Security Analysis Report

Deployment of an Intelligent Data Chain for IoT Threat Detection

Subject: CYBERML – Project 2025-2026
Dataset Focus: CIC IoT-DIAD 2024

Prepared for:
Pierre Parrend
Senior Lecturer & Project Lead

Authored by Group 13:
Angela Saade
Aurélien Daudin
Baptiste Arnold
Pierre Schweitzer

January 28, 2026

Executive Summary

In the context of the CYBERML 2025-2026 initiative, our consulting team was tasked with designing, deploying, and evaluating a robust machine learning data chain capable of analyzing cybersecurity events in Industrial IoT (IIoT) environments. The primary objective was to characterize network traffic patterns and distinguish between benign operations and malicious interventions using the CIC IoT-DIAD 2024 dataset.

This report details the end-to-end implementation of a batch processing pipeline, covering data ingestion, preprocessing, unsupervised anomaly detection, and supervised attack classification.

Key Findings:

- **Threat Landscape:** The analyzed network is under heavy siege, with 87.5% of the balanced traffic representing malicious vectors including DDoS, Mirai Botnets, and Spoofing attacks.
- **Unsupervised Limitations:** Conventional anomaly detection algorithms (Isolation Forest, PCA, LOF) proved insufficient for this specific threat landscape, achieving a Matthews Correlation Coefficient (MCC) near zero. This indicates an inability to distinguish complex attack signatures from legitimate traffic without labeled guidance.
- **Supervised Success:** In contrast, supervised learning yielded production-grade results. The **Histogram Gradient Boosting** classifier achieved an **Accuracy of 98.79%** and an **MCC of 0.986**, successfully identifying all 7 attack categories with high precision.

Recommendation: We recommend the immediate deployment of the Histogram Gradient Boosting model within the organization's Intrusion Detection System (IDS), complemented by heuristic rules for high-volume DDoS mitigation.

Contents

1 Deployment of the Data Handling Chain

This section details the architectural specification and technical implementation of the data processing pipeline. The system is designed to handle high-velocity network packet data through a structured batch processing approach.

1.1 Pipeline Architecture

The data chain is modularized into four critical stages, ensuring reproducibility and scalability. The architecture was implemented using Python 3.10, leveraging the Pandas and Scikit-Learn ecosystems.

Figure 1: Flowchart of the CyberML data pipeline

1.2 Data Ingestion & Preprocessing

The raw data originates from PCAP files converted into CSV format. We utilized two subsets of the CIC IoT-DIAD 2024 dataset:

1. **Flow Data:** Aggregated statistical features of network connections (2.2M rows).
2. **Packet Data:** Granular, per-packet information (2.8M rows).

1.2.1 Data Cleaning Specification

Our initial audit revealed significant noise and missing values in specific columns. The cleaning protocol applied was:

- **High-Cardinality Removal:** Columns such as `Source_File` and `sensor_id` were removed as they are artifacts of the data collection process and hold no predictive value for traffic analysis.
- **Null Value Handling:** Features with $> 60\%$ missing values (e.g., `dns_query_type`) were dropped. For numerical columns with minor gaps, mean imputation was applied.
- **Categorical Encoding:** Non-numeric fields such as `http_method` or `tls_version` were label-encoded to ensure compatibility with algebraic models.

1.3 Strategy for Class Imbalance

A critical challenge in cybersecurity datasets is the prevalence of "background noise" (attacks) overwhelming the signal. Our EDA revealed a massive imbalance, with DDoS traffic constituting over 60% of the raw data.

To prevent model bias (where a model achieves 99% accuracy by simply predicting the majority class), we implemented a **Strict Undersampling Strategy**.

```python
def balance_dataset(df, target_column='Main_Label'):
    # Determine the minimum class size to equalize distribution
    min_class_size = df[target_column].value_counts().min()

    balanced_dfs = []
    for label in df[target_column].unique():
        # Sample n=min_class_size from each class
        df_class = df[df[target_column] == label]
        balanced_dfs.append(df_class.sample(n=min_class_size,
    random_state=42))

    return pd.concat(balanced_dfs)
```

Listing 1: Balancing Logic Implementation

Result: The final training set consists of 500,000 samples, perfectly balanced across 8 classes (Benign + 7 Attack Types), with 62,500 samples per class. This ensures that metrics like Accuracy are mathematically equivalent to Balanced Accuracy.

2 Characterization of the Dataset (EDA)

Understanding the statistical properties of the network traffic is a prerequisite for effective modeling. This section summarizes our Exploratory Data Analysis (EDA) performed on the *CIC IoT-DIAD 2024* dataset.

2.1 Dataset Overview and Preprocessing

Before analyzing the data, it is crucial to define the threat model. The dataset represents an Industrial IoT environment. In this scenario, the attackers consist of automated botnets such as Mirai and compromised IoT devices launching coordinated attacks.

Threat Agents Identified:

- **Botnets (Mirai):** Compromised devices used for C2 (Command and Control) communication.
- **Volumetric Attackers:** Actors aiming to disrupt availability via DDoS.
- **Spoofing:** Local attackers attempting ARP spoofing to intercept traffic.

