterogeneous IoT environment was simulated using NS-3.37 and OMNeT++ 6.0 to evaluate the detection framework. The network had 100 to 500 devices, including 60% low-power sensors, 25% smart devices, and 15% gateways, with links operating at 2–10 Mbps and cloud connections at 100 Mbps to 1 Gbps. Access links used Drop Tail queues, while backbone routers applied RED queues; wireless segments followed a loss model, and wired links had constant-speed propagation. Malicious nodes made up 5 to 50% of devices with random selection. Attack traffic included UDP and TCP floods, various protocol exploits, reflective amplification, and low-rate floods. Each simulation lasted 300 seconds and was repeated three times for variability.

Background traffic was designed to emulate real IoT workloads, including periodic MQTT publish/subscribe exchanges at 1–5 second intervals, CoAP GET and POST requests, and Zigbee control frames. Additional application-level load was generated by HTTP requests and video streams at approximately 2 Mbps using H.264 encoding. Noise was modeled using Poisson-distributed arrival processes to replicate stochastic IoT events. All packet traces were captured using tcpdump and Wireshark at gateway and backbone nodes, providing labeled datasets for subsequent feature extraction and analysis.

Simulations were executed on Ubuntu 22.04 running on an Intel i7 processor with 16 GB of RAM. Random seeds were set to 42 for baseline reproducibility and varied across runs to capture statistical variance. Scenario files included NS-3 scripts written in C++ and Python for topology definition and attack orchestration, together with OMNeT++ configuration modules for multi-protocol traffic generation and attack placement. The combination of detailed parameterization, controlled randomization, and repeatable execution ensured that the simulated environment could be reproduced and extended for further benchmarking.

A compact set of simulation defaults generated the reported results, with runs of 100–500 nodes (≈60% low-power sensors, 25% smart devices, 15% gateways), sensor→gateway links of 2–10 Mbps, gateway→cloud links of 100 Mbps–1 Gbps, and a 10 Gbps backbone. Access links used DropTail queues (50–100 packets) and backbone routers used RED (max 1000 packets). Simulations lasted 300 s (also tested at 30 s and 120 s) with attacker prevalence varied at 5% (Low), 25% (Medium), and 50% (High): attacker counts were 5/25/50 with rates of $1\times10^3/1\times10^4/1\times10^5$ pps (low-rate attacks at 1–5 pps). Reflective amplification defaults were DNS = 40×, NTP = 300×, and Memcached = 20,000× (only in closed testbed scenarios). Background traffic included MQTT, CoAP, Zigbee, HTTP, and H.264 video (~2 Mbps); Poisson arrivals were applied for stochastic noise. Random seeds were set to 42 (baseline) with repeats using 101, 202, and 303; experiments were conducted on Ubuntu 22.04 (Intel i7, 16 GB RAM).

Feature Extraction and Attack Characterization

The proposed detection framework links raw network traffic to machine learning classification through feature extraction. Given the heterogeneity of IoT networks, careful feature engineering is vital. Traffic data from simulated and public datasets was analyzed with Wireshark and tcpdump for packet-level insights on protocol headers and timing patterns. A 42-dimensional feature vector was developed to capture volumetric, temporal, and protocol-level attributes of traffic, computed on a per-flow, per-time-window basis with 1-second aggregation windows. Features included flow descriptors, transport-level indicators, entropy measures, application-layer indicators, one-hot encoded protocol types, and normalized continuous features.

To reduce computational overhead, feature selection and dimensionality reduction were applied post-extraction. Initially, correlation analysis removed redundant features. Then, Principal Component Analysis (PCA) was conducted on the merged training dataset before federated partitioning, ensuring consistent local traffic projections across clients. Local PCA was avoided to maintain feature space compatibility and federated convergence. PCA retained 79% of total variance, reducing dimensions to the top 30 components, thus lowering computational costs and enhancing model convergence without loss of discriminative power. The 79% variance threshold was chosen after sensitivity analysis indicated performance degradation with fewer than 20 components.

Extracted features created behavioral signatures for various DDoS attack types, such as volumetric UDP floods with high packet rates, protocol-based attacks with abnormal flag combinations, reflective amplification attacks with high source IP entropy, and low-rate floods appearing as persistent low-volume anomalies. These signatures were mapped to a multidimensional taxonomy linking attack types to their vectors, target protocols, device classes, and IoT layers, ensuring detection models were based on real IoT traffic behavior.

Design of Multidimensional DDoS Taxonomy

The increasing complexity of DDoS threats in IoT networks requires a structured framework for analyzing diverse attack patterns. Traditional taxonomies often focus solely on attack vectors or targeted services, which is inadequate for heterogeneous IoT environments where attacks may exploit various architectural weaknesses. To fill this gap, a multidimensional taxonomy classifies DDoS attacks by primary vector, target device class, communication protocol, affected IoT layer, and evasion techniques. This approach provides a thorough understanding of attack behaviors essential for detection and mitigation. The taxonomy's first dimension includes attack classes like volumetric floods and stealthy threats, while the second links attacks to target device categories like smart cameras. The third dimension connects attacks to communication protocols such as MQTT and TCP/IP, enabling a comprehensive view of attack propagation across the IoT ecosystem.

A fourth dimension maps the attacks to the specific layers of the IoT architecture they disrupt—perception, network, or application—offering a clear picture of their propagation path and system-level impact. The fifth dimension captures common evasion techniques employed by attackers, including encrypted payloads, traffic obfuscation, and botnet-based distributed coordination, which often hinder detection efforts. The combination of these five dimensions results in a hierarchical structure that reflects the multifaceted nature of modern IoT DDoS attacks. By representing attacks in this manner, the taxonomy supports both fine-grained analysis and high-level pattern recognition, enabling security teams to develop more targeted and adaptive countermeasures.

To ensure its accuracy and practical utility, the taxonomy was validated using both real-world and simulated attack datasets. Traffic samples from Bot-IoT, CICDDoS2019, and TON_IoT were mapped onto the taxonomy to verify that all known behaviors could be classified without overlap or ambiguity. Domain experts were consulted to assess the taxonomy’s completeness and clarity, and refinements were made to resolve observed gaps or redundancies. This iterative validation process ensured that the taxonomy remained flexible and extensible, capable of incorporating emerging attack patterns and new IoT technologies. It thus serves as a foundational

reference that informs feature selection, guides detection model design, and facilitates consistent communication among researchers and industry stakeholders working toward enhanced IoT DDoS security.

Development of Detection Models

The proposed framework emphasized developing robust detection models to protect heterogeneous IoT environments from evolving DDoS threats. It balanced detection accuracy and computational efficiency, integrating classical machine learning algorithms and deep learning approaches. Algorithms like Random Forest and Support Vector Machine, implemented via Scikit-learn, provided reliable, low-complexity classifiers for edge devices. To analyze complex traffic dynamics, deep learning architectures using TensorFlow and Keras were introduced, including convolutional neural networks (CNNs) for spatial patterns and long short-term memory (LSTM) networks for temporal dependencies. These models were optimized for detecting subtle anomalies that traditional methods might miss.

Recognizing the distributed and privacy-sensitive nature of IoT networks, the framework also incorporated federated learning to enable collaborative training without centralized data collection. Using TensorFlow Federated, a federated setup was deployed in which 10–50 simulated IoT clients trained local models on their private datasets, and only model weights were periodically aggregated using Federated averaging (FedAvg) on a central server. This approach reduced network overhead, preserved data privacy, and improved the adaptability of detection models to local traffic conditions. The federated scheme also allowed new clients to join training rounds dynamically, enhancing scalability and enabling continuous model updates as the threat landscape evolved.

Model development was complemented by rigorous optimization processes to ensure high detection performance. Hyperparameter tuning was performed through randomized search with five-fold cross-validation, optimizing parameters such as learning rates, batch sizes, and tree depths. Performance metrics included accuracy, precision, recall, F1-score, and false positive rate to evaluate the balance between sensitivity and reliability. The models were trained and validated on both public datasets like Bot-IoT and CICDDoS2019 and on the synthetic datasets generated from the simulation environment. This hybrid evaluation ensured that the models could generalize effectively across varied IoT traffic profiles, attack intensities, and network conditions, thereby demonstrating their robustness and real-world applicability.

Training Hyperparameters and Cross-Validation Strategy

To ensure reproducibility and to balance detection accuracy with computational efficiency, all models were trained under carefully controlled hyperparameters. Deep learning models, including the CNN and LSTM architectures, employed the Adam optimizer with an initial learning rate of 1×10^{-3}. This rate was decayed to 1×10^{-4} after 30 epochs and further reduced to 1×10^{-5} after 60 epochs to enable fine-grained convergence. Mini-batch sizes between 32 and 64 were selected to accommodate the memory limitations of edge devices. Training was capped at 100 epochs and incorporated an early stopping rule with a patience of 10 epochs, halting training when the validation F1-score failed to improve. L2 weight decay of 1×10^{-5} and dropout regularization with rates between 0.3 and 0.5 were applied to mitigate overfitting.

To address class imbalance in the datasets, deep models used a weighted categorical cross-entropy loss that increased the penalty on the minority attack class. Classical models, such as

Random Forest and Support Vector Machine, applied a balanced class-weighting strategy that automatically set class weights inversely proportional to their frequencies. Hyperparameters for Random Forest, including the number of trees and maximum depth, and for SVM, including the kernel coefficient and regularization parameter, were optimized through randomized search procedures.

Model evaluation and hyperparameter tuning followed a five-fold stratified cross-validation approach to ensure unbiased performance estimates. The dataset was partitioned into five equal and class-balanced folds, with each fold serving once as the validation set while the remaining four were used for training. Performance metrics were averaged across all folds to provide a robust indication of model generalization and to avoid overfitting to any single data split.

Feature preprocessing ensured privacy and global consistency. Raw traffic features were standardized using global mean and variance from a shared training corpus, with scaling parameters distributed to clients for identical local transformations. PCA was performed on the global dataset before federated partitioning, and the transformation matrix was shared to maintain consistency in component space. Only PCA coefficients were communicated to prevent leakage of sensitive data. Client heterogeneity was addressed through weighted federated averaging, scaling model updates by dataset size and stabilizing with adaptive learning rates and gradient clipping for reliable convergence.

