Attack-to-Defense IoT: A Physical IoT Security Testbed for Labeled Dataset Generation and Machine Learning-Based Attack Detection

Hebert Silva[1,2], Mateus Lopes[1], João Luisi[1], Tiago Demay[1], Regina Moraes[2]

[1] Serviço Nacional de Aprendizagem Industrial (SENAI)
Escola SENAI Paulo Antonio Skaf São Caetano do Sul – SP – Brasil

[2]Universidade Estadual de Campinas (UNICAMP)
Faculdade de Tecnologia (FT)– Limeira – SP – Brasil

hebert.oliveira@sp.senai.br, h144617@dac.unicamp.br
mateus.l.oliveira11@aluno.senai.br
joao.luisi@sp.senai.br, tiago.demay@sp.senai.br
regina@ft.unicamp.br

Abstract. *The increasing adoption of Internet of Things (IoT) technologies has significantly expanded the attack surface of Cyber-Physical Systems (CPS), making realistic experimental environments essential for cybersecurity research. This work presents a low-cost and reproducible IoT cybersecurity testbed designed for realistic attack execution, multimodal labeled dataset generation, and Machine Learning (ML)-based intrusion detection. The proposed environment integrates ESP32-based embedded devices, MQTT/TLS communication, AWS cloud services, automated telemetry extraction pipelines, and supervised ML models. The infrastructure supports realistic attack scenarios including MQTT flooding, Sliver Command and Control (C2) beaconing, Denial-of-Service attacks, network scanning, and brute-force attacks against embedded services. The generated datasets combine PCAP-derived network traffic, MQTT telemetry, and sensor-generated events into a unified multimodal representation for intrusion detection experiments. Experimental results indicate that supervised ML algorithms, particularly Random Forest, are effective for detecting volumetric attacks and heterogeneous malicious traffic patterns in constrained IoT environments. The proposed platform contributes a reproducible physical infrastructure for IoT cybersecurity experimentation and establishes a foundation for future research involving Federated Learning (FL), TinyML, and distributed intrusion detection for CPS.*

1. Introduction

The rapid growth of Internet of Things (IoT) technologies has transformed modern digital infrastructures by enabling large-scale connectivity between embedded devices, cloud services, and Cyber-Physical Systems (CPS). IoT devices are increasingly deployed in residential, industrial, healthcare, and critical infrastructure environments, where they perform automation, monitoring, and operational control tasks. However, the limited computational resources and lightweight communication protocols commonly adopted in constrained environments make IoT infrastructures highly vulnerable to cyberattacks [Meidan et al. 2018, Ferrag et al. 2020]. Among the communication technologies widely

used in embedded systems, MQTT (Message Queuing Telemetry Transport) has become one of the most adopted protocols due to its lightweight publish/subscribe model and low communication overhead [OASIS 2019]. Despite its operational efficiency, MQTT infrastructures remain susceptible to attacks involving topic flooding, insecure authentication, broker exploitation, and malicious telemetry manipulation.

Machine Learning (ML)-based intrusion detection approaches have been extensively investigated for IoT security because of their capability to identify anomalous communication patterns in network traffic and telemetry data [Ferrag et al. 2020]. Nevertheless, many existing studies still rely on synthetic datasets, simulated traffic, or highly controlled environments that do not accurately represent real-world IoT deployments. In addition, reproducible physical testbeds integrating MQTT telemetry, realistic attack execution, multimodal data collection, and hybrid edge/cloud infrastructures remain scarce in the current literature [Silva 2022, Silva and Moraes 2024]. To address these limitations, this work presents a low-cost and reproducible IoT cybersecurity testbed designed for labeled dataset generation and ML-based intrusion detection. The proposed environment integrates physical IoT devices, MQTT/TLS communication, cloud-based services, and realistic attack orchestration to support intrusion detection experiments in embedded environments.

The main scientific contributions of this work are: i) a reproducible physical IoT cybersecurity testbed; ii) an automated multimodal dataset generation pipeline; iii) integration of edge and cloud infrastructures for telemetry collection and analysis; iv) realistic attack orchestration against embedded IoT devices; v) comparative evaluation of supervised ML algorithms; and vi) an extensible foundation for future FL and distributed intrusion detection experiments.

The rest of the paper is organized as follows: Section 2 presents the background necessary to understand the proposal and related work; Section 3 presents the proposed architecture; dataset generation pipeline is presented in Section 5; the methodology based on ML is presented in Section 5; Results and Discussion can be find in Section 6; ending in Section 7 with conclusions and future works.

2. Background and Related Work

Internet of Things (IoT) and Cyber-Physical Systems (CPS) environments commonly rely on constrained embedded devices and lightweight communication protocols. Among these protocols, MQTT (Message Queuing Telemetry Transport) is widely adopted because of its lightweight publish/subscribe architecture and low communication overhead [OASIS 2019]. However, MQTT-enabled infrastructures remain exposed to threats involving insecure authentication, broker exploitation, topic flooding, malicious telemetry manipulation, and service disruption. Although TLS and mutual authentication (mTLS) can protect communications between devices and cloud services, these mechanisms do not eliminate the need for monitoring and intrusion detection capabilities [Fagan et al. 2020, International Electrotechnical Commission 2021].

Machine Learning (ML) and Deep Learning techniques have been extensively investigated for intrusion detection in IoT environments because of their ability to identify anomalous patterns in network traffic and application-level telemetry [Ferrag et al. 2020]. Foundational contributions have addressed network-based anomaly detection using traf-

fic generated by physical IoT devices, including devices affected by Mirai and Bashlite botnets [Meidan et al. 2018]. Other studies have investigated encrypted traffic classification, feature engineering, and dimensionality-reduction techniques for improving classification efficiency in constrained environments [Rezaei and Liu 2019, Silva et al. 2021, Verma and Patel 2024]. These works provide important methodological foundations, but generally focus on specific analytical tasks or previously collected datasets rather than the complete process of physical attack execution, synchronized data acquisition, and automated dataset generation.