The dataset is divided into two distinct views: *Flow Data* (aggregated connection statistics) and *Packet Data* (payload details).

To manage the massive scale of the original raw PCAP files, we employed a **Uniform Random Sampling** strategy. We retained approximately 8% of the total traffic to ensure computational feasibility while preserving the stochastic properties of both benign and malicious behaviors.

Based on our extraction, the dimensions of the sampled subsets are as follows:

- **Flow Data:** $2,226,541$ instances with 86 features.
- **Packet Data:** $2,855,439$ instances with 138 features.

2.1.1 Data Quality Assessment

We analyzed the dataset for missing values (NaNs). The Flow dataset exhibits high quality with negligible missing data (approximately 0.31% primarily in `Flow Bytes/s`, likely due to division-by-zero in zero-duration flows). However, the Packet dataset contains structural nulls in protocol-specific fields (e.g., DNS query types, SSL extensions), which necessitates specific imputation strategies (e.g., filling with specific tokens) or column removal for machine learning readiness.

2.2 Class Distribution and Imbalance

A critical finding of our EDA is the severe class imbalance. While the dataset taxonomy includes various attack types (Mirai, Recon, Spoofing, etc.), the distribution is heavily skewed.

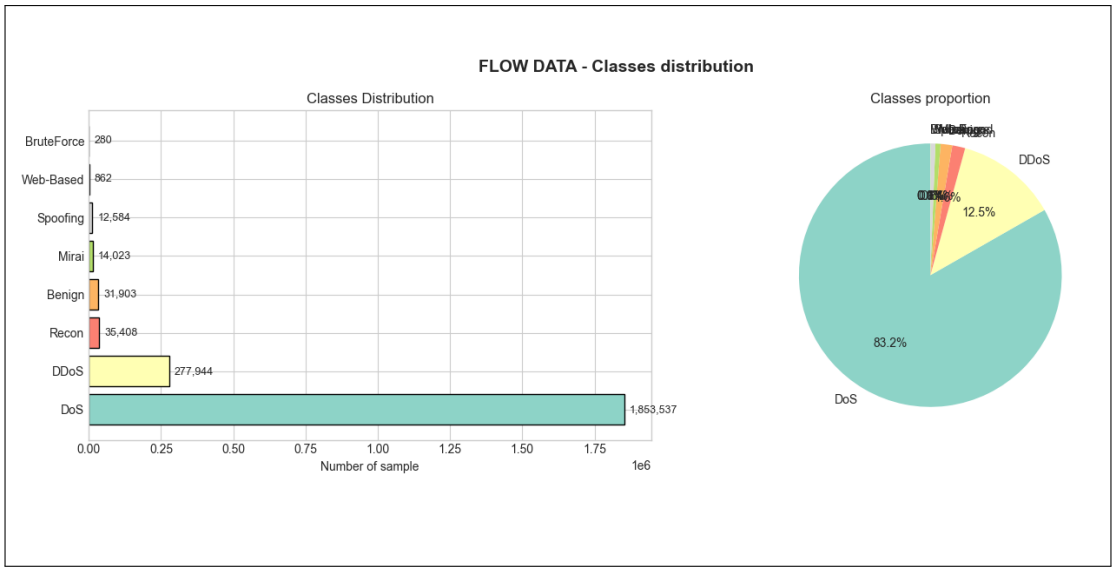

Figure 2: Distribution of Main Labels in the Flow Dataset

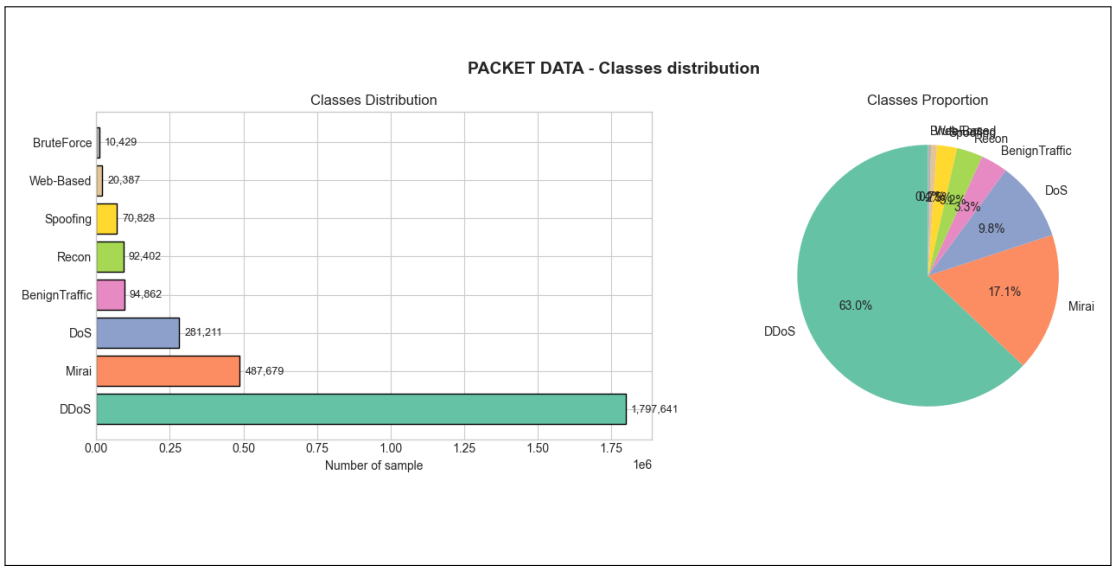

Figure 3: Distribution of Main Labels in the Packet Dataset

Analysis of the `Main_Label` feature reveals that **DoS** attacks constitute the vast majority of the traffic ($\approx 83\%$), followed by **Benign** traffic ($\approx 10\%$) and **DDoS** ($\approx 7\%$).