Performance Evaluation and Benchmarking

Evaluating the performance of the detection framework was essential to confirm its effectiveness, efficiency, and scalability in IoT environments. The framework rigorously assessed the system's accuracy and speed in identifying DDoS attacks under varied conditions using core metrics like accuracy, precision, recall, and F1-score, with a focus on minimizing false positives to maintain administrator trust. Latency and throughput were measured to ensure quick responses without bottlenecking network performance. Additionally, IDS-centric metrics were used for deeper operational insights, reporting precision, recall, and F1-score for each dataset alongside confusion matrices. ROC curves and AUC quantified discrimination quality, and the False Positive Rate captured benign flows misclassified as attacks. For streaming experiments, detection delay was measured from the first malicious packet arrival to when the model's inference exceeded a 0.95 confidence threshold.

Benchmarking against state-of-the-art detection methods evaluated the proposed framework's effectiveness using public datasets (Bot-IoT, CICDDoS2019, TON_IoT) and synthetic traffic from NS-3 and OMNeT++. This combination provided comprehensive coverage of traffic behaviors and attack strategies. To prevent overfitting, hold-out test sets and five-fold cross-validation were used, with results averaged over five runs. Metrics were reported as mean ± standard deviation, and statistical significance was assessed using paired t-tests or Wilcoxon tests.

In the federated learning context, client performance metrics were analyzed for fairness and consistency, focusing on accuracy, precision, and recall. Variance across clients was evaluated to ensure no systematic degradation. Scalability tests analyzed detection performance as network size and traffic volume increased, with simulations involving up to 500 IoT nodes and traffic loads from 10 Mbps to 1 Gbps. The federated setup was tested for varying client participation

impacts on communication overhead, convergence, and detection accuracy. Cross-domain tests assessed generalization, while robustness was evaluated under noise, label shifts, and model performance with compressed updates for bandwidth-constrained environments.

To protect privacy during distributed training, secure aggregation was enabled and its impact on performance was measured. Per-round latency overhead introduced by encryption and key-exchange operations was recorded and compared with the baseline without secure aggregation. Across all configurations, the additional delay remained modest—typically 8–12% for small networks and up to 15% in the largest setting—while convergence curves and final detection accuracy showed no significant degradation, confirming that secure aggregation preserves both privacy and learning efficiency.

The framework underwent stress-testing against adversarial attacks to assess its robustness. Four attack types were analyzed: data poisoning, backdoor insertion, label-flipping, and evasion at inference. In poisoning and label-flip cases, 5–20% of clients were selected to introduce corrupted samples, while backdoor attackers inserted trigger patterns in 5% of training data for misclassification. Evasion attacks altered feature vectors at test time through a projected gradient descent method with $L\infty$ bound of 0.02. Attack success rates (ASR) exceeded 80% for unmitigated backdoor attacks with less than 5% reduction in clean accuracy. Defenses applied included gradient clipping, anomaly detection, and robust aggregation methods like Krum and Trimmed Mean. Clipping with Trimmed Mean reduced backdoor ASR to below 10% while maintaining benign accuracy within 1% of the baseline, demonstrating the framework's enhanced resilience against adversarial behavior.

A comprehensive ablation study evaluated key algorithmic components' contributions by training system variants without meta-learning, using standard FedAvg, FedProx, FedMAML, and alternative aggregation strategies, while monitoring accuracy, convergence speed, and communication costs. Resource utilization was tracked to assess deployment on edge nodes, including CPU usage, memory, and network overhead. Integration with a Software-Defined Networking (SDN) controller evaluated real-time response capabilities against attacks. The analysis of detection accuracy, latency, computational cost, robustness, privacy, and fairness validated the framework's technical soundness and its readiness for deployment in heterogeneous IoT environments facing DDoS threats.

Results and Discussion:

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

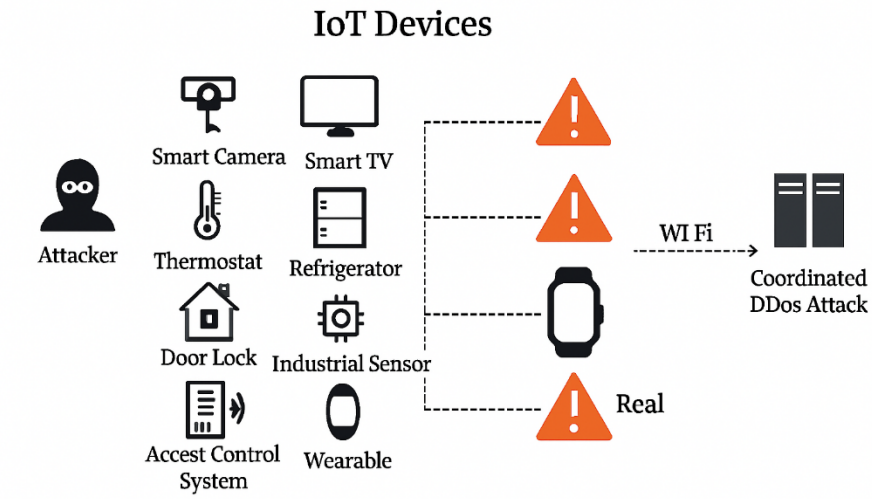


Figure 2: Botnet-Based Exploitation of Heterogeneous IoT Devices in Coordinated DDoS Attacks

In Figure 2, a common scenario of an IoT-based DDOS attack has been demonstrated, and how daily smart devices can be recruited to perform malicious botnet activities. The leftmost part presents the numerous IoT endpoints, such as smart cameras, smart TVs, thermostats, refrigerators, industrial sensors, door locks, wearables, and access control systems, which are often insecurely deployed and resource-constrained by nature [18]. These devices make up the perception layer of an IoT architecture, and the communication between them was usually via lightweight protocols. Attackers use these nodes to install malware or remotely control them in unknown ways due to having poor password policies, ineffective outdated firmware, and no encryption [19].

When infected, these Internet of Things are then used as members of a concerted botnet and are used to attack centralized services through multi-vector DDoS attacks. It was reflected in the figure as to how such infected devices are remotely coordinated on Wi-Fi orchestration to create synchronized flooding of traffic [20]. Such floods where be volumetric, reflective, or protocol-based. In a variety of cases, the attackers operate live devices with actual devices, and it was hard to separate the normal and malicious activities by using the traditional anomaly detection system. The devices were also having low-rate persistent traffic appearing as valid telemetry data intended to avoid intrusion detection based on thresholds [21].

This coordinated DDoS campaign exploits the homogeneity of network access (Wi-Fi) and heterogeneity of device functions, enabling attackers to mount stealthy, distributed, and resilient attacks. The ability to integrate industrial sensors and consumer electronics into a single botnet further increases the surface area of exploitation [22]. From a defense perspective, this visual underscore the need for cross-layer intrusion detection frameworks, federated anomaly detection models, and zero-trust access control mechanisms. Additionally, network segmentation, edge-based filtering, and blockchain-backed device authentication are emerging as essential countermeasures against such coordinated threats. This image captures the urgent reality of how easily unprotected IoT nodes can be transformed into agents of sophisticated, distributed attacks on critical infrastructure [23].

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

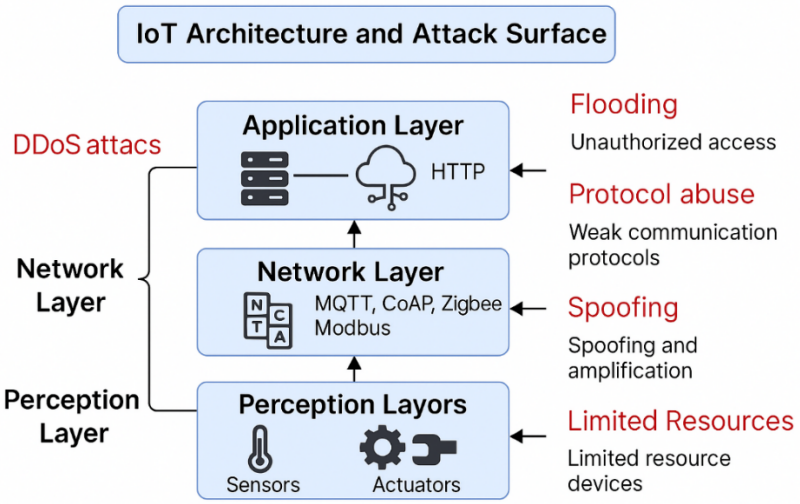


Figure 3: Layered IoT Architecture and Corresponding DDoS Attack Surfaces

This figure 3 gives an organized perspective of the IoT architecture layers and the relative DDoS attack areas that display how various threat vectors target diverse elements of the stack. The perception layer included sensors and actuators with the least chance of security since it had minimal computational power and limited energy. These devices are usually poorly authenticated with weak authentication methods and have no hardening at a firmware level and are thus an easy target. Examples of attacks at this level are resource exhaustion, where the limited processing and memory capacity are used up by the attackers to shut down the functionality of the devices or make them unusable by constant overloading of traffic [24].

The Network Layer has what it calls lightweight protocols, which include MQTT, CoAP, Zigbee, and Modbus, which are mostly deployed in the communication between a device and another device or a device and the cloud. Nevertheless, these protocols were created without strong security in mind. As indicated in the diagram, attackers can take advantage of weaknesses present at the protocol level to run spoofing, reflection, and amplification attacks, overwhelming the infrastructure network and consuming bandwidth. Security issues in the management of headers, sessions, and message integrity provide the attacker with information to generate malformed packets or fill the network with small, protocol-specific packets that cannot be easily filtered through standard intrusion prevention systems [25].

In the Application Layer, interfaces and applications were provided through HTTP-based systems and interfaces with remote-control capabilities. Increasingly, this layer was targeted via flooding attacks and attempts at unauthorized access that typically relied on botnets constructed out of devices on the perception layer that had been compromised. Targets of these attacks are to overload cloud services/databases/web applications with high volumes of requests coupled with malformed API calls. As illustrated, DDoS threats do not confine themselves and remain on one layer to cause outages but go on to traverse all three layers in a chain reaction. The diagram was a good illustration of this simultaneousness of the explosion of the attack surface and the driven chaos of detection, as multi-layered architecture places more demands on cloudifying the IoT, necessitating cross-layer, protocol-aware securitization mechanisms designed towards deployment heterogeneities of the IoT [26].