Federated Learning (FL) and distributed intrusion-detection approaches have also been investigated as alternatives for improving privacy, scalability, and collaboration across IoT and CPS infrastructures. Instead of centralizing raw traffic data, FL allows edge nodes to train local models and share only model parameters or gradients [Nguyen et al. 2021]. Existing studies have explored FL-based intrusion detection in MQTT-enabled and distributed IoT environments [Alshamrani and Anwar 2022, Lazzarini et al. 2023, Omotosho et al. 2023]. However, these approaches frequently rely on existing datasets, simulated MQTT traffic, or architectures without a physical multimodal data-generation workflow.

Conceptual frameworks have also addressed AI-based cybersecurity characterization and privacy-preserving intrusion detection for CPS and distributed IoT environments [Silva 2022, Silva and Moraes 2024]. Similarly, containerized MQTT environments support reproducible vulnerability analysis and attack experimentation [Freitas et al. 2024]. Nevertheless, containerized or conceptual environments do not necessarily reproduce the interactions among physical devices, sensor events, network traffic, MQTT telemetry, and cloud services found in operational IoT deployments. In the present work, FL is considered a future architectural extension and is not part of the current experimental evaluation.

Recent research has further advanced the development of MQTT-oriented cybersecurity datasets and physical IoT testbeds. These initiatives include physical environments focused on DoS and DDoS attacks, public datasets composed of labelled MQTT traffic, domain-specific e-health platforms, and heterogeneous smart-home or smart-office infrastructures [Alatram et al. 2023, Aveleira-Mata et al. 2025, Aqachtoul et al. 2025, Farag et al. 2025]. Although these studies provide relevant resources for intrusion-detection research, their scope commonly remains concentrated on particular attack families, application domains, device configurations, or predominantly network-level data.

Table 1 summarizes the studies discussed in this section. The comparison, which addresses complementary aspects of IoT experimentation, considers the type of contribution, the use of physical devices and MQTT, the adopted data sources, attack coverage, learning strategy, and the main distinction of each study in relation to the proposed platform. Foundational and methodological studies provide detection models, feature-engineering strategies, and relevant datasets. Federated, distributed, and conceptual approaches mainly address privacy, collaboration, and scalability, but commonly depend on existing datasets, simulations, or non-physical environments. Recent MQTT testbeds provide realistic traffic and valuable attack datasets, although they frequently concentrate on specific attack classes, application domains, or individual data sources.

Table 1. Comparison of the academic works discussed

Work	Type	Physical / MQTT	Data sources	Attacks or scope	Learning	Main difference from this work
Foundational and methodological studies						
[Ferrag et al. 2020]	Survey	No / No	Multiple public cybersecurity datasets	Deep Learning approaches for intrusion detection	DL review	Does not generate a new dataset or provide an MQTT-oriented physical testbed.
[Meidan et al. 2018]	Dataset and testbed	Yes / No	Network traffic from nine commercial IoT devices	Mirai and Bashlite botnet activity	Deep autoencoders	Focuses on botnet detection using network traffic without MQTT, cloud processing, or multimodal data.
[Rezaei and Liu 2019]	Survey	No / No	Studies and datasets involving encrypted traffic	Encrypted traffic classification	DL review	Does not address MQTT attacks, physical IoT experimentation, or dataset generation.
[Silva et al. 2021]	Methodol. study	No / No	CPS experimental result datasets	Feature selection and dimensionality reduction	Centralized ML	Focuses on feature selection rather than physical attack orchestration and data collection.
[Verma and Patel 2024]	Methodol. study	No / No	IoT network traffic features	Efficient feature sets for IoT intrusion detection	ML feature analysis	Does not introduce a physical testbed or a synchronized multimodal dataset.
Federated, distributed, and conceptual approaches						
[Nguyen et al. 2021]	Survey	No / No	Literature on FL applications in IoT	Privacy, communication, and scalability challenges	FL survey	Provides a general FL taxonomy without an MQTT intrusion-detection testbed.
[Alshamrani and Anwar 2022]	Exp. study	No / Yes	MQTT traffic or existing IoT security datasets	Distributed MQTT intrusion-detection scenarios	Federated ML	Explores FL-based MQTT intrusion detection without providing a multimodal physical edge/cloud platform.
[Lazzarini et al. 2023]	Exp. study	No / No	Existing IoT intrusion-detection datasets	Generic IoT intrusion scenarios	Federated DL	Evaluates FL using existing datasets without physical MQTT data generation.
[Omotosho et al. 2023]	Simulation	No / Yes	Simulated MQTT traffic and an existing MQTT dataset	Seven MQTT attacks, including DoS and MitM scenarios	Centralized ML and FL	Uses a simulated MQTT environment without physical devices or multimodal edge/cloud collection.
[Silva 2022]	Framework	No / No	CPS security characteristics and indicators	AI-based cybersecurity characterization	AI framework	Introduces a CPS cybersecurity characterization model without generating a physical intrusion-detection dataset.
[Silva and Moraes 2024]	Framework extension	No / No	Privacy and security scenarios for distributed IoT	Privacy-preserving intrusion-detection challenges	FL-oriented framework	Discusses privacy-preserving mechanisms without physical experimental validation.
[Freitas et al. 2024]	Containerized testbed	No / Yes	MQTT broker and client configurations	MQTT protocol vulnerabilities and exploitation scenarios	No ML evaluation	Provides a container-based vulnerability environment rather than a physical multimodal IDS platform.
MQTT datasets and physical IoT testbeds						
[Alatram et al. 2023]	Dataset and testbed	Yes / Yes	PCAP-derived MQTT network traffic	Ten MQTT-oriented DoS and DDoS scenarios	Conventional ML	Concentrates on denial-of-service attacks and network-level traffic.
[Aveleira-Mata et al. 2025]	Dataset and testbed	Yes / Yes	Labelled CSV network traffic with MQTT protocol features	DoS, Man-in-the-Middle, and intrusion attacks	ML validation	Provides a reusable dataset without sensor events, cloud processing, or a multimodal pipeline.
[Aqachtoul et al. 2025]	Dataset and testbed	Yes / Yes	TCP and MQTT traffic from an e-health IoT environment	DoS, SlowITe, malformed data, brute force, and publish flooding	AI/ML benchmarking	Uses a domain-specific e-health scenario and remains primarily network-traffic oriented.
[Farag et al. 2025]	Physical testbed	Yes / Not central	Traffic from heterogeneous smart-home and smart-office devices	Unauthorized access, scanning, DoS, and intrusion scenarios	XGBoost and SVM	Provides heterogeneous physical experimentation but not a synchronized MQTT-oriented multimodal pipeline.
Attack-to-Defense IoT	**Physical platform**	**Yes / Yes**	**PCAP, MQTT telemetry, cloud data, images, and sensor events**	**MQTT flooding, Sliver C2, DoS, scanning, and brute force**	**RF, DT, KNN, SVM, and MLP**	**Integrates ESP32 devices, MQTT/TLS, realistic attacks, automated labelling, multimodal data, and edge/cloud processing. FL and TinyML remain future extensions.**