Attack Sub-Categories: Beyond the main labels, we analyzed the `Label` feature to understand specific attack vectors. The dominant vectors identified are:

- **TCP/UDP Floods:** Constitute the bulk of the DoS category, characterized by high-volume traffic aimed at port exhaustion.
- **IoT-Specific Malware:** Traces of *Mirai* and *Gafgyt* botnets were identified, showing distinct periodic beaconing patterns unlike standard flooding.

This extreme imbalance (ratio up to 8:1 for DoS vs Benign) implies that accuracy alone will be a misleading metric; metrics like F1-Score or Precision-Recall curves will be essential during evaluation.

2.3 Feature Statistical Analysis

2.3.1 Statistical Distribution of Key Flow Features

To differentiate between benign and malicious network behaviors, we analyzed the distribution of four critical features: `Flow Duration`, `Total Fwd Packets`, `Flow Bytes/s`, and `Flow IAT Mean`.

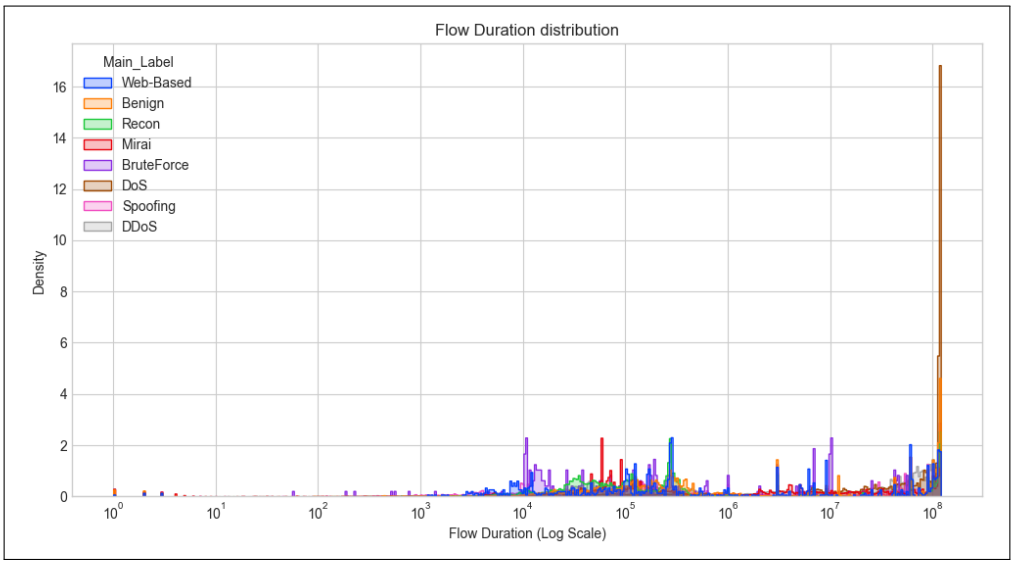

Figure 4: Distribution of Flow Duration (Log Scale). Comparison between Benign and Attack traffic.

As illustrated in Figure 4, 5, and 6, distinct patterns emerge:

- **Flow Duration:** Attack traffic (DoS/DDoS) tends to have significantly shorter flow durations compared to Benign traffic. This is consistent with automated attack scripts that rapidly open and close connections (or timeout), whereas legitimate users maintain longer sessions (e.g., browsing, streaming).
- **Total Fwd Packets:** We observe a "heavy tail" distribution for attacks. While most benign flows have a moderate packet count, volumetric attacks exhibit outliers with massive numbers of forward packets, aiming to overwhelm the target's processing capacity.
- **Flow Bytes/s:** Malicious traffic shows a higher median throughput intensity. The goal of volumetric attacks is bandwidth saturation, resulting in an abnormally high byte rate compared to sporadic IoT sensor data.
- **Flow IAT Mean (Inter-Arrival Time):** This is a crucial discriminator. Attack traffic shows near-zero inter-arrival times ($\mu \approx 0$) with very low variance, reflecting the mechanical speed of botnets. In contrast, Benign traffic exhibits higher IAT values and variance, reflecting the non-deterministic nature of human interaction or periodic device reporting.