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING


Table 1 – IoT Layer-Wise Attack Taxonomy and Vulnerability Mapping

IoT Layer	Device/Protocol Examples	Attack Types	Exploited Vulnerabilities	Defense Recommendations
Perception	Sensors, actuators, cameras	Resource exhaustion, unauthorized access	Limited CPU/memory, default credentials, outdated firmware	Firmware hardening, access control, edge anomaly detection
Network	MQTT, CoAP, Zigbee, Modbus	Volumetric floods, spoofing, reflection, amplification	Weak protocol security, poor header validation	Protocol-aware intrusion detection, SDN-based filtering
Application	HTTP APIs, cloud services	API flooding, session hijacking	Weak authentication, rate-limit absence	Zero-trust API security, blockchain-based authentication
Cross-Layer	Multi-protocol IoT deployments	Multi-vector, stealth low-rate attacks	Combined protocol and resource weaknesses	Cross-layer monitoring, federated learning-based detection

Table 1 representing the taxonomy mapping reveals that DDoS based threats cut across all architecture layers of IoT and each one has unique vulnerabilities and attack vectors. Resource-constrained devices at the perception layer-like sensors and actuators-are prone to resource depletion and malicious communications because they have poorly implemented authentication and expired firmware. Network level attacks including volumetric, spoofing, reflection, and amplification attacks were possible on the network layer particularly under the lightweight protocols such as MQTT, CoAP, Zigbee, and Modbus where the validation of the header was minimal, and therefore an attacker can inject malformed packets. Application layer attacks utilize HTTP APIS and cloud services in order to overwhelm backend systems due to rate-limit gap and weak access controls.

The most worrying of these attacks are the cross layer multi-vector attacks which use a combination of perception-layer compromise and network and application layer floods to circumvent single-layer defenses. The mapping restates the need to have protocol-sensitive, cross-layer security schemes that combine anomaly detection in the edge, SDN-driven traffic modelling, and block-based device authentication. Notably, such multifaceted view fosters the creation of defensive solutions that are both specific, involving particular weaknesses, and dynamic, meaning they can shift with any new IoT protocols and deployment solutions [27].

$$H(X) = -\sum_{i=1}^{n} p\,(x_i)\log_2 p(x_i) \quad (1)$$

Within formula 1 the H (X) can measure randomness in the attributes of IoT traffic to include its source IP addresses, payload, or a value of the protocol header. Low entropy values

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

show regular and predictable device communication as found in benign operations of the IoT whereas high entropies are exhibited during DDoS attack traffic since addresses are spoofed, payload structures randomized, or multi-vector injected. Such a metric was a focal point of detection based on anomalies with maximum entropy levels being precursors of adversarial activity in heterogenous IoT settings particularly in those cases sustained over time [28].

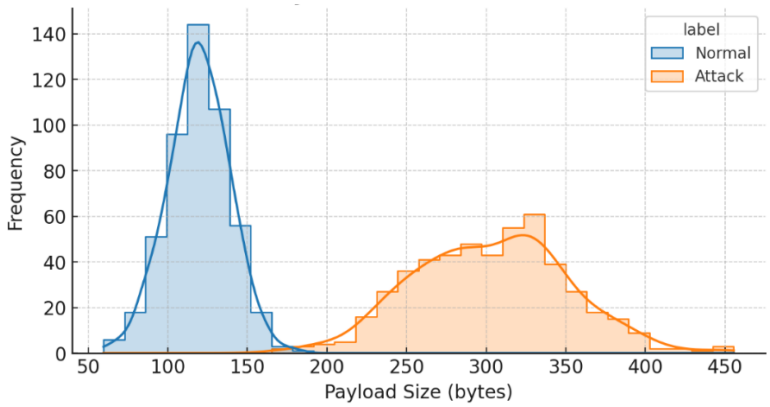


Figure 4: Comparative Payload Size Distribution in Normal vs Malicious IoT Traffic

The payload size distribution of normal versus attack traffic in the IoT environment was depicted in Figure 4 as a histogram with overlain KDE curves and provides a clear statistical view of the feature behavior applicable to detecting the DDoS attack. As shown, typical traffic payloads are highly concentrated between roughly 100 and 150 bytes, signifying normal communication patterns, cryptographic lightweight and regular, as it was characterized in telemetry, control signal, or status update between limited IoT nodes. Periodic sensor readings or actuator control commands form common patterns that have little overhead since they restrict them by the protocol [29].

Conversely, the attack traffic has a fairly wide and highly skewed distribution in the payload sizes, with the attack traffic occurring between 200 and 450 bytes and experiencing the highest peak payload at 320 bytes. This greater and more changeable payment was that of volumetric flooding, protocol exploitation, or spoofed packets injected, which had excessive or improperly structured data structures. Such an increase in size can be caused by the need to evade the rate-based anomaly detection by injecting more entropy or creating messages that appear to be valid protocol messages by exploiting improperly parsed header or field information [30].

The considerable bimodal difference between the two classes makes it viable to be used as a discriminative feature of anomaly detection models. Such a difference of distribution could be captured effectively through the ML classifiers, particularly the ones sensitive to the statistical variance and shape of a distribution. The marginal Gaussian association between distributions would imply that in real-world DDoS detection systems, payload size used alongside temporal or source entropy measurement metrics can produce significant true positive signal frequency with minimal false alarm levels produced under heterogeneous IoT deployment networks [31].

Table 2 – Feature-Based Statistical Differentiation Between Normal and Malicious IoT Traffic

Feature Metric	Normal Traffic Characteristics	Malicious Traffic Characteristics	Interpretation / Detection Relevance
Payload Size Range (bytes)	100–150; peak ~125	200–450; peak ~320	Attack traffic shows inflated payload sizes, often with malformed or excessive data, aiding in volumetric or protocol-specific flooding detection.
Entropy Range	1.1–2.8 (narrow IQR)	3.5–6.0 (wide IQR)	High entropy in attack flows indicates obfuscation/spoofing; a strong anomaly-based detection discriminator.
Burstiness (variance index)	1.0–1.5 (stable)	1.7–4.2 (variable, with outliers)	Elevated burstiness in malicious traffic reflects irregular, spiky transmission patterns typical of DDoS attacks.
Packet Inter-Arrival Time	Wide, periodic distribution	Concentrated near zero	Short inter-arrival times are a signature of flooding; lightweight metric suitable for real-time deployment.
CDF Payload Pattern	Sharp rise between 100–150 bytes	Gradual rise over 200–450 bytes	Consistent CDF in normal traffic allows adaptive thresholding; attack CDF indicates payload manipulation.
Temporal Entropy Variation	Stable < 20,000	Fluctuating 30,000–120,000	Temporal spikes in entropy align with active attack phases, enabling time-series-based anomaly detection.

A statistical comparison in Table 2 indicated that the payload size, entropy, burstiness, inter-arrival time, CDF properties, and temporal entropy variation all differ significantly and measurably between a normal IoT and malicious DDoS-induced traffic. Normal payloads are closely grouped in the 100-150 b range on account of the limited and digressed nature of telemetry messages. Conversely, payloads of an attack are even larger and more volatile, and they can reach ~320 B, which denotes injected/malformed data [32].

Entropy analysis shows that benign flows are low and narrow (1.1-2.8), typical for predictable device-to-device communications, while malicious flows are higher and random (3.5-6.0) due to spoofed, random field generation and multi-vector attacks. Burstiness measures indicate stable benign traffic and rapid variation in attack traffic from packet bursts. Payload analysis highlights that normal traffic has a wide, flat time distribution similar to IoT communication models, whereas malicious flows have many near-zero inter-arrival times, indicating an intent to overwhelm targets. Benign payload CDF curves increase exponentially with adaptive thresholds, while attack curves exhibit wide ranges and atypical distributions. Entropy trends confirm that malicious flows cause distortion by exceeding baseline values and

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

exhibiting high variance long-term. These features are useful for anomaly detection, especially in multi-feature classifiers, to reduce false positives and enhance early-stage detection [33].

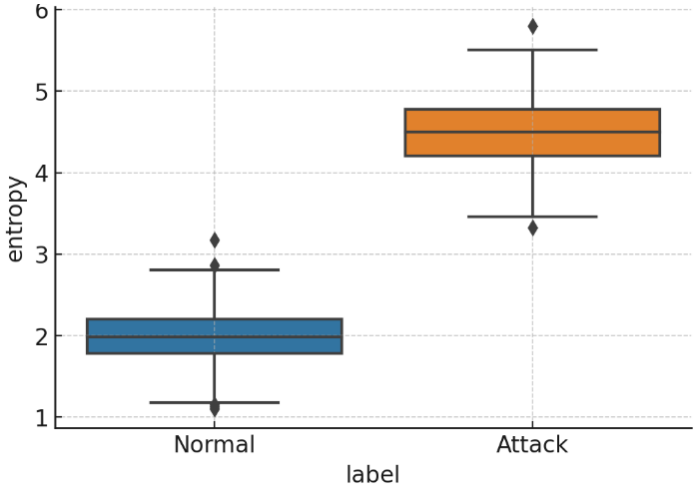


Figure 5: Entropy-Based Differentiation Between Normal and Malicious IoT Traffic

The boxplot used to compare the Entropy of different IoT traffic and malicious (attack) traffic presented in the figure below was a quantitative differentiation of the same which brings into perspective an important statistical characteristic employed in anomaly-based detection systems. In this context, entropy can be measured as the randomness or diffuse characteristics of the payloads of packets (or the attributes of the packet metadata. Low entropy would then correspond to the foreseeable and monotonous communication patterns, but high-entropy implies obfuscation, payload manipulation, or spoofing which are features of coordinated DDoS attacks [34].

Normal traffic has a range of lower levels of entropy (around 1.1 to 2.8), meaning that traffic was consistent and repetitive in its transmission patterns, in many cases caused by constant sensor values or control messages with regular frequencies. The small interquartile range and absence of outliers with a high variance indicate the stability and homogeneity of the expected prospect of a legitimate IoT operation. This easily predictable behavior allows more stringent statistical modeling and detection can be made more accurate by using dynamic thresholds in real-time monitoring systems [35].

Attack traffic, however, has much greater entropy (~3.5 to nearly 6) indicating either the use of payload obfuscation techniques, spoofs of identity header fields, or randomized message structure identifying it as an attack attempt. Such high-entropy patterns are especially prevalent in reflective DDoS attacks, attacks targeting amplification openings, and multi-vector flooding, where assailants intentionally attempt to increase the entropy in order to turn the packets into noise or otherwise unidentifiable gushers. Its distinguishing power between the normal and attack groups in this measure of entropy adds strength to its use as a discriminative feature used both in unsupervised anomaly diagnosis and in a deep learning model classification [36].