The resulting research gap is not the absence of MQTT datasets or physical IoT testbeds, but the limited integration of the complete experimental workflow within a single reproducible platform. Existing environments do not commonly combine low-cost embedded devices, secure MQTT communication, heterogeneous attack execution, synchronized network and application-level data acquisition, sensor-generated events, automated labeling, cloud processing, and supervised ML evaluation. Attack-to-Defense IoT addresses this gap through a physical edge/cloud platform that integrates ESP32-based devices, MQTT over TLS, realistic attack orchestration, PCAP traffic collection, MQTT telemetry extraction, and automated dataset construction. The broader infrastructure supports the collection of network traffic, MQTT messages, cloud data, image-related events, and physical sensor events. However, the ML evaluation reported in this study specifically uses PCAP-derived network features and MQTT telemetry attributes. The platform also establishes an experimental foundation for future investigations involving FL, TinyML, and distributed intrusion detection in IoT and CPS environments.

3. Proposed Architecture

This section presents the proposed IoT cybersecurity architecture designed for realistic attack generation, multimodal dataset collection, and ML-based intrusion detection. The environment integrates ESP32-based devices, MQTT/TLS communication, cloud services, and automated feature extraction pipelines within a unified edge/cloud infrastructure. To ensure reproducibility and foster further research, the source code, hardware firmware, and configuration files are available in a public repository[1]. Figure 1 presents the overall architecture of the proposed environment.

The local IoT infrastructure is centered on an ESP32-C5 operating in STA+AP mode, enabling isolated traffic monitoring and PCAP collection. Two embedded devices are connected to this environment: an ESP32-CAM AI Thinker and an ESP32-S3 M5Stack Cardputer. The ESP32-CAM performs motion detection and publishes image events through MQTT/TLS communication, while the Cardputer transmits telemetry data and also acts as an attack orchestration device for MQTT flooding and malicious traffic generation. Figure 2 presents the IoT devices adopted in the experiments.

Communication between IoT devices and cloud services is performed using MQTT over TLS with certificates generated through a custom Public Key Infrastructure (PKI). A Mosquitto MQTT broker hosted on AWS EC2 acts as the central communication hub, while cloud-based subscriber services consume MQTT topics for telemetry extraction and image forwarding to Amazon S3. Figure 3 illustrates the MQTT/TLS communication workflow between IoT devices and cloud services. The cloud infrastructure also processes PCAP traffic collected from the local IoT network and extracts network-level and MQTT-level features for dataset generation and ML experiments. The resulting datasets combine network traffic, MQTT telemetry, and sensor events into a unified multimodal representation.

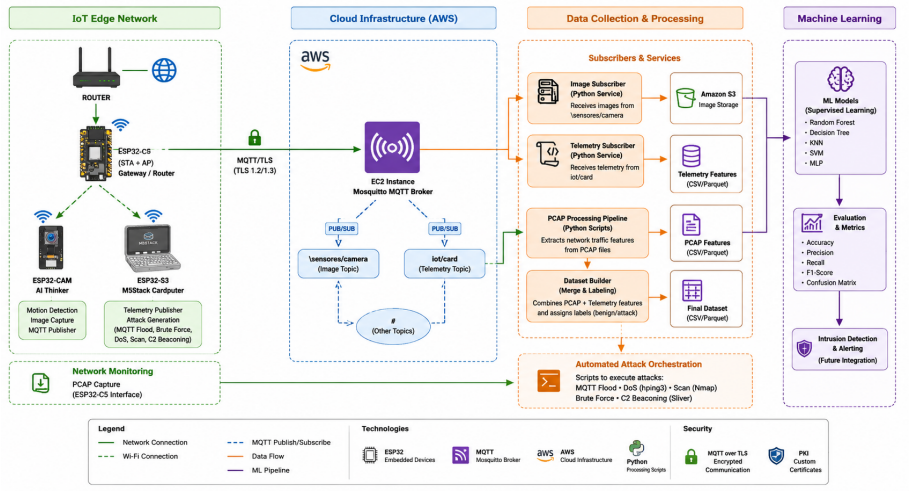

Figure 1. Proposed IoT cybersecurity architecture for attack generation, dataset collection, and Machine Learning-based intrusion detection.

The proposed environment supports realistic cyberattack execution against

[1]https://github.com/csai4cps/Attack-to-Defense_IoT

MQTT-enabled IoT infrastructures. The implemented scenarios include MQTT flooding, Sliver Command and Control (C2) beaconing traffic, Denial-of-Service attacks using `hping3`, network scanning using `Nmap`, and brute-force attacks targeting embedded web services. MQTT flooding attacks generate excessive publish operations to overload MQTT brokers and communication channels, affecting service availability and telemetry processing [HiveMQ 2023]. Sliver C2 beaconing traffic simulates adversarial command-and-control communication patterns commonly used in advanced intrusion scenarios and malware operations [Bishop Fox 2024]. Denial-of-Service attacks generated with `hping3` aim to exhaust network or service resources through abnormal packet transmission rates [Sanfilippo 2005]. Network scanning using `Nmap` is employed to identify active hosts, open ports, and exposed services within the IoT infrastructure [Lyon 2009]. Finally, brute-force attacks attempt repeated authentication requests against embedded web interfaces in order to obtain unauthorized access credentials [OWASP Foundation 2024]. Figure 4 illustrates the attack execution workflow adopted in the proposed environment. The generated malicious traffic is captured through PCAP collection and MQTT telemetry extraction pipelines, enabling automated labeled dataset generation for intrusion detection experiments.