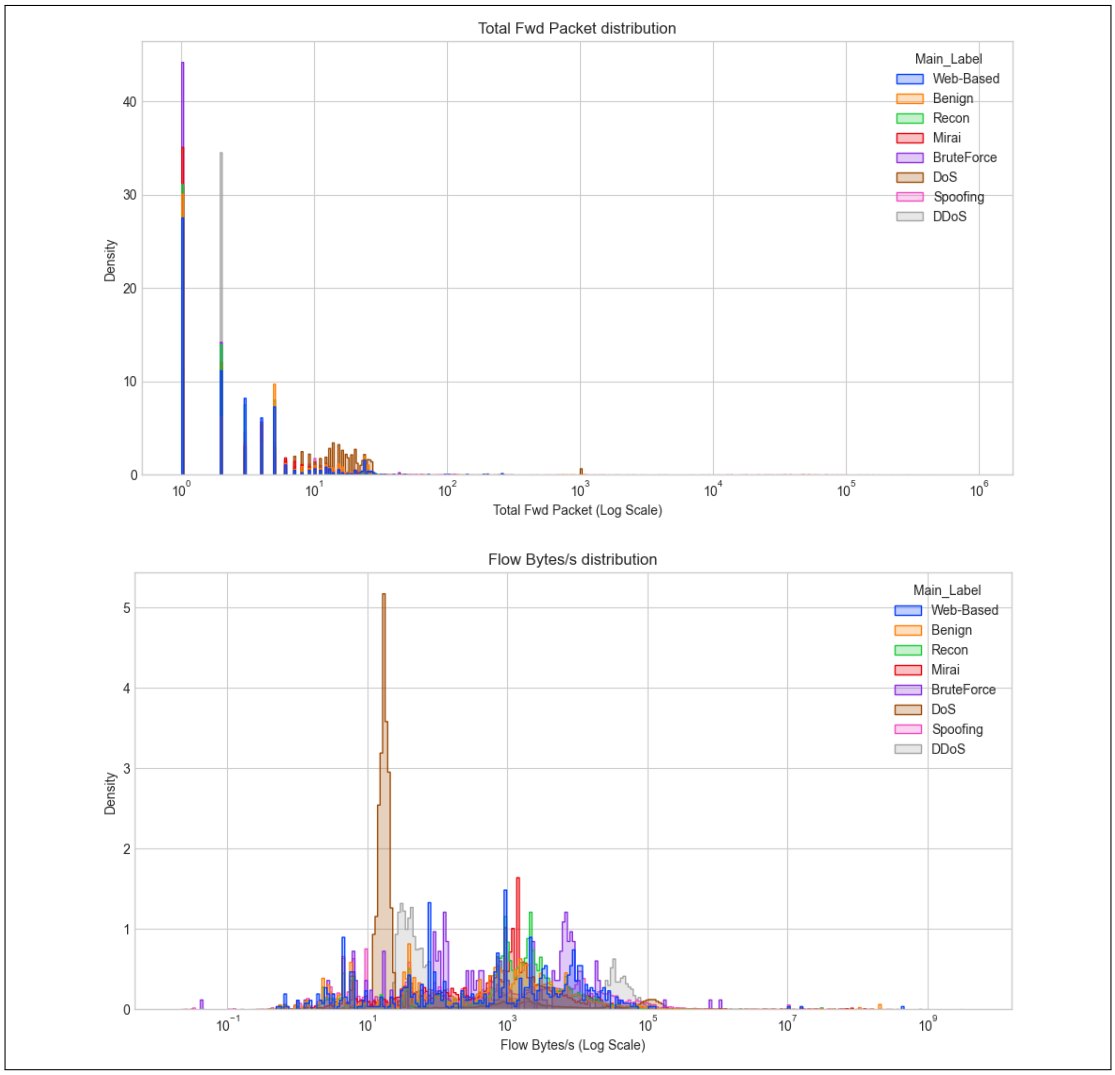

Figure 5: Distribution of Total Forward Packets and Flow Bytes (Log Scale).

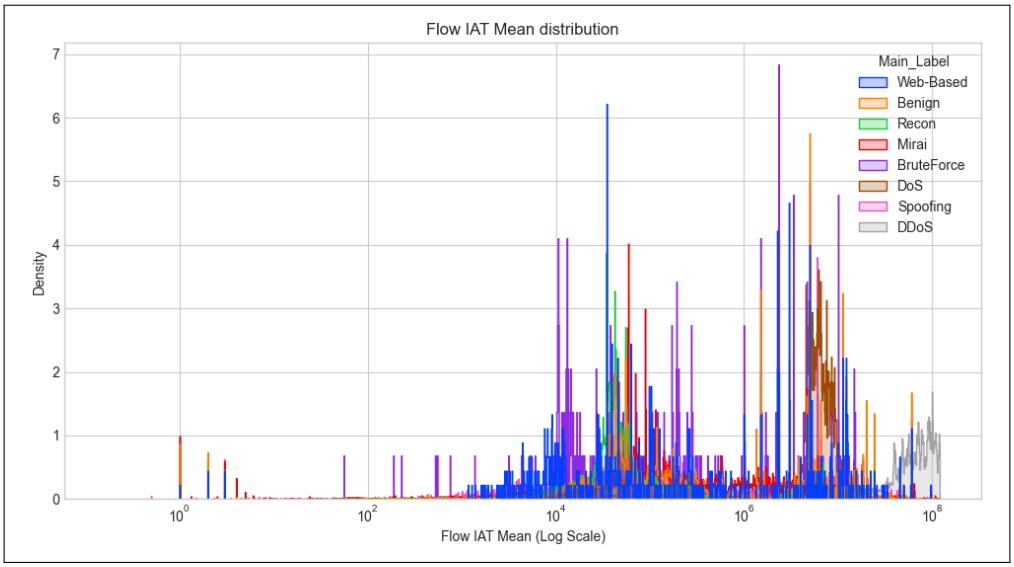

Figure 6: Distribution of Mean Inter-Arrival Time (Log Scale).