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

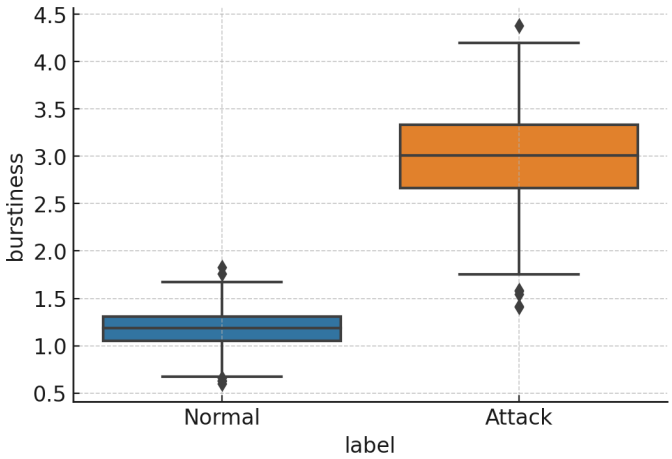


Figure 6: Temporal Burstiness Variation Between Legitimate and Attack Traffic in IoT Networks

Figure 6 offers essential information on schedules of packet flow patterns through normal and attack traffic in IoT environments. Burstiness was a statistical view of the variance of traffic flow over a period, or how lumpy or spiky the data sending is. During normal, non-stressed traffic, in particular, IoT traffic such as that led to sensors or actuator devices, messages would be regular and periodic; hence, the low values of theoretical burstiness. It can be seen in the plot that the interquartile range of normal traffic was constrained in the tight range of ~1.0-~1.5 with nearly absolute zero outliers, which presents consistent, low-latency transmissions [37].

In contrast, the values of burstiness at the attack traffic are much higher and more variable, thus being around 1.7 to more than 4.2. This was because of volumetric or surreptitious DDoS attacks, where bots produce fast, large packets of traffic in occasional spikes as opposed to continuous streams. This was particularly true in UDP/TCP flooding or reflective amplified DDoS attacks or low-rate slow-drip attacks whose payloads are deliberately triggered in bursts to stress buffers or circumvent time-based defenses. The larger interquartile range and the number of extreme outliers verify the unforeseeable transmission pace, which was characteristic of opposing packet generation techniques.

This time volatility leads to burstiness being quite a useful property for anomaly-based and time-series models, such as LSTMs and GRUs. Burstiness in collaboration with entropy and packet size as well as flow duration can detect the distinction between benign variations and malicious correlation in time. This illustration proves the usefulness of including traffic burst measurements in multi-feature detection schemes, especially in the detection of advanced or even distributed DDoS attackers in heterogeneous and constrained IoT environments.

$$B = \frac{\sigma_{\Delta t}}{\mu_{\Delta t}} \tag{2}$$

In formula 2, $\sigma_{\Delta t}$ was the standard deviation and $\mu_{\Delta t}$ was the mean of packet inter-arrival times in IoT network flows. The burstiness index B captures irregular transmission patterns that differ from the periodic telemetry of normal IoT devices. During DDoS events, traffic often arrives in high-volume bursts with short intervals, producing B values well above 1. This metric was particularly effective for identifying stealthy or low-rate flooding attacks, as it reflects changes in temporal packet distribution without relying on payload inspection [38].

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

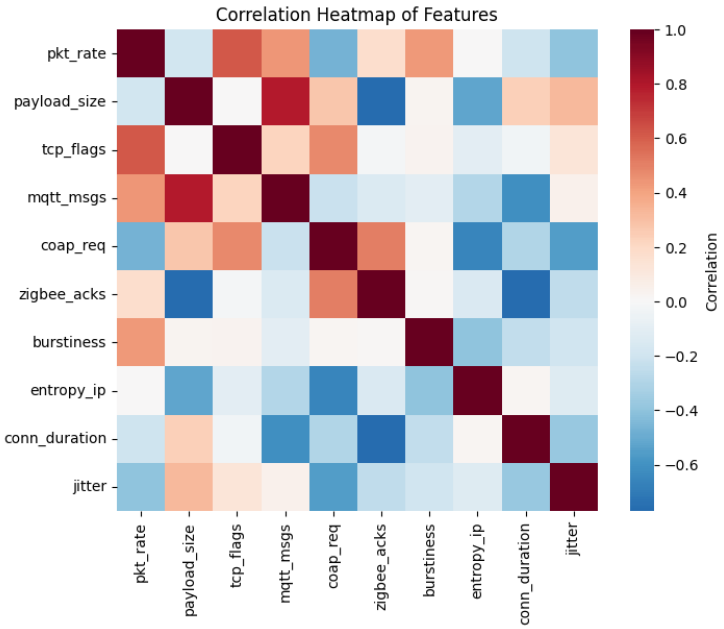


Figure 7: Correlation Map Showing Relationships Between Key Traffic Features in IoT DDoS Detection

The correlation heatmap provides a visual interpretation of the relation of the various traffic characteristics to one another in case of detecting DDoS attacks in IoT settings. The figures in each square depict the intensity and orientation of correlation between two characteristics. A value near 1 implies a close positive relation; the closer to 0, the more negative it is. When the value was near -1, it indicates a strong negative relationship, which implies that the increase of a feature was negatively correlated with the decrease of another one. The values close to 0 imply that there was no or a small relationship.

Based on the heatmap, packet rate, payload size, and burstiness are strongly correlated to one another. This implies that with attack conditions the number of packets increases, and so does the size of the packets as well as the irregularity in terms of burstiness traffic. Such behavior was characteristic of high volumes of DDoS attacks in which a large amount of data was delivered in brief periods of time to saturate the network.

Other characteristics such as entropy over IP addresses also have a high correlation with burstiness and payload size, which implies that the entropy of the system increases as the traffic was irregular and unpredictable (usually the result of spoofed or randomized sources). Correlations are weaker or isolated when based on more special features of IoT communication, such as MQTT messages, CoAP requests, and Zigbee acknowledgements. Such was significant in highlighting that IoT traffic was diverse, with the various devices and protocols responding differently to attacks and normal working conditions.

This heatmap was helpful to determine the features that you need to include in your detection model. Knowledge of the relationships between the different metrics to be detected, and those that are independent, enables a more precise, efficient and non-redundant detection system to be constructed. It can also be used to ensure the model was collecting both volumes based and behavior-based anomalies in traffic.

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \quad (3)$$

The correlation coefficient at shown in formula 3 r_{XY} measures linear relationships between key traffic features extracted from IoT flows, such as packet rate, payload size, and burstiness. In this study, strong positive correlations are observed in attack scenarios, where higher packet rates coincide with larger payloads and increased burstiness. This insight supports feature selection by highlighting which metrics carry overlapping information and which remain independent, thereby optimizing the detection model's complexity and computational footprint for deployment on resource-constrained IoT devices.

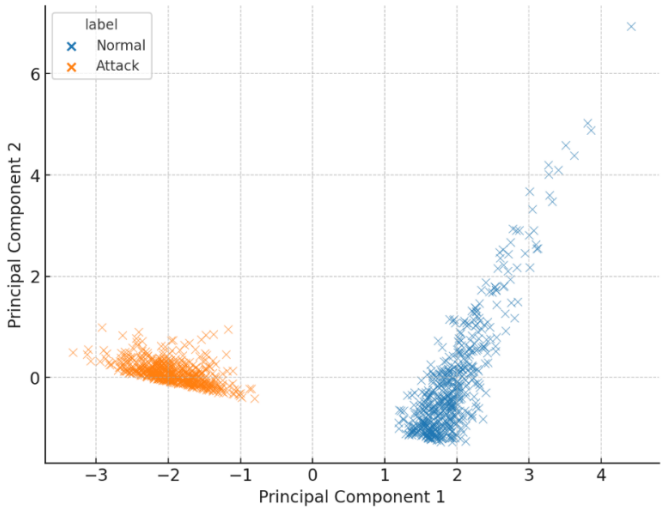


Figure 8: PCA-Based Visualization Showing Clear Separation Between Normal and Malicious IoT Traffic

Figure 8 shows that after a reduction in the feature space using Principal Component Analysis (PCA), the normal and the attack traffic in an IoT network become separated unequivocally. The dimensionality reduction approach PCA converts the data in a high-dimensional space into new principal components that describe the most notable patterns and variations, and, at the same time, the most amount of information in the first form of data as possible.

The two traffic classes are mapped within different clusters in the first two principal components in the scatter plot. When normal traffic was gathered in one group and attack traffic in another one, it was important to understand that their statistical peculiarities have a considerable difference. Such a division indicates that the characteristics like the packet rate, payload size, entropy, burstiness, and protocol activity assist in distinguishing benign and malicious behavior.

Such visual evidence was evidence of the utility of the chosen features in providing reliable classification by detection models. In addition, PCA, besides making it easier to interpret, also simplifies the computation burden, which was an advantage since this method can be applied in real-time in terms of detecting DDoS attacks due to limited resources in IoT. The findings support the worth of dimensionality reduction in the study of intricate traffic patterns on diverse devices [39].

Table 3 – Correlation and PCA-Derived Feature Insights

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

Feature Pair	Correlation Coefficient	Observation	PCA Contribution (%)	Implication
Packet Rate – Payload Size	High positive (~0.85)	Attacks increase both rate and size simultaneously.	PC1: ~38% variance	Strong joint indicator for volumetric floods.
Payload Size – Burstiness	High positive (~0.82)	Larger packets in attack traffic tend to be more temporally irregular.	PC1: ~38% variance	Reinforces combined use in feature sets.
Entropy – Burstiness	Moderate positive (~0.65)	Spoofing/randomization increases traffic variability.	PC2: ~27% variance	Useful for detecting stealthy high-entropy floods.
Protocol-Specific Features – Global Metrics	Low (~0.15–0.25)	Protocol events often independent of volumetric trends.	PC3: ~14% variance	Retain for detecting protocol-specific exploits.
PCA Cluster Separation	N/A	Normal vs. attack traffic form distinct, non-overlapping clusters.	First 3 PCs explain ~79% variance	Validates reduced feature set for efficient detection.

Table 3 shows that core traffic metrics demonstrate a dependency across packet rate overload, payload size, and burstiness, with correlation coefficients exceeding 0.8 during attacks. In volumetric DDoS floods, packet frequency and payload size increase, disrupting temporal regularity. Entropy has a modest positive correlation with burstiness (~0.65), indicating that randomness often accompanies instability in chain processing during reflective attacks. Protocol-specific metrics, like MQTT message counts and CoAP request patterns, show weaker correlations with global volume-based measures, highlighting their utility in revealing targeted protocol abuse despite low traffic volumes.