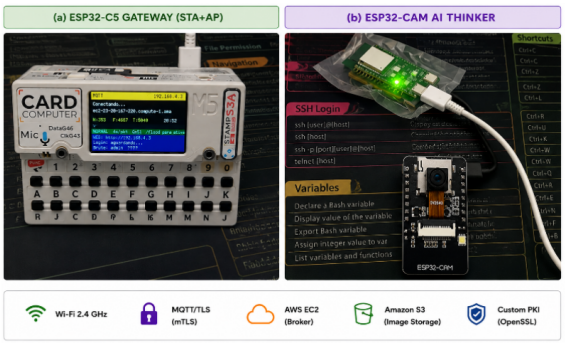

Figure 2. IoT devices used in the experimental environment.

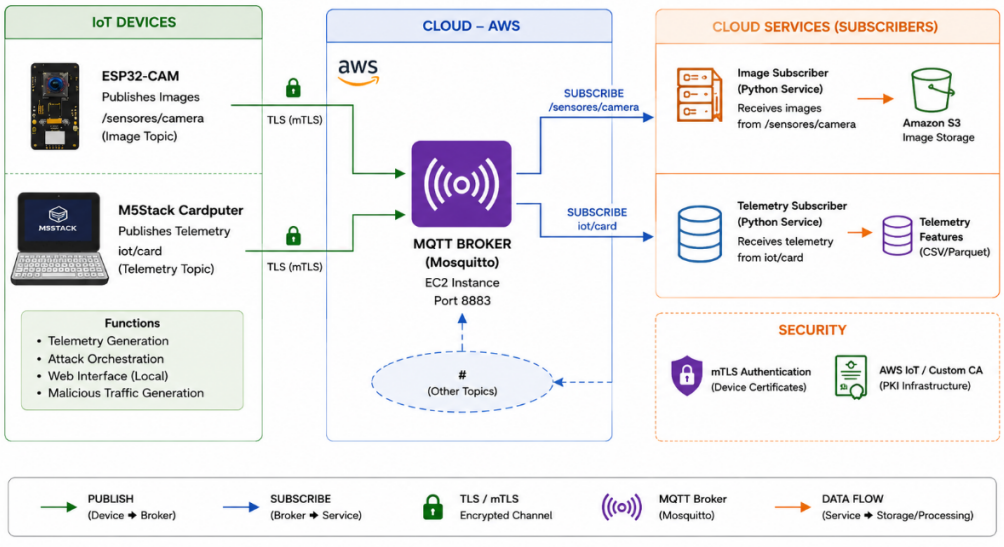

Figure 3. MQTT/TLS communication flow between IoT devices and cloud-based subscriber services.

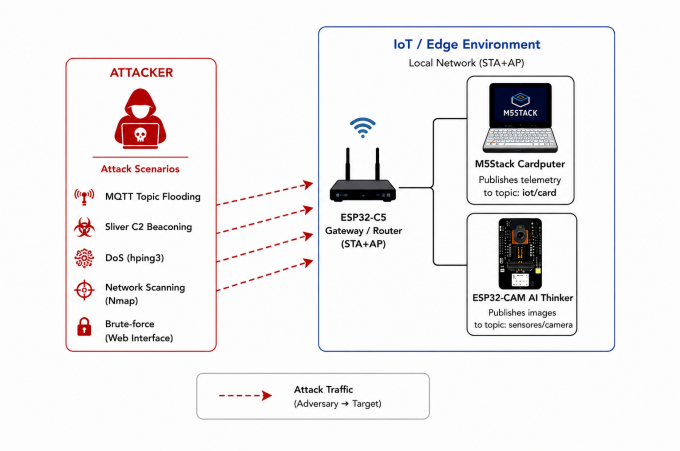

Figure 4. Attack scenarios executed against the IoT edge environment.

4. Dataset Generation

This section describes the dataset generation pipeline adopted in the proposed IoT cybersecurity environment. The workflow integrates PCAP traffic collection, MQTT telemetry extraction, automated labeling, and feature engineering to generate multimodal datasets for supervised ML experiments. Figure 5 illustrates the overall data processing workflow. Network traffic is collected directly from the ESP32-C5 gateway operating in STA+AP mode, enabling monitoring of communication flows between IoT devices and cloud services. Traffic capture is performed using `tcpdump`, generating PCAP files containing both legitimate and malicious traffic, including MQTT/TLS communication, DNS requests, flooding attacks, beaconing traffic, and network scanning activity. The captured traffic is later processed through Python-based feature extraction scripts to generate structured datasets for ML analysis. The environment also captures MQTT telemetry generated by the IoT devices.

The ESP32-CAM publishes image events, while the M5Stack Cardputer transmits telemetry data through MQTT topics. Cloud-based subscriber services consume MQTT messages and extract communication-level features such as topic activity, payload size, publish frequency, timestamp intervals, QoS levels, and publish/subscribe behavior patterns. The extracted telemetry is exported to CSV and Parquet files for integration with PCAP-derived features. Dataset labeling is performed automatically according to the attack orchestration workflow executed during the experiments. Timestamp synchronization between attack execution and traffic collection enables accurate association between captured traffic and attack events, reducing manual intervention during dataset construction.

4.1. Feature Engineering

Feature engineering combines PCAP-derived network traffic attributes and MQTT telemetry features into a unified representation for the ML experiments. The extracted network-level attributes include packet count, flow duration, source and destination ports, protocol distribution, TCP flags, and connection frequency. The MQTT-level attributes include topic activity, payload size, publish frequency, QoS level, and publish/subscribe behavioral patterns.