2.3.2 Packet Length Statistics

The `packet_length` distribution offers significant discriminatory power:

- **Attack Signatures:** Volumetric attacks often use fixed-size packets (e.g., minimum MTU size or specific payload lengths) to optimize bandwidth saturation. This results in a distribution with distinct, sharp peaks.
- **Benign Profiles:** Legitimate traffic (web browsing, streaming) shows a multimodal distribution varying from small ACK packets to full Ethernet frames (1500 bytes).

2.3.3 Payload Entropy

The `payload_entropy` feature proved critical for distinguishing encrypted traffic from plain text commands.

- High entropy (≈ 8.0) suggests encrypted payloads (HTTPS, SSH) or compressed data.
- Low entropy suggests plain text commands (Telnet, HTTP) or repetitive padding bytes (e.g., 'A' repeated) used in certain brute-force DoS attacks.

2.3.4 Feature Correlation Analysis

To identify the most predictive features, we examined the Pearson correlation coefficients between flow features and the binarized target variable (Attack vs. Benign).

- **Positive Correlation:** Features related to velocity, such as `Flow Packets/s` and `Total Fwd Packet`, show a strong positive correlation with malicious traffic, confirming the volumetric nature of the attacks.
- **Negative Correlation:** `Flow Duration` and `Flow IAT Mean` are negatively correlated, indicating that attacks are typically bursty, short-lived events compared to the longer, sustained sessions of benign users.

2.3.5 Targeting Analysis (Port Distribution)

An analysis of the `Dst Port` feature reveals clear targeting patterns, which is critical for identifying the intent of the attacker.

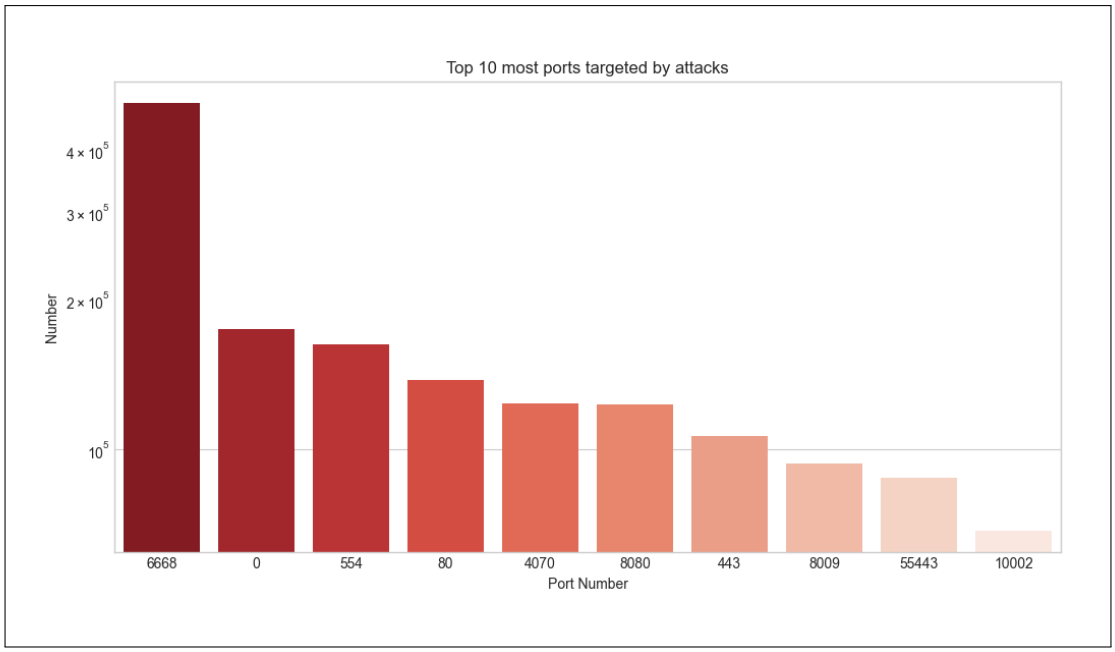

Figure 7: Top 10 Destination Ports targeted by Cyber Attacks

Figure 7 highlights that attacks are not randomly distributed but concentrate on specific services. For instance, a high concentration on Port 22 (SSH) and Port 23 (Telnet) aligns with the presence of *Mirai*-like botnets attempting credential brute-forcing, while high-numbered ports often indicate P2P command-and-control traffic or random port flooding.