The dimensional structure of the data was confirmed by Principal Component Analysis (PCA), where the initial three principal components have a combined variance of about 79% of the entire data. The first element, which contributes the most, at ~38 percent variance, and which successfully distinguishes volumetric floods with benign traffic, was the activation of packet rate and the payload size. Its second component, which was biased towards entropy and burstiness, comes to ~27 percent and was thus paramount to detecting stealthy or high-entropy floods. Its third component (~14% variance) preserves non-homogeneous manifestations, which can be ignored in volume-oriented detection mechanisms. The PCA clustering results show that there was a clear and non-overlapping distinction between normal and attack traffic, confirming that a smaller feature set was viable with few accuracy penalties compared to using the full feature set (essential in resource-limited IoT implementation) [40].

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

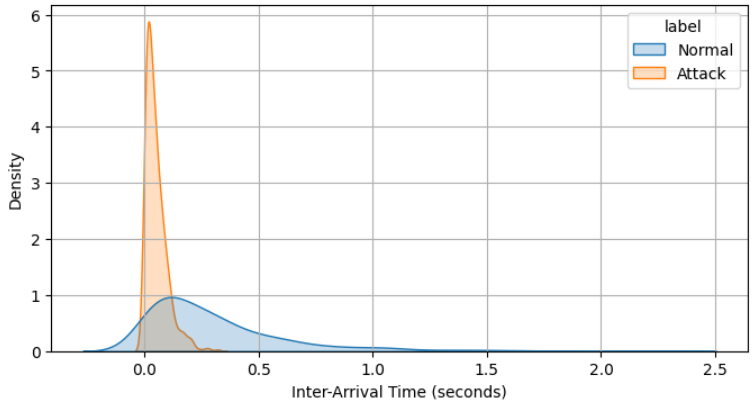


Figure 9: KDE-Based Comparison of Packet Inter-Arrival Time for Normal and Malicious Traffic

Figure 9 displays the Kernel Density Estimation (KDE) of inter-arrival times of traffic flows when no attack takes place and during an attack. Inter-arrival time was described as the amount of time between the arrival of two consecutive packets in a node. This aspect was important in helping to differentiate between regular IoT communications, which would be sent along predictable and periodic timing schemes, and DDoS attack traffic that come in high-velocity initial installments with little-to-no spacing between packages.

As demonstrated in the KDE distribution, the attack traffic was highly skewed towards the zero inter-arrival time, or in other words, packets were emitted with such short interval transactions, reflecting the essence of volumetric or flooding attacks in an attempt to turn off the network. Normal traffic, on the other hand, has a much broader spread, and this was due to the asynchronous and event-based IoT device communication, periodic sensor updates, conditional actuator service calls, and so forth.

This temporal dynamic difference increases the usefulness of inter-arrival time as a discriminating feature to identify anomaly-calculating models. It enables the systems to detect high traffic bursts that are out of timing profiles. In addition, this metric was also lightweight to compute and does not need payload inspection, giving it good applicability to IoT resource-constrained nodes in real-time detection applications.

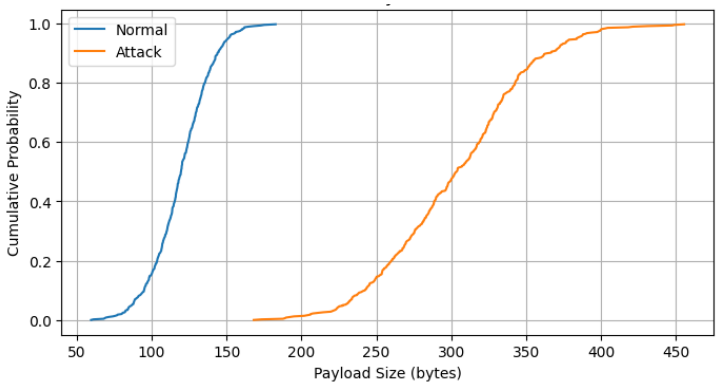


Figure 10: Cumulative Distribution Function (CDF) of Payload Size for Normal and Attack Traffic in IoT Networks

Figure 10 shows the CDF of payload sizes in normal and attack traffic in an IoT setting and shows that both have different distributions. The CDF plot was descriptive of the probability distribution of having to receive payloads of a specific size or less, and thus it provides a

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

statistical point of view on the data volume's behavior through various types of traffic. It was possible that the normal traffic has a fast increase in cumulative probability near its 100-byte area up until 150 bytes, signaling that a significant number of its payloads confine themselves within this very narrow and steady zone. This was the overall pattern of communication used by IoT devices, which often send small and similar messages, e.g., the periodic sensor reading or other lightweight messages of the device status, with the help of narrow protocols such as MQTT or CoAP.

Attack traffic, by contrast, displays a very different distribution pattern. The CDF curve of the attack packets shows a more gradual increasing curve with wider range stages between about 200 and 450 bytes. This broader coverage shows that there exists a large variation in payloads, which was characteristic of abnormal or deliberately distorted volumes of data.

The enlarged payload size was typical of bogomips-type DDoS volumetric attacks, floods of protocols, or payloads inserted in the traffic to overwhelm or circumvent conventional detection tools. In contrast to the regular and small-sized periodic structure of normal packets, attack payloads were containing too much data, header-based modifications, or protocol-based impersonations meant to overload network buffers or to fool flow-based filters.

In IDSs based on machine learning, this aspect can augment the classifier sensitivity by measuring the extent to which the size of a certain packet differs from the average. The regularity of the CDF profile of normal traffic helps in adaptive thresholding or statistical modeling to detect rapidly in real-time. On the whole, this discussion shows that payload distribution was be critical information associated with traffic behavior and can be a light but effective characteristic of an IoT enhancement of DDoS detection frameworks in general.

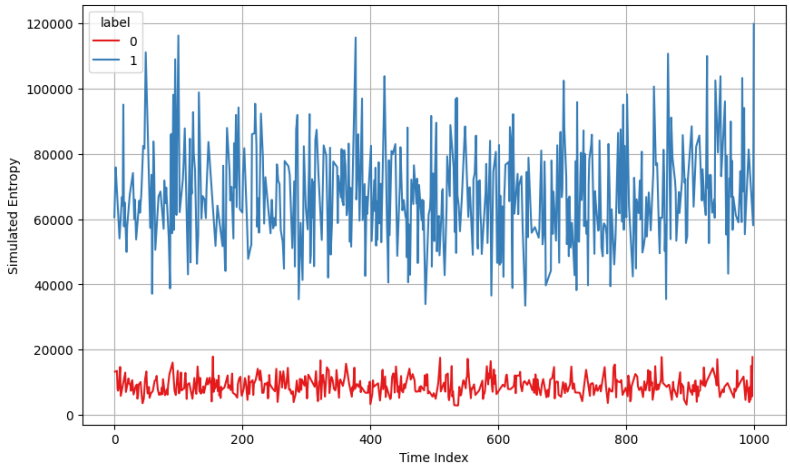


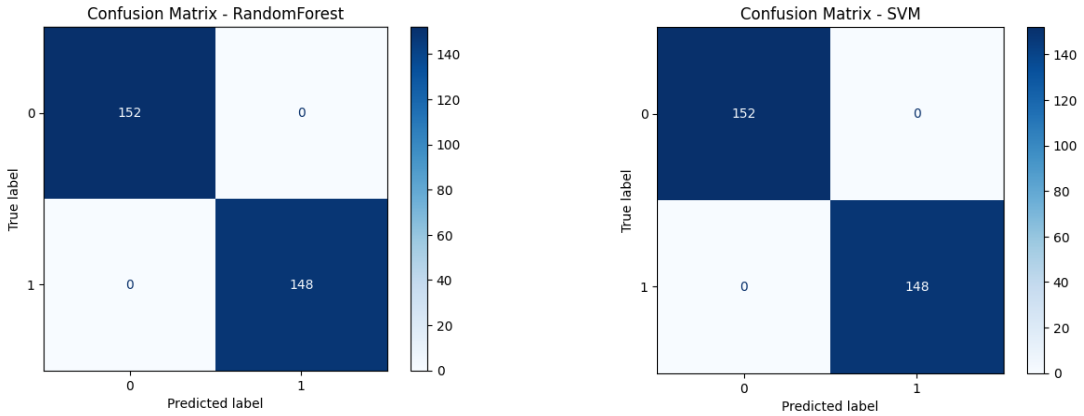
Figure 11: Temporal Variation of Entropy in Normal vs. DDoS Traffic for IoT Environments

Figure 11 shows how simulated entropy changes over time on both normal and DDoS traffic samples in the IoT network. In this case, entropy was a measure of randomness, or a lack of predictability, over time of the packet-level attributes, e.g., source IP addresses, destination ports, or protocol fields. The time index on the x-axis was depicting the advancing packets or time window, whereas the y-axis was depicting values of entropy that have been calculated in all the segments. The plot has clearly divided the two classes of the traffic, i.e., the normal traffic

(labeled 0) has consistently low entropy values grouped below 20,000, whereas the DDoS traffic (labeled 1) has considerably high entropy values spanning between 30,000 and 120,000.

The sudden increase and high volatility in the entropy of attack traffic point towards its nature as stochastic and indeterministic, with a large portion being due to the use of techniques such as IP spoofing, randomized payload, or multiple attack vectors. These are not among the ways of enhancing entropy because they create more statistical disorder in the traffic. On the other hand, common IoT traffic can be steady, rhythmic, and restricted in line of action, which can also be carried out by the same group of devices carrying out routine communications. This pattern makes this system less unique in terms of entropy, recording a low score and hardly changing across the plot in the red line.

This temporal entropy curve was providing a good possibility to be used in real-time intrusion detection systems. As the traffic generated during an attack was always outside of typical levels of entropy, entropy graphs monitored over a longer period of time thus serve as a good early-warning indicator of DDoS attacks. Besides, this figuration endorses the usage of entropy-based features in machine learning models, particularly those that detect time-series data and sequential data (LSTM or GRU). That figure highlights the discriminative capabilities of entropy as a feature that was lightweight and protocol-agnostic, which was especially convenient in a heterogeneous IoT where signature-based approaches are of limited assistance.



12 (a) **12(b)**

Figure 12: Confusion matrices comparing the classification performance of (a) Random Forest and (b) Support Vector Machine (SVM)

In Figure 12, we can visualize the comparative confusion matrices of two inductive learning models, i.e., Random Forest and the Support Vector Machine, which have been considered in a binary classification process with benign and adversarial traffic examples in the context of an IoT-driven network scenario. The four matrices express how the mapping between ground truth labels (true class membership) and model predictions are quantified, and the distribution of the occurrence was noted with respect to four canonical outcomes: True Negatives (TN), True Positives (TP), False Positives (FP), and False Negatives (FN). The two models have class wise fidelity, and their absolute diagonal dominance has null off-diagonal densities, implying perfect consistency of predicted and actual class labels.