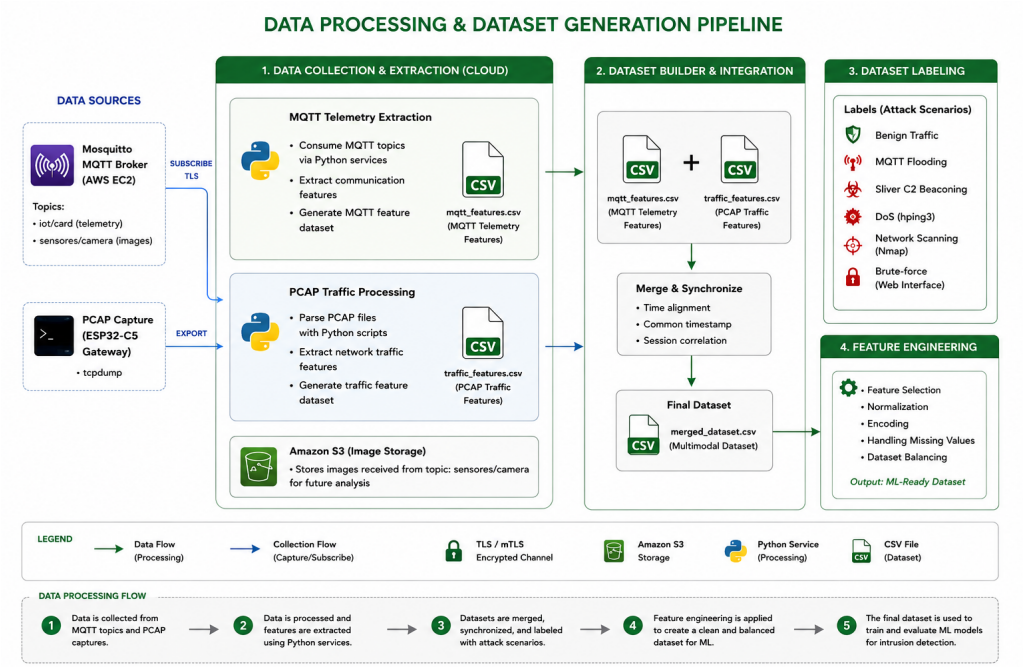

Figure 5. Data processing and dataset generation pipeline integrating MQTT telemetry extraction, PCAP traffic processing, dataset labeling, and feature engineering for Machine Learning-based intrusion detection.

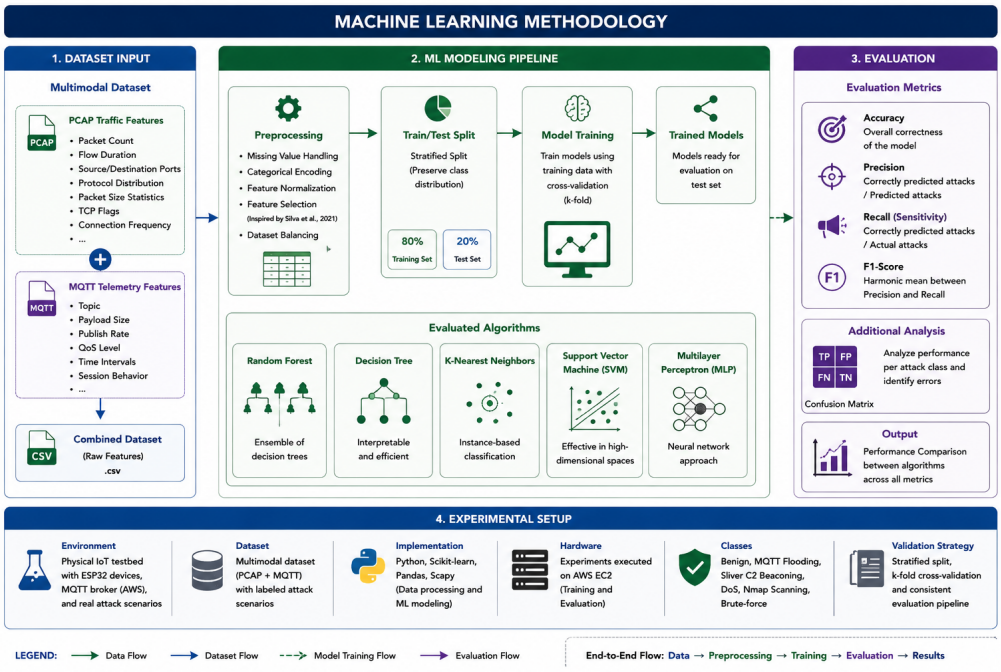

Figure 6. Machine Learning methodology including dataset preprocessing, supervised model training, evaluated algorithms, and performance evaluation metrics.

Feature selection techniques inspired by [Silva et al. 2021] were applied to identify the most relevant attributes and reduce dataset dimensionality, which is particularly important in resource-constrained IoT environments. Table 2 presents examples of the features extracted from the PCAP and MQTT data sources.

Table 2. Examples of extracted dataset features.

Feature	Type	Source
Packet Count	Numerical	PCAP
Flow Duration	Numerical	PCAP
MQTT Topic	Categorical	MQTT
Payload Size	Numerical	MQTT
Publish Rate	Numerical	MQTT
TCP Flags	Numerical	PCAP
QoS Level	Categorical	MQTT
Connection Frequency	Numerical	PCAP

5. Machine Learning Methodology

This section describes the ML methodology adopted to evaluate intrusion detection performance using the generated multimodal dataset composed of PCAP traffic and MQTT telemetry. Figure 6 summarizes the preprocessing, training, and evaluation workflow adopted in the experiments.

5.1. Evaluated Algorithms

Five supervised ML algorithms were evaluated: Random Forest, Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). These models were selected due to their widespread adoption in intrusion detection research and applicability to cybersecurity datasets. Random Forest was evaluated because of its robustness against noisy and imbalanced datasets, while Decision Tree was selected due to its interpretability and low computational cost for embedded environments. KNN was included as a distance-based classifier capable of identifying traffic similarity patterns. SVM was evaluated for its capability to separate complex classes in high-dimensional spaces, whereas MLP was selected to analyze the effectiveness of neural-network-based classification for IoT intrusion detection. The experiments focused on multiclass classification considering both legitimate traffic and malicious activity generated during the attack orchestration process.

5.2. Experimental Setup

The experiments were implemented in Python using the Scikit-learn framework, while data processing and feature extraction were performed using Pandas and Scapy. The ML pipeline includes missing-value handling, categorical encoding, feature normalization, dataset balancing, and feature selection, which was described in Section 4.1. The resulting dataset was divided into training and testing subsets using a stratified split strategy to preserve the distribution of the evaluated classes. These classes included benign traffic, MQTT flooding, Sliver C2 beaconing, Denial-of-Service, network scanning, and brute-force attacks. All classifiers were trained and evaluated using the same preprocessing, partitioning, and validation procedures to ensure a fair comparison among the models.