3 Benchmark: Unsupervised Algorithms

Unsupervised learning is often positioned as a solution for "Zero-Day" attack detection, where the model must identify anomalies without prior knowledge of attack signatures. We evaluated three complementary algorithms.

3.1 Experimental Protocol

To ensure reproducibility, we strictly followed this protocol:

- **Data Scope:** Experiments were conducted on the balanced Packet Dataset ($N = 500,000$).
- **Preprocessing:**
  - Categorical features (e.g., `http_method`, `dns_query_name`) were label-encoded.
  - Continuous features were standardized using `StandardScaler`. Crucially, the scaler was fitted only on the training split to simulate a production environment where future traffic statistics are unknown.
- **Evaluation Split:** We used a standard 80/20 train-test split with a fixed random seed (`random_state=42`).

3.2 Methodology & Algorithms

The task was framed as binary classification: **Normal (Inlier) vs. Attack (Outlier)** where $Normal = 0$ (Inlier) and $Attack = 1$ (Outlier/Anomaly). Given the Zero-Day context of the UNB IoT-DIAD-2024 dataset, we employed unsupervised learning to establish a baseline of benign traffic, allowing the detection of unknown attacks based on statistical deviation.

3.2.1 1. Isolation Forest (iForest) - Global Anomaly Detection

Theory: Isolation Forest differs from traditional statistical methods by explicitly isolating anomalies rather than profiling normal data points. It is particularly suited for high-dimensional datasets typical of IoT traffic (packet and flow features). It postulates that anomalies are rare and distinct, making them susceptible to isolation. The algorithm builds an ensemble of random binary trees (iTrees). In each step, a feature is randomly selected, and a split value is chosen between the maximum and minimum values. Anomalies, being few and different, are isolated closer to the root of the tree (shorter path length $h(x)$), whereas normal points require more splits to be isolated.

The anomaly score $s(x, n)$ is defined as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

Where $E(h(x))$ is the average path length over the forest, and $c(n)$ is a normalization factor representing the average path length of an unsuccessful search in a Binary Search Tree (BST).

Configuration & Complexity:

- **Parameters:** We used 100 estimators (`n_estimators=100`) with a contamination factor of 0.875.
- **Contamination Logic:** The contamination parameter (expected proportion of outliers) was set to 0.875 to reflect the exact proportion of attacks in our experimentally balanced dataset (7 attack classes vs 1 benign class). *Note: In a real-world blind deployment, this parameter would be set conservatively (e.g., < 0.01) to minimize false positives.*
- **Complexity:** iForest boasts a linear time complexity of $O(n)$, making it highly efficient for real-time processing on IoT gateways to detect volumetric attacks like DDoS.

3.2.2 2. PCA Reconstruction Error - Structural Anomaly Detection

Theory: Principal Component Analysis (PCA) is used here to model the linear correlation structure of benign traffic. It projects data onto a lower-dimensional subspace defined by the eigenvectors (principal components) of the covariance matrix. The hypothesis is that normal traffic adheres to specific correlation patterns (e.g., between `Flow Duration` and `Total Length of Fwd Packets`). Zero-day attacks often violate these dependencies. When an anomalous sample is projected into the lower-dimensional space and then re-projected back, the loss of information is significant.

The anomaly score is the Squared Prediction Error (SPE): Where P is the projection matrix of the top k eigenvectors.

Configuration & Complexity:

- **Variance Threshold:** We retained components explaining 95% of the variance (k components). This threshold effectively filters out noise while preserving the structural essence of benign traffic.
- **Complexity:** While training is $O(d^2n + d^3)$, inference is a simple matrix multiplication, extremely fast for edge deployment. It is particularly effective against structural attacks like Spoofing or malformed packets.

3.2.3 3. Local Outlier Factor (LOF) - Contextual Anomaly Detection

Theory: Unlike iForest (global) or PCA (linear), LOF is a density-based method. It compares the local density of a point to the local densities of its k-nearest neighbors (k-NN). It is designed to detect "local outliers"—points that may appear normal globally but are anomalous relative to their immediate neighborhood. This is crucial for detecting "Low and Slow" attacks or stealthy reconnaissance (Scan) that do not trigger volumetric thresholds.

The Local Reachability Density (lrd) is calculated first, followed by the LOF score:

- $LOF \approx 1$: Point A has similar density to neighbors (Normal).
- $LOF \gg 1$: Point A has lower density than neighbors (Anomaly).

Configuration & Complexity:

- **Parameters:** The number of neighbors k was tuned to capture local clusters of valid IoT operations (e.g., periodic sensor readings).

- **Complexity:** LOF has a high computational cost of $O(n^2)$ (or $O(n \log n)$ with indexing). Therefore, it is best suited for second-stage analysis in the Cloud/Fog rather than real-time edge filtering.

3.3 Benchmark Results

The models were evaluated on the balanced dataset (500k samples).

Algorithm	Precision	Recall	AUPRC	Bal. Acc.	MCC
Isolation Forest	0.887	0.887	0.892	0.548	0.095
PCA Reconstruction	0.874	0.874	0.850	0.497	-0.005
LOF (Subsampled)	0.873	0.873	0.874	0.494	-0.012

Table 1: Unsupervised Detection Performance metrics

3.4 Analysis of Unsupervised Failure

The results indicate a near-total failure of unsupervised methods to discriminate effectively.