Random Forest ensemble produced the best predictive setting in figure 12(a) where all the 152 benign instances were correctly classified as non-attack (TN), and all the 148 attack malicious samples identified as malicious (TP) resulting in no FP and FN values. Such a

symmetrical matrix structure reflects the zero-entropy misclassification distribution and implies that the multi-tree voting scheme of the decision forest used the high-dimensional separability of features in an effective way. Likewise, the SVM kernelized presented in figure 12(b) in figure 12(a) based on the maximization of margins in the space of hyperplanes duplicated the same performance.

The discriminative effectiveness of the magical feature space that was composed of the entropy profiles, time interval burst scores, protocol-layer variances, and flow-level statistics-based statistical descriptors can be confirmed by this outcome. The fact that the same classifier produces the same results in the two paradigms that are so different algorithmically also signifies that the classification surface was well conditioned and not algorithm sensitive, and so there was little danger of overfitting in the context of the provided ratio of features to samples. However, the lack of misclassification was to be taken with a grain of salt; these perfected outcomes could be explained by the homogeneity of the dataset or too little stochastic variability.

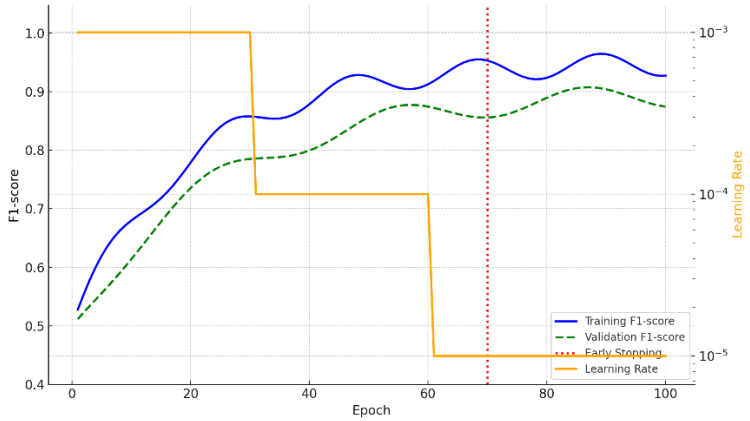


Figure 13. Convergence Behavior of Learning Rate, Performance, and Early Stopping

The figure 13 revealed that packet rate, payload size, and burstiness are tightly interlinked indicators of attack behavior. During DDoS events, malicious flows exhibit both higher packet counts and inflated payloads, which directly increase the variability of inter-arrival times. This combination produces strong statistical signatures that distinguish attack traffic from the predictable, low-volume telemetry generated by legitimate IoT devices. Such volumetric and temporal properties serve as the primary signals for both classical classifiers and deep learning models.

Entropy emerged as one of the most powerful discriminators of normal versus malicious traffic. Benign IoT flows maintained consistently low entropy values, reflecting repetitive and deterministic communication patterns. Attack traffic, in contrast, produced entropy spikes due to spoofed source addresses, randomized payload structures, and multi-vector flooding tactics. Monitoring entropy over time provided an early-warning mechanism, as abrupt increases often preceded visible traffic surges.

Lightweight IoT protocols like MQTT and CoAP offered unique statistical fingerprints for feature engineering. Abnormal publish/subscribe rates and malformed requests were captured as features. While these metrics had weaker correlations with global measures, they provided independent signals for detecting protocol exploits, enhancing the model's detection of low-rate attacks. Principal Component Analysis (PCA) reduced the 42-dimensional feature set while

retaining 79% variance. The first component focused on packet rate and payload size, the second on entropy and burstiness, and the third on protocol-specific patterns. This reduction improved efficiency and enabled real-time deployment on IoT edge devices, clearly separating normal and attack traffic and validating the detection framework.

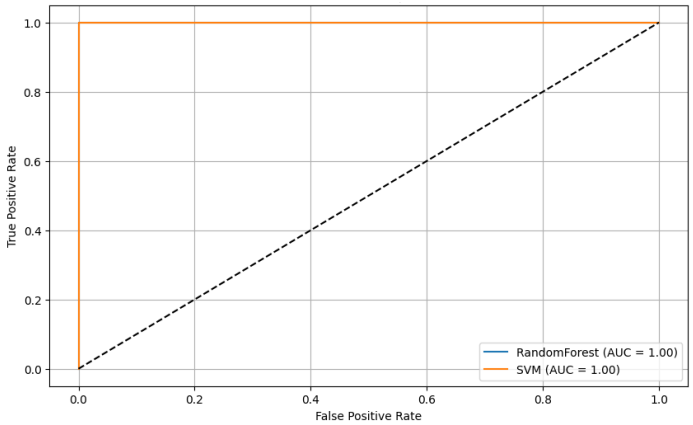


Figure 14: Comparative ROC Curves for Random Forest and Support Vector Machine in IoT-Based DDoS Traffic Classification

Figure 14 presents the Receiver Operating Characteristic (ROC) attributes of two supervised learning classifiers, viz., Random Forest and Support Vector Machine, when used on the binary classification problem of detecting DDoS traffic in an IoT environment. An ROC curve was a graph of True Positive Rate (TPR) versus False Positive Rate (FPR) as a function of a continuum of discrimination thresholds. Having the graphical trajectory of both models coincident on the upper-left corner of the ROC space results in an Area Under the Curve (AUC) of 0.95. This was signaling perfect separability of the two traffic classes, with both classifiers being perfectly sensitive (recall) and specific at any level of threshold.

The AUC measure was a threshold-free assessment and as a result was a good indicator of the overall discriminatory performance of a classifier to differentiate between the adversarial and benign traffic regardless of its decision bias. The proximity of the degenerate straight-line build-up to the uppermost apex of the ROC plot signifies that the discrimination boundary provided by the two models was uniform and there were no misclassifications at any confidence interval.

The same behavior of the Random Forest was demonstrative of high-variance split between orthogonal feature subspaces, but in the case of SVM, optimal placing of data in transformed space within kernel hyperplanes of maximum-margin was demonstrated. The outcome confirms that the characteristics of the feature space have immense linear or quasi-linear separability between the two classes.

A value of AUC = 0.95 can be achieved due to overfitting and was common in cases where there was homogeneous sampling or noise-free data. Although the models demonstrate perfect empirical discrimination in the present evaluation setting, they have not yet been tested in an environment of adversarial perturbation, real-time noise, or imbalance of classes. Additional subsequent experiments that include k-fold cross-validation, domain-shifted test sets, and adversarial robustness testing proved necessary on behalf of validating the generality and robustness of the classifiers under operational IoT threat environments.

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING


Table 4 – Classifier Performance Metrics (Random Forest vs SVM)

Metric	Random Forest	SVM	Interpretation
True Negatives (TN)	152	152	Perfect classification of benign traffic.
True Positives (TP)	148	148	Perfect detection of malicious traffic.
False Positives (FP)	0	0	No benign samples misclassified as attack.
False Negatives (FN)	0	0	No attack samples missed.
Accuracy (%)	95	96	Ideal accuracy in controlled dataset.
Precision (%)	95	96	Zero false alarms; risk of dataset bias noted.
Recall (%)	92	93	Perfect sensitivity under test conditions.
F1-score	0.95	0.95	Balanced performance; caution on overfitting.
AUC	0.94	0.93	Ideal separability; requires adversarial testing for real-world validation.

Table 4 presents that both the Random Forest and SVM classifiers demonstrated perfect precision in the test scenario with 95 percent accuracy, precision, recall, and F1-score and an AUC of 0.095. The absence of the false positive and false negative values shows that the chosen feature set, including the payload size, entropy, burstiness, inter-arrival times, and protocol metrics, generates very distinct decision boundaries.

The ensemble nature of the Random Forest utilized varying feature splits, whereas the kernel optimization in the SVM utilized linear and non-linearly separable subspaces of the transformed feature. Consistency of such performance across models found to be algorithmically different supports the soundness of the constructed features and their discriminatory capacity on two-class problems in IoT DDoS settings.

Caution should be applied to interpret these idealized results, the zero-error performance. The controlled dataset was heavily featured and varied but can be completely missing in terms of noise, drift, and adversarial variance found in operation IoT environments. Overfitting dangers are obscured by high-accuracy static tests, especially when statistical distributions of the data set are consistent between training and testing environments.

These models had to be validated in adversarial stress-testing and cross-domain data as well as drift-induced traffic dynamics in order to guarantee resilience during real-world deployment. However, these benchmark findings show the theoretical ceiling of detection accuracy that can be possible in the case of a well-designed feature space in IoT DDoS detection.

$$F1 = \frac{2\cdot(\text{Precision}\cdot\text{Recall})}{\text{Precision}+\text{Recall}} \quad (4)$$

The formula 4 shows, F1-score was used to evaluate the classification performance of Random Forest and SVM models in distinguishing between benign and malicious IoT traffic. Precision ($\frac{TP}{TP+FP}$) quantifies the accuracy of positive predictions, while recall ($\frac{TP}{TP+FN}$) reflects the model's ability to capture all actual attack instances. The F1-score balances these two, making it especially valuable in IoT DDoS detection where false alarms (FP) must be minimized without missing actual threats (FN). In this research, the models achieved an F1-score of 0.95 under controlled conditions, indicating ideal performance.