5.3. Evaluation Metrics

Model performance was evaluated using standard classification metrics commonly adopted in intrusion detection research. Accuracy was used to measure the overall proportion of correctly classified samples, while precision and recall were employed to evaluate malicious traffic detection capability. The F1-score was adopted to balance precision and recall performance. Confusion matrices were also generated to analyze the classification behavior across different attack categories, enabling the identification of false positives, false negatives, and low-frequency attack detection limitations. Particular attention was given to stealthier scenarios such as Sliver C2 beaconing traffic, which presents communication patterns similar to legitimate MQTT activity.

6. Results and Discussion

This section presents the experimental results obtained from the ML models evaluated using the multimodal dataset generated in the proposed IoT cybersecurity environment.

6.1. Classification Results

The evaluation considered both PCAP-derived traffic features and MQTT telemetry attributes. Figure 7 summarizes the overall performance of the evaluated models. The results demonstrate that lightweight supervised approaches were capable of detecting high-volume attacks such as MQTT flooding and network scanning with high accuracy. However, low-frequency and stealthier behaviors, particularly Sliver C2 beaconing traffic, presented additional classification challenges.

Model	Accuracy (%)	Precision	Recall	F1-Score
Random Forest	96.0	0.69	0.69	0.69
Decision Tree	95.9	0.70	0.70	0.70
KNN	95.6	0.62	0.62	0.62
MLP	95.2	0.53	0.53	0.53
SVM	95.0	0.46	0.46	0.46

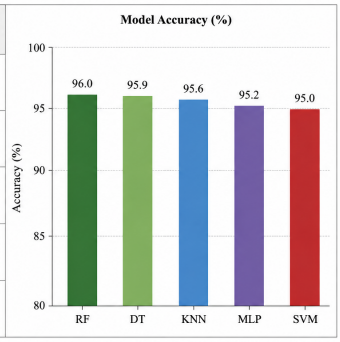

Figure 7. Overall comparison of evaluated Machine Learning algorithms.

The Random Forest (RF) model achieved the best accuracy of approximately 96%, demonstrating robustness against class imbalance and heterogeneous attack behaviors, successfully identifying both volumetric and low-frequency attack patterns. Decision Tree (DT) achieved similar accuracy results while presenting slightly higher overfitting behavior for minority classes, maintaining interpretability and lower computational complexity. SVM presented the lowest F1-score among the evaluated approaches, particularly struggling to distinguish benign traffic from flooding scenarios. The SVM lowest performance is due to sensitivity to imbalanced classes and distortion caused by flooding and brute-force traffic distributions.

6.2. Confusion Matrix and Feature Analysis

Figures 8 and 9 present the confusion matrices obtained from PCAP-derived features and MQTT telemetry, respectively. The PCAP-based experiments demonstrated strong detection performance for brute-force and scanning attacks, particularly when using Random Forest and Decision Tree. Low-volume Sliver C2 beaconing traffic was more difficult to distinguish because its low communication frequency and traffic characteristics were similar to legitimate MQTT activity. In contrast, the MQTT telemetry experiments achieved near-perfect classification performance for MQTT flooding across the evaluated models. MQTT-level attributes such as publish frequency, payload sequence, and inter-arrival time were therefore especially effective for identifying volumetric activity.

These results indicate that network-level and application-level features provide complementary information. Network attributes contributed to the detection of scanning and brute-force behavior, whereas MQTT behavioral attributes were particularly relevant for identifying publish-flooding patterns.

6.3. Discussion

The experimental results demonstrate the feasibility of using a low-cost ESP32-based infrastructure for realistic IoT cybersecurity experimentation and multimodal dataset generation. The proposed architecture successfully integrated physical IoT devices, MQTT communication, PCAP traffic capture, application-level telemetry extraction, cloud-based processing, and supervised machine learning evaluation into a unified experimental platform.

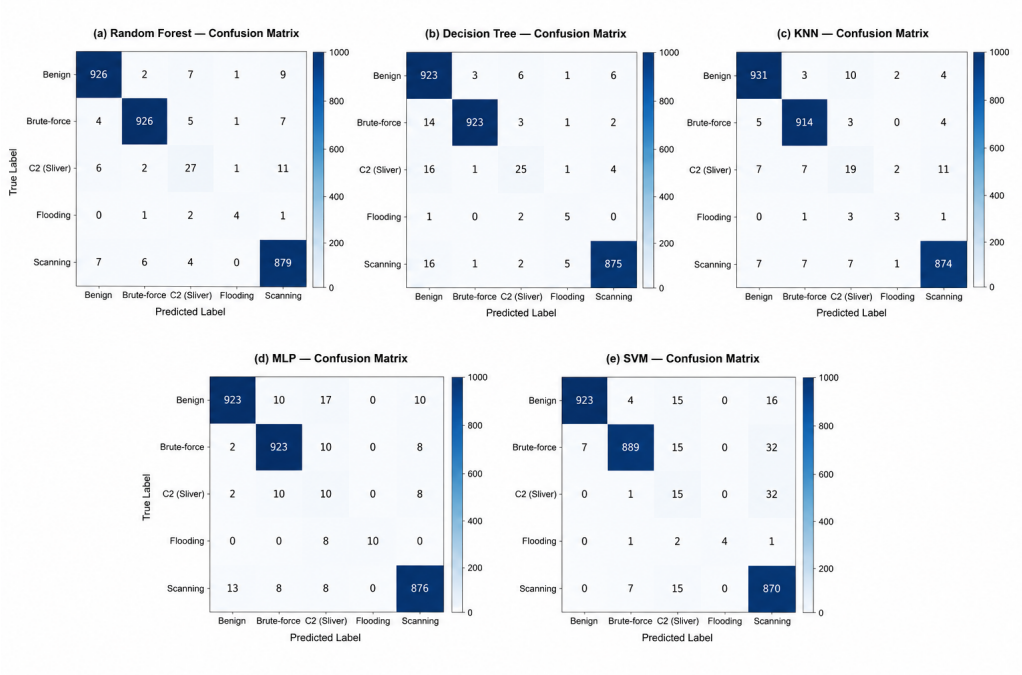

Figure 8. Confusion matrices and feature importance analysis for PCAP-derived network traffic features.