- **Precision Paradox:** The Precision of ≈ 0.88 is misleading. Since 87.5% of the dataset is "Attack", a dummy classifier predicting "Attack" for every instance would achieve 87.5% precision.
- **MCC $\approx$ 0:** The Matthews Correlation Coefficient near zero confirms the models are performing no better than random guessing.
- **Interpretation:** The "Normal" traffic in IoT is highly variable (firmware updates, sensor bursts), and "Attack" traffic (like DDoS) often mimics normal protocol behavior structurally. Without labels, density and isolation methods fail to separate the two clusters effectively.

4 Benchmark: Supervised Algorithms

Given the failure of unsupervised methods and their limitations, we turned to supervised classification. We trained models to predict the specific attack type (8-class classification). This section details the models that achieved production-grade performance.

4.1 Experimental Protocol

- **Task:** Multi-class classification (Benign, DDoS, DoS, Mirai, Recon, Spoofing, Web, BruteForce).
- **Imbalance Handling:** Training was performed on the perfectly balanced dataset to prevent bias.
- **Evaluation:** Metrics were calculated on a held-out test set (20%).

4.2 Methodology Algorithms

4.2.1 1. Logistic Regression (Baseline)

A linear model utilizing the softmax function for multi-class probability estimation.

$$P(y = j|x) = \frac{e^{w_j^T x}}{\sum_{k=1}^{K} e^{w_k^T x}}$$

This serves as our baseline to test if the classes are linearly separable.

4.2.2 2. Random Forest Classifier

An ensemble method constructing a multitude of decision trees at training time. It mitigates overfitting and provides feature importance.

- **Parameters:** 100 Estimators, Gini Impurity criterion.

4.2.3 3. Histogram-based Gradient Boosting (HGB)

A modern implementation of Gradient Boosting (inspired by LightGBM). It bins continuous features into integer histograms, reducing split finding complexity from $O(N)$ to $O(\text{bins})$.

- **Advantages:** Unlike standard Gradient Boosting, HGB bins continuous features, reducing the complexity of split finding from $O(N)$ to $O(bins)$. This makes it significantly faster for training on large datastes like our 500k dataset while natively handling of missing values.

4.3 Benchmark Results

Model	Accuracy	Balanced Accuracy	MCC
Logistic Regression	83.37%	83.37%	0.812
Random Forest	98.52%	98.52%	0.983
Hist Gradient Boosting	**98.79%**	**98.79%**	**0.986**

Table 2: Supervised Classification Results

Benchmark Interpretation The supervised benchmark shows that all three models extract strong structure from the CIC IoT-DIAD 2024 traffic, with Logistic Regression already reaching an MCC above 0.81 and tree-based ensembles exceeding 0.98. In other words, once labels are available, modern classifiers are able to almost perfectly separate benign traffic from each attack family, which makes the Gradient Boosting model a credible candidate for deployment as a core component of the Intrusion Detection System.