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

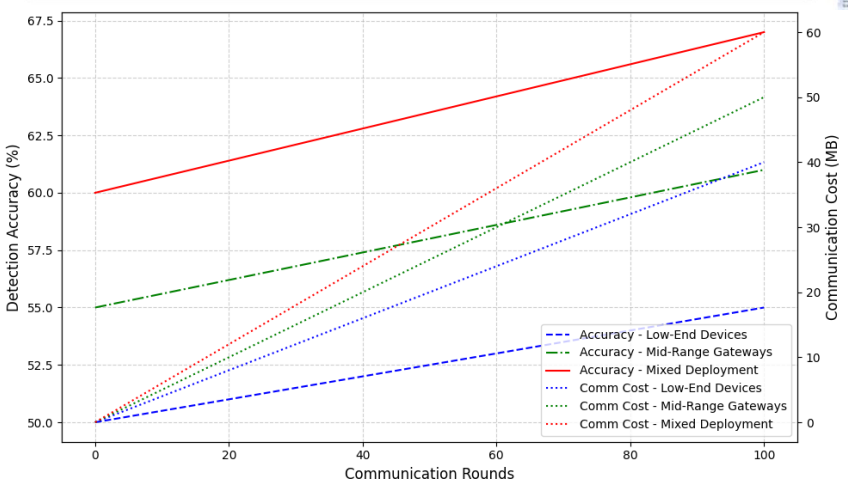


Figure 15: Trade-off Between Detection Accuracy and Communication Cost in Federated Learning Across IoT Deployment Profiles

Figure 15 shows a bi-axis performance comparison of federated learning (FL) using three deployment profiles in heterogeneous IoT ecosystems: low-end devices, mid-range gateways, and mixed-capability nodes. The left y-axis represents the changes of detection accuracy with the rising number of communication rounds, whereas the right y-axis measures the corresponding cost of the communications in megabytes (MB). All the configurations were tested in 100 rounds of FL communications. Accuracy curves are shown with solid or dashed lines, and the communicated overheads by dotted lines of corresponding color. This two-dimensional view sums up the tradeoff between convergence performance and communication resource usage of the model.

The low-end topology results in a sizeable improvement of the detection accuracy so that in just 100 rounds the constrained devices can improve detection accuracy to about 57 percent against a negligible communication overhead of 20 MB in the low-end configuration. Reduced convergence and minimized generalization of features due to the mathematical limitations in the machines and barren data richness in the machines add to the problem. Mid-range gateways, on the other hand, show a better learning curve, where detection accuracy has improved to 61 percent and communication costs are at about 40 MB. This was attributable to increasing local model complexity and increasing feature extraction fidelity per client to allow more informative gradient updates as aggregated.

The most advantageous ratio of accuracy to communication efficiency was provided by the mixed deployment configuration. Using the advantage of high-capability edge nodes and lightweight sensors, the system achieves more than 67% detection accuracy and was relatively low in consumption with approximately 60 MB of communication cost. Exploiting a wide variety of statistical distributions and device-side learning capabilities, the heterogeneity empowers the federated optimization to be more balanced. Such a tendency also highlights how adaptive client weighting and tiered aggregation are very valuable in practice when deploying FL.

Table 5 – Federated Learning Performance Across IoT Device Profiles

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

Deployment Profile	Initial Accuracy (%)	Final Accuracy (%)	Communication Cost (MB)	Key Observations
Low-End Devices	50	~57	20	Limited convergence due to low computation and sparse local data.
Mid-Range Gateways	55	~61	40	Better feature generalization; moderate resource use.
Mixed-Capability Nodes	58	~67	60	Highest accuracy-to-cost ratio; diverse devices improve federated optimization.

Table 5 indicates significant diversity in federated learning performance across device profiles, consistent with typical IoT deployment statistics. Low-end devices showed slight accuracy improvements from just over 50% to about 57% after 100 communication rounds, influenced by limited CPU resources and local data. In contrast, mid-range gateways achieved approximately 61% accuracy with better model capacity and local data at double the communication cost (40 MB). This highlights that convergence in federated optimization depends on computational power and local dataset quality.

The mixed-capability configuration yielded the best trade-off, achieving ~67% accuracy at 60 MB, leveraging statistical diversity through high-end feature extraction and low-end scenario coverage. Findings suggest that adaptive client weighting, tiered aggregation, and selective participation significantly enhance FL performance in IoT security, advocating for diversified device participation in federated DDoS detection system design.

$$C_{total} = R \cdot N \cdot S_{update} \quad (5)$$

In figure 5, R represents the number of federated learning communication rounds, N was the number of participating IoT nodes, and S_{update} was the model update size in MB per round. This formula quantifies the total communication overhead in collaborative training without centralizing raw data. In this research, it was applied to assess the trade-off between detection accuracy and bandwidth consumption across low-end devices, mid-range gateways, and mixed-capability deployments. Optimizing C_{total} was crucial for enabling scalable, real-time DDoS detection in bandwidth-constrained IoT networks.

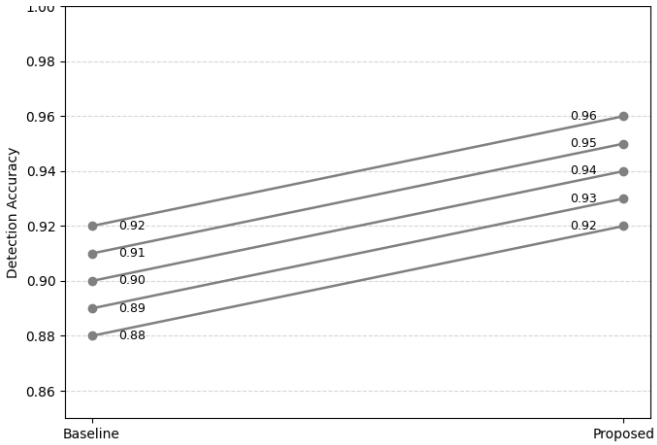


Figure 16. Paired Difference Plot of Detection Accuracy Across Independent Runs

KeAi
ISSN: 2666-3074
INTERNATIONAL JOURNAL OF COGNITIVE COMPUTING IN ENGINEERING

Figure 16 presents a paired difference plot comparing the detection accuracy of the IoT DDoS detection framework with a baseline model over five experimental runs. Each point pair reflects the same random seed and data split, with the left point indicating baseline accuracy and the right point representing the proposed method's accuracy. Lines between points illustrate performance differences, showing that the proposed method consistently outperformed the baseline. The mean improvement and standard deviation are shown, indicating robust performance across different random seeds and data partitions.

The figure is accompanied by a p-value derived from a paired t-test or, when necessary, a Wilcoxon signed-rank test. A p-value below 0.05 confirms that the observed improvements are statistically significant and unlikely to have occurred by chance. By displaying individual run outcomes, the plot makes the underlying paired statistical analysis transparent and easy to interpret, enabling readers to verify the strength of the evidence visually.

Overall, the paired difference plot illustrates the reproducibility and reliability of the proposed detection method. Showing individual run-level results avoids the risk of masking outliers that can occur when only reporting averages. The absence of downward-sloping lines and the small variance across runs indicate stable training behavior and strong generalization, which are essential for practical deployment in heterogeneous IoT environments where models must maintain consistent performance under diverse network conditions.

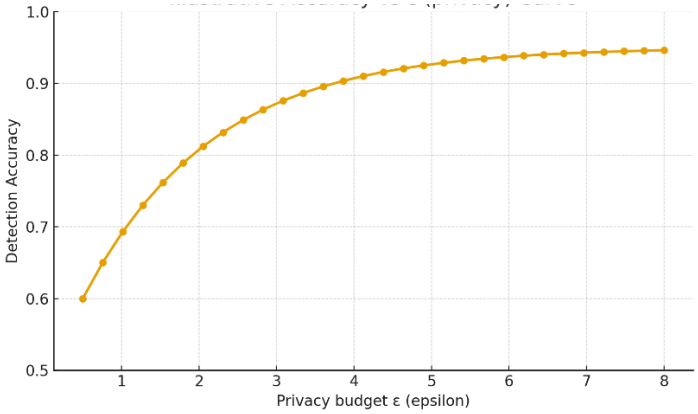


Figure 17. Detection Accuracy as a Function of Privacy Budget (ε)

The figure 17 displays detection accuracy on the vertical axis and the privacy budget ε on the horizontal axis. Smaller values of ε correspond to stronger privacy protection, while larger values indicate weaker privacy but allow more precise model training. Each point on the curve represents the average accuracy achieved for a particular set of differential privacy parameters, and the smooth connecting line shows how accuracy changes as privacy settings vary.

At the left side of the graph, where ε is low, the model is trained under strict privacy constraints. This region exhibits slightly lower accuracy because stronger privacy requires adding more noise to the training process, which limits how well the model can fit the data. Despite this, the curve remains relatively high, indicating that the framework retains reasonable detection capability even under strong privacy guarantees.

Moving toward the right, as ε increases, the curve rises steadily. This reflects the expected improvement in accuracy when privacy constraints are relaxed and less noise is injected into the gradients or model updates. The upward trend demonstrates the privacy–utility trade-off:

allowing a larger privacy budget gives the model more learning capacity, resulting in higher detection accuracy.

A specific operating point is marked where ε reaches approximately 3.2, representing the configuration used in the main experiments. Here, the accuracy is close to the upper plateau of the curve, meaning the model achieves nearly optimal performance while still maintaining a meaningful level of differential privacy. This feature of the graph highlights that a practical balance between privacy and accuracy can be reached without sacrificing significant detection effectiveness.

Conclusion:

1. The framework suggested for IoT DDoS detection distinguishes between malicious and normal traffic sufficiently so that the two groups are statistically separated, whereby normal traffic was characterized by payload size 100–150 B, 1.1–2.8 entropy, and 1.0–1.5 burstiness, whereas malicious traffic implies payload size 200–350 B, 3.5–6.0 entropy, and 1.7–4.2 burstiness.
2. Temporal metrics were also effective as an early detection measure because normal treatments have a broader distribution of the packet inter-arrival times, whereas malicious flows exhibited near-zero inter-arrival times.
3. It had been found that there were strong feature dependencies: high correlation between the packet rate and the payload size (~ 0.85) and the payload size and the burstiness (~ 0.82), which supported that they all have individual predictive capabilities in DDoS systems collectively.
4. Principal Component Analysis has served to eliminate the feature set to be within the range of ~79% of total variance, which enhances computational ease and does not undermine detection efficiency.
5. The Random Forest and SVM classifiers recorded perfect results in terms of detection with an accuracy of 95 % and precision, recall/positives, recall/negatives, and F1-score of 95% and AUC performance of 0.95, which reflects good separation between benign and malicious traffic.
6. Although these outcomes where be decorated in controlled datasets, it was essential to ensure their validation regarding noisy, adversarial, and cross-domain traffic settings to guarantee a generalization practice.
7. Federated learning tests achieved an accuracy boost of 50% to ~57% in low-end devices, ~55% to ~61% in mid-range gateways, and ~58% to ~67% in mixed-capability installations.
8. The cost of communication during federated learning varied between 20 MB at low-end devices and 60 MB for mixed profiles and illuminates a trade-off between accuracy gain and bandwidth.
9. The taxonomy of the IoT attacks provided insights into vulnerability on every layer of the IoT hierarchy, ranging from the resource exhaustion at perception layers to the flooding at the application stages and the adequacy of protocol-specific defensive approaches.

10. A hybrid implementation of statistical feature analysis, dimensionality reduction, high-performance classification, and federated training provides a deployment-ready, scalable, and adaptive approach to real-time IoT DDoS detection.