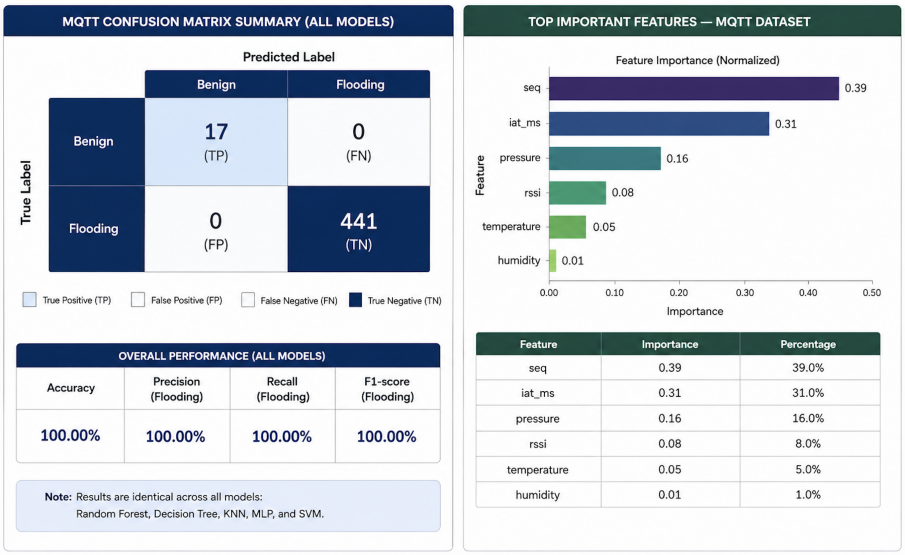

Figure 9. Confusion matrices and feature importance analysis for MQTT telemetry features.

The results indicate that traditional supervised machine learning algorithms remain effective for detecting attacks that produce substantial changes in network behavior. Random Forest achieved the best overall trade-off among detection capability, robustness, and multiclass classification performance. Volumetric attacks were more accurately identified because they generated distinguishable variations in packet volume, transmission frequency, connection behavior, protocol activity, and traffic distribution.

In contrast, Sliver C2 beaconing was more difficult to distinguish from legitimate MQTT communication. Its low transmission frequency and periodic communication pattern produced less evident variations in the extracted features than those observed in volumetric attacks. Because beaconing traffic may resemble routine device communication, conventional flow-based and statistical features provide limited evidence for separating malicious and legitimate behavior. This result explains the comparatively lower detection performance observed for stealthier attack classes.

The combination of PCAP-derived features and MQTT telemetry provided complementary perspectives of the monitored environment. PCAP analysis represented network and transport-level characteristics, including packet counts, protocol distribution, connection frequency, ports, TCP flags, and flow statistics. MQTT telemetry described application-level behavior through publication activity, message intervals, topics, and payload-related characteristics. This multimodal representation enabled attacks to be examined across different layers of the IoT communication stack rather than exclusively from network-flow information.

The experimental environment also demonstrated reproducibility and modularity. Its components were separated into device operation, MQTT communication, traffic capture, feature extraction, dataset construction, and model evaluation. This organization allowed attack executions and data-processing procedures to be repeated under controlled conditions while preserving the correspondence between physical device behavior, network traffic, and MQTT telemetry.

Overall, the findings show that the proposed testbed provides a viable platform for generating realistic cybersecurity datasets and evaluating intrusion detection models in resource-constrained IoT environments. The results also demonstrate that detection effectiveness depends on the observable characteristics of each attack: high-volume attacks produce clearer statistical deviations, whereas low-frequency and behaviorally similar threats remain more difficult to identify.

6.4. Distributed Intrusion Detection

The proposed testbed adopts a distributed monitoring structure in which data acquisition occurs at different points of the IoT environment. ESP32-based devices generate legitimate and malicious communication, the MQTT broker manages application-level message exchange, network capture components collect PCAP traffic, and subscriber services extract and store MQTT telemetry. These components operate as separate data sources while preserving their temporal and operational relationships.

This organization distributes evidence collection across the device, network, application, and cloud layers. Network captures provide visibility into packet and flow behavior, whereas MQTT subscribers observe protocol-specific interactions. The cloud environment centralizes the processing of these data sources, enabling their conversion into structured datasets and their subsequent evaluation using supervised machine learning models.

Therefore, the implemented environment should be characterized as a distributed data acquisition and monitoring architecture with centralized model training and evaluation. The physical and logical separation of the monitoring components increases visibility across the IoT communication chain and reduces dependence on a single source of security evidence. However, the current implementation does not perform distributed model training, collaborative inference, or autonomous detection directly on the ESP32 devices.

Limitations. The testbed contains a limited number of physical IoT devices, network configurations, and attack scenarios, which restricts the direct generalization of the results to larger and more heterogeneous IoT or CPS deployments. The experiments were conducted using offline datasets and centralized supervised learning, without evaluating real-time inference, processing latency, memory consumption, energy usage, or adaptation to changes in device and network behavior.

Model performance also depends on the representativeness and class distribution of the collected samples. Although the multimodal dataset incorporates both network-level and MQTT-level information, the evaluated features were less effective in characterizing low-frequency attacks whose communication patterns resemble legitimate device activity. In addition, the current evaluation does not measure the individual contribution of each data modality through an ablation analysis, limiting conclusions about the specific influence of PCAP and MQTT features on classification performance.

7. Conclusion and Future Work

This work presented a low-cost and reproducible IoT cybersecurity testbed for realistic attack execution, multimodal dataset generation, and machine learning-based intrusion

detection. The environment integrates ESP32-based devices, MQTT communication protected by TLS, PCAP traffic capture, MQTT telemetry extraction, cloud-based processing, and supervised machine learning evaluation.

The platform supported the execution of MQTT flooding, Sliver C2 beaconing, Denial-of-Service, network scanning, and brute-force attacks. The resulting datasets enabled complementary evaluations using network-level and application-level attributes. In the PCAP-based multiclass experiment, Random Forest achieved the highest accuracy, whereas Decision Tree obtained a slightly higher F1-score. Volumetric and scanning-related activities produced clearer behavioral deviations, while low-frequency C2 beaconing remained more difficult to distinguish from legitimate communication. In the MQTT-based binary experiment, telemetry attributes effectively separated benign and flooding traffic within the reported evaluation partition.