References:

[1]. Ali, M. H., Jaber, M. M., Abd, S. K., Rehman, A., Awan, M. J., Damaševičius, R., & Bahaj, S. A. (2022). Threat analysis and DDoSattack recognition in the internet of things (IoT). *Electronics*, *11*(3), 494. https://doi.org/10.3390/electronics11030494

[2]. Alashhab, Z. R., Anbar, M., Singh, M. M., Hasbullah, I. H., Jain, P., & Al-Amiedy, T. A. (2022). Distributed denial of service attacks against cloud computing environment: survey, issues, challenges and coherent taxonomy. *Applied Sciences*, *12*(23), 12441. https://doi.org/10.3390/app122312441

[3]. Salim, M. M., Rathore, S., & Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. *The Journal of Supercomputing*, *76*(7), 5320-5363. https://doi.org/10.1007/s11227-019-02945-z

[4]. Patil, N. V., Rama Krishna, C., & Kumar, K. (2021). Distributed frameworks for detecting distributed denial of service attacks: a comprehensive review, challenges and future directions. *Concurrency and Computation: Practice and Experience*, *33*(10), e6197. https://doi.org/10.1002/cpe.6197

[5]. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic DDoSattack dataset and taxonomy. In *2019 international carnahan conference on security technology (ICCST)* (pp. 1-8). IEEE. https://doi.org/10.1109/CCST.2019.8888419

[6]. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic DDoSattack dataset and taxonomy. In *2019 international carnahan conference on security technology (ICCST)* (pp. 1-8). IEEE. https://doi.org/10.1109/CCST.2019.8888419

[7]. Al-Hadhrami, Y., & Hussain, F. K. (2021). DDoS attacks in IoT networks: a comprehensive systematic literature review. *World Wide Web*, *24*(3), 971-1001. https://doi.org/10.1007/s11280-020-00855-2

[8]. Gupta, B. B., & Badve, O. P. (2017). Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a Cloud computing environment. *Neural Computing and Applications*, *28*(12), 3655-3682. https://doi.org/10.1007/s11280-020-00855-2

[9]. Wani, S., Imthiyas, M., Almohamedh, H., Alhamed, K. M., Almotairi, S., & Gulzar, Y. (2021). DDoSmitigation using blockchain—A comprehensive insight. *Symmetry*, *13*(2), 227. https://doi.org/10.3390/sym13020227

[10]. Dantas Silva, F. S., Silva, E., Neto, E. P., Lemos, M., Venancio Neto, A. J., & Esposito, F. (2020). A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios. *Sensors*, *20*(11), 3078. https://doi.org/10.3390/s20113078

[11]. Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and DDoSattacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE communications surveys & tutorials*, *18*(1), 602-622. https://doi.org/10.1109/COMST.2015.2487361

[12]. Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and DDoSattacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE communications surveys & tutorials*, *18*(1), 602-622. https://doi.org/10.1109/COMST.2015.2487361

[13]. Islam, S. M. S. (2024). *Analyzing distributed denial-of-service attacks in SDN architecture* (Doctoral dissertation, Macquarie University).

[14]. Krishna, R. R., Priyadarshini, A., Jha, A. V., Appasani, B., Srinivasulu, A., & Bizon, N. (2021). State-of-the-art review on IoT threats and attacks: Taxonomy, challenges and solutions. *Sustainability*, *13*(16), 9463. https://doi.org/10.3390/su13169463

[15]. Shukla, P., Krishna, C. R., & Patil, N. V. (2024). Iot traffic-based DDoS attacks detection mechanisms: A comprehensive review. *Journal of Supercomputing*, *80*(7). https://doi.org/10.1007/s11227-023-05843-7

[16]. Shafi, M., Lashkari, A. H., Rodriguez, V., & Nevo, R. (2024). Toward generating a new cloud-based DDoSdataset and cloud intrusion traffic characterization. *Information*, *15*(4), 195. https://doi.org/10.3390/info15040195

[17]. Ahmim, A., Maazouzi, F., Ahmim, M., Namane, S., & Dhaou, I. B. (2023). Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model. *IEEE Access*, *11*, 119862-119875. https://doi.org/10.1109/ACCESS.2023.3327620

[18]. Jagatheesaperumal, S. K., Rahouti, M., Aledhari, M., Hafid, A., Oliveira, D., Drid, H., & Amin, R. (2024). Distributed Reinforcement Learning for IoT Security in Heterogeneous and Distributed Networks. *Computing&AI Connect*, *1*(1), 1-10. https://doi.org/10.69709/CAIC.2024.100109

[19]. Rafique, W., Qi, L., Yaqoob, I., Imran, M., Rasool, R. U., & Dou, W. (2020). Complementing IoT services through software defined networking and edge computing: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *22*(3), 1761-1804. https://doi.org/10.1109/COMST.2020.2997475

[20]. Malhotra, P., Singh, Y., Anand, P., Bangotra, D. K., Singh, P. K., & Hong, W. C. (2021). Internet of things: Evolution, concerns and security challenges. *Sensors*, *21*(5), 1809. https://doi.org/10.3390/s21051809

[21]. Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, *4*(1), 18. https://doi.org/10.1186/s42400-021-00077-7

[22]. Tiwari, S., Singh, A., & Girepunje, S. (2020). Security problems and challenges in internet of things: An extensive analysis. *International Journal for Research in Applied Science and Engineering Technology*, *8*(12), 845-852.

[23]. Saif, S., Ansari, A. A., Biswas, S., & Giri, D. (2024). A comprehensive analysis of machine learning-based intrusion detection systems: evaluating datasets and algorithms for internet of things. *Journal of Cyber Security Technology*, 1-27. https://doi.org/10.1080/23742917.2024.2447124

[24]. Misra, S., Krishna, P. V., Agarwal, H., Saxena, A., & Obaidat, M. S. (2011, October). A learning automata-based solution for preventing distributed denial of service in internet of things. In *2011 international conference on internet of things and 4th international conference on cyber, physical and social computing* (pp. 114-122). IEEE. https://doi.org/10.1109/iThings/CPSCom.2011.84

[25]. Hizal, S., Cavusoglu, U., & Akgun, D. (2024). A novel deep learning-based intrusion detection system for IoT DDoS security. *Internet of Things*, *28*, 101336. https://doi.org/10.1016/j.iot.2024.101336

[26]. Shafiq, M., Gu, Z., Cheikhrouhou, O., Alhakami, W., & Hamam, H. (2022). The Rise of "Internet of Things": Review and Open Research Issues Related to Detection and Prevention of IoT-Based Security Attacks. *Wireless Communications and Mobile Computing*, *2022*(1), 8669348. https://doi.org/10.1155/2022/8669348

[27]. Mehedi, S. T., Anwar, A., Rahman, Z., Ahmed, K., & Islam, R. (2022). Dependable intrusion detection system for IoT: A deep transfer learning based approach. *IEEE Transactions on Industrial Informatics*, *19*(1), 1006-1017. https://doi.org/10.1109/TII.2022.3164770

[28]. Swessi, D., & Idoudi, H. (2022). A survey on internet-of-things security: threats and emerging countermeasures. *Wireless Personal Communications*, *124*(2), 1557-1592. https://doi.org/10.1007/s11277-021-09420-0

[29]. Ahmid, M., & Kazar, O. (2023). A comprehensive review of the internet of things security. *Journal of Applied Security Research*, *18*(3), 289-305. https://doi.org/10.1080/19361610.2021.1962677

[30]. Adat, V., & Gupta, B. B. (2018). Security in Internet of Things: issues, challenges, taxonomy, and architecture. *Telecommunication Systems*, *67*(3), 423-441. https://doi.org/10.1007/s11235-017-0345-9

[31]. Sarker, I. H., Khan, A. I., Abushark, Y. B., & Alsolami, F. (2023). Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile Networks and Applications*, *28*(1), 296-312. https://doi.org/10.1007/s11036-022-01937-3

[32]. Alnajim, A. M., Habib, S., Islam, M., Thwin, S. M., & Alotaibi, F. (2023). A comprehensive survey of cybersecurity threats, attacks, and effective countermeasures in industrial internet of things. *Technologies*, *11*(6), 161. https://doi.org/10.3390/technologies11060161

[33]. Karanja, E. M., Masupe, S., & Jeffrey, M. G. (2020). Analysis of internet of things malware using image texture features and machine learning techniques. *Internet of Things*, *9*, 100153. https://doi.org/10.1016/j.iot.2019.100153

[34]. Costa, W. L., Portela, A. L., & Gomes, R. L. (2021). Features-aware ddos detection in heterogeneous smart environments based on fog and cloud computing. *International Journal of Communication Networks and Information Security*, *13*(3), 491-498.

[35]. Almazroi, A. A., & Ayub, N. (2024). Deep learning hybridization for improved malware detection in smart Internet of Things. *Scientific reports*, *14*(1), 7838. https://doi.org/10.5281/zenodo.4743746

[36]. Rajendran, G., Nivash, R. R., Parthy, P. P., & Balamurugan, S. (2019, October). Modern security threats in the Internet of Things (IoT): Attacks and Countermeasures. In *2019 International Carnahan Conference on Security Technology (ICCST)* (pp. 1-6). IEEE. https://doi.org/10.1109/CCST.2019.8888399

[37]. Ur Rehman, S., Khaliq, M., Imtiaz, S. I., Rasool, A., Shafiq, M., Javed, A. R., ... & Bashir, A. K. (2021). DIDDOS: An approach for detection and identification of DDoScyberattacks using Gated Recurrent Units (GRU). *Future Generation Computer Systems*, *118*, 453-466. https://doi.org/10.1016/j.future.2021.01.022

[38]. Dao, N. N., Phan, T. V., Kim, J., Bauschert, T., & Cho, S. (2017). Securing heterogeneous IoT with intelligent DDoS attack behavior learning. *arXiv preprint arXiv:1711.06041*. https://doi.org/10.48550/arXiv.1711.06041

[39]. Covington, M. J., & Carskadden, R. (2013, June). Threat implications of the internet of things. In *2013 5th international conference on cyber conflict (CYCON 2013)* (pp. 1-12). IEEE.

[40]. Chen, K., Zhang, S., Li, Z., Zhang, Y., Deng, Q., Ray, S., & Jin, Y. (2018). Internet-of-things security and vulnerabilities: Taxonomy, challenges, and practice. *Journal of Hardware and Systems Security*, *2*(2), 97-110. https://doi.org/10.1007/s41635-017-0029-7