Overall, the results demonstrate the feasibility of the proposed testbed as a modular platform for controlled attack execution, synchronized data acquisition, automated dataset construction, and comparative intrusion detection experiments. The current implementation distributes monitoring and data collection across device, network, application, and cloud components, while model training and evaluation remain centralized.

Future work will extend the platform toward collaborative intrusion detection and Federated Learning across multiple heterogeneous IoT edge nodes. Each node will locally collect PCAP and MQTT telemetry and train an intrusion detection model without transmitting raw traffic data. Only model parameters or updates will be shared with a cloud-based coordination server. The existing AWS EC2 infrastructure may initially support model aggregation using Federated Averaging as a baseline.

The distributed evaluation will consider non-independent and non-identically distributed data, intermittent client participation, communication failures, privacy preservation, convergence, and communication overhead. Robust aggregation, adaptive trust, and reputation mechanisms will also be evaluated to reduce the influence of unreliable or malicious participants, including clients performing poisoning or model-manipulation attacks. These mechanisms may be investigated in alignment with the adaptive cybersecurity principles of the CSAI-4-CPS framework.

Additional experiments will investigate whether knowledge shared across different operational environments improves the detection of low-frequency and stealthy attacks such as Sliver C2 beaconing. Temporal and sequential analysis will incorporate communication periodicity, inter-arrival times, sliding windows, and behavioral sequences that are not adequately represented by aggregated flow statistics.

TinyML-based inference will also be evaluated directly on ESP32 devices to measure the feasibility of local and real-time intrusion detection. This evaluation will consider processing latency, memory consumption, energy usage, model size, and resilience during temporary loss of cloud connectivity. Online and adaptive learning approaches will be investigated to address changes in device behavior, network conditions, and attack patterns.

Finally, it will be compared to models trained with PCAP-derived features, MQTT telemetry attributes, and their combined representation to quantify the individual contributions of each data source to intrusion detection performance. Privacy-

preserving telemetry-sharing mechanisms and resilient coordination approaches, including blockchain-assisted validation of collaborative model updates, may also be investigated to strengthen distributed cybersecurity architectures for IoT and CPS.

Acknowledgements

This study was financed in part by CAPES - Finance Code 001. The authors thank SENAI São Caetano do Sul for the physical infrastructure for the academic support provided during this research.

References

Alatram, A., Sikos, L. F., Johnstone, M., Szewczyk, P., and Kang, J. J. (2023). Dos/ddos-mqtt-iot: A dataset for evaluating intrusions in iot networks using the mqtt protocol. *Computer Networks*, 231:109809.

Alshamrani, A. and Anwar, A. (2022). Federated learning-based intrusion detection for mqtt-based iot networks. *Sensors*, 22(18):6854.

Aqachtoul, A., Karam, K., Elamrani, A., Najib, M., Rafalia, N., and Bakhouya, M. (2025). Mqtteeb-d: A real-world iot cybersecurity dataset for ai-powered threat detection in mqtt networks. *Data in Brief*, 62:111897.

Aveleira-Mata, J., Alaiz-Moretón, H., Bayón-Guitérrez, M., García-Ordás, M. T., Prieto-Fernandez, N., and García-Rodríguez, I. (2025). Mqtt_uad: Mqtt under attack dataset. a public dataset for the detection of attacks in iot networks using mqtt protocol. *Data in Brief*, 63:112167.

Bishop Fox (2024). Sliver c2 framework. Available at: `https://github.com/BishopFox/sliver`. Accessed: 2026-05-10.

Fagan, M., Megas, K. N., Scarfone, K., and Smith, M. (2020). Foundational cybersecurity activities for iot device manufacturers. Technical Report NISTIR 8259, National Institute of Standards and Technology.

Farag, W., Wu, X.-W., Ezekiel, S., Rado, D., and Lassinger, J. (2025). Development and evaluation of a novel iot testbed for enhancing security with machine learning-based threat detection. *Sensors*, 25(18):5870.

Ferrag, M. A. et al. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50:102419.

Freitas, A. et al. (2024). Mqtt-vet: Exploring mqtt protocol vulnerabilities. In *Anais Estendidos do XIII Latin-American Symposium on Dependable Computing*, page 111–113, Porto Alegre, RS, Brasil. SBC.

HiveMQ (2023). Mqtt security fundamentals. Available at: `https://www.hivemq.com/blog/mqtt-security-fundamentals/`. Accessed: 2026-05-10.

International Electrotechnical Commission (2021). Iec 62443: Security for industrial automation and control systems.

Lazzarini, R. et al. (2023). Federated learning for iot intrusion detection. *Internet of Things*, 22:100775.

Lyon, G. F. (2009). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.

Meidan, Y. et al. (2018). N-baiot: Network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22.

Nguyen, D. et al. (2021). Fediot: Federated learning for internet of things security. *IEEE Internet of Things Journal*, 8(16):12786–12795.

OASIS (2019). Mqtt version 5.0. Available at: `https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html`. Accessed: 2026-05-10.

Omotosho, A. et al. (2023). Ids-ma: Intrusion detection system for iot mqtt attacks using centralized and federated learning. *Array*, 18:100305.

OWASP Foundation (2024). Authentication cheat sheet. Available at: `https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html`. Accessed: 2026-05-10.

Rezaei, S. and Liu, X. (2019). Deep learning for encrypted traffic classification: An overview. *IEEE Communications Magazine*, 57(5):76–81.

Sanfilippo, S. (2005). hping3: Active network security tool. Available at: `https://www.kali.org/tools/hping3/`. Accessed: 2026-05-10.

Silva, H. (2022). Csai-4-cps: A cyber security characterization model based on artificial intelligence for cyber physical systems. In *2022 IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 47–52. IEEE.

Silva, H. et al. (2021). Feature selection: Supporting the mining process on cyber-physical systems result datasets. In *Workshop de Trabalhos de Iniciação Científica e Graduação (WTF)*.

Silva, H. and Moraes, R. (2024). Privacy-preserving iot intrusion detection: Challenges and solutions in implementing the csai-4-cps model. *Computer Science & Information Technology (CS & IT)*, 14(7):95–108.

Verma, S. and Patel, A. (2024). Efficient network traffic feature sets for iot intrusion detection. *arXiv preprint arXiv:2406.08042*.