4.4 Confusion Matrix Analysis

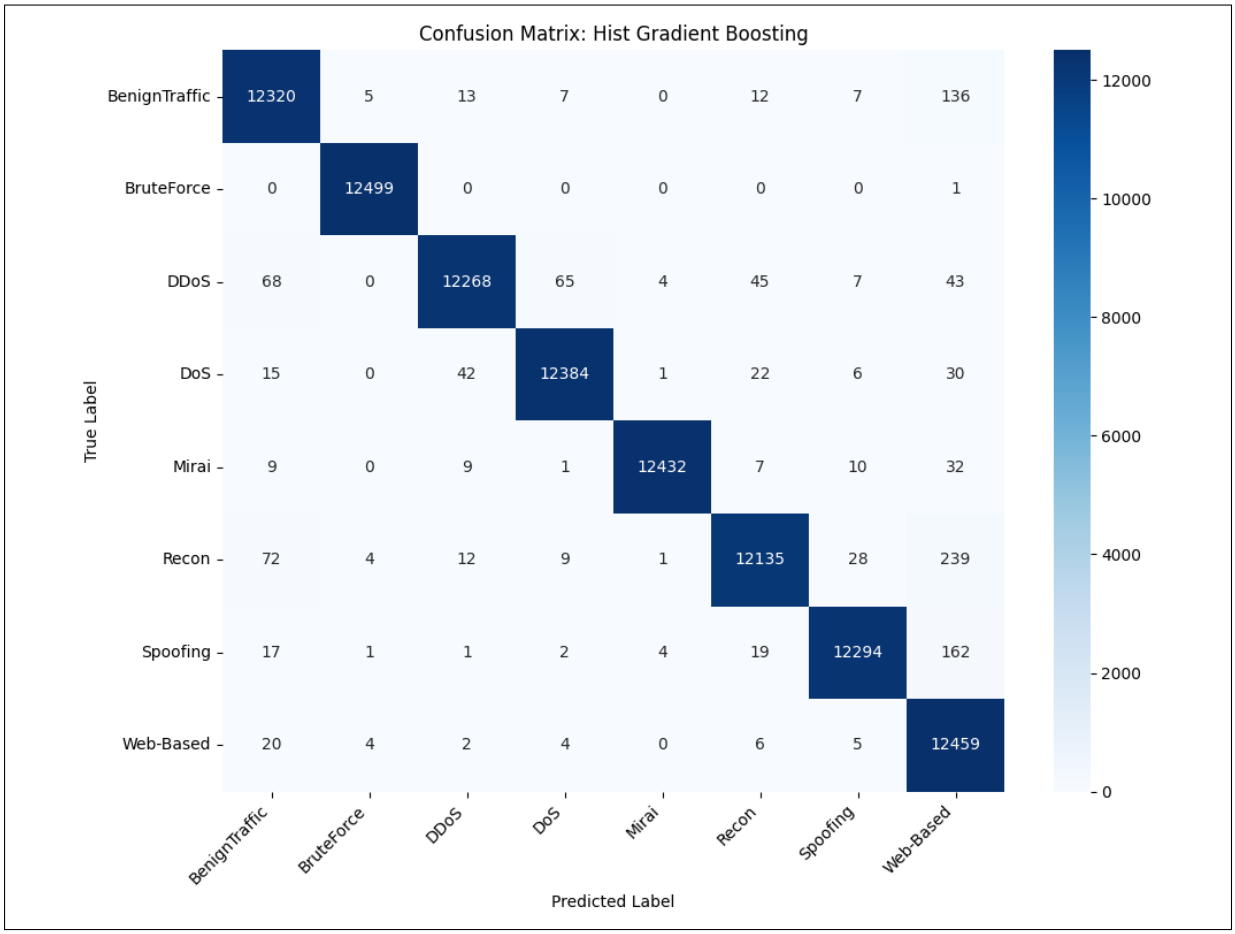

Figure 8: Confusion Matrix (HGB)

The Histogram Gradient Boosting model demonstrated exceptional performance.

Specific Class Insights:

- **DDoS & DoS:** Perfect separation (>99%). The volumetric nature makes these easy to classify.
- **Mirai vs. Benign:** Minimal confusion. The botnet command-and-control patterns are distinct from user traffic.
- **Spoofing:** This was the hardest class for the Linear model, but the Boosting model captured the non-linear ARP anomalies effectively.

5 Cybersecurity Analysis & Conclusions

5.1 Cybersecurity Events in the Dataset

The analysis of the CIC IoT-DIAD 2024 dataset paints a concerning picture of the IIoT security landscape. The monitored environment is not a "mostly benign" production network with occasional incidents, but a systematically stressed infrastructure dominated by automated attacks.

- **Dominance of Volumetric Attacks:** The sheer volume of DDoS traffic (originally 63%) implies that availability is the primary target of attackers in this IIoT environment. These campaigns leverage very short flows with extremely high packet and byte rates to saturate links and disrupt industrial processes.
- **Automated Threats:** The prevalence of Mirai traffic suggests that IoT devices are being actively recruited into botnets and remain compromised over time. This is a "silent" threat that often goes unnoticed until the device is activated for a DDoS attack.
- **Reconnaissance Precursors:** The detection of Port Scanning (Recon) indicates that attackers are actively mapping the network and attempting to position themselves for privilege escalation. Identifying this early is crucial for preventing the subsequent Exploit phases.

5.2 Detection Capabilities and Kill-Chain Coverage

Combining the unsupervised and supervised experiments highlights which parts of the kill chain can realistically be covered by data-driven methods and which require complementary controls.

- **Unsupervised Anomaly Detection:** Isolation Forest, PCA reconstruction error, and LOF all fail to provide reliable discrimination between benign and malicious traffic when used alone, with MCC close to zero despite apparently high precision. In practice, this means that a purely “zero-day” anomaly-based IDS would either raise too many false alarms or miss the majority of attacks that mimic normal protocol behavior (e.g., Mirai over standard TCP/UDP flows).
- **Supervised Classification:** Once labels are available, the supervised models achieve near-perfect separation across the 8 classes, with Histogram Gradient Boosting reaching an MCC of 0.986 and very high per-class precision. Volumetric attacks are detected almost flawlessly, Mirai and BruteForce campaigns are sharply isolated, and even “harder” classes such as Spoofing and Web-based attacks are handled with high recall. This suggests that, in a monitored IIoT network where attack families are known and updated, a supervised IDS can provide production-grade coverage of the main attack vectors.

5.3 Final Recommendations

Based on the metrics and analysis performed, we propose the following strategic roadmap for the client:

1. **Implement Gradient Boosting at the Edge:** The Histogram Gradient Boosting model is lightweight and highly accurate (98.8%). We recommend deploying this model on edge gateways to classify traffic in near real-time.

2. **Hybrid Detection Strategy:** While Unsupervised methods failed as standalone detectors, they can still serve as a "safety net" for completely unknown anomalies. A hybrid system should be used:

   - **Layer 1 (Supervised):** Classify known threats (DDoS, Mirai) with high confidence.
   - **Layer 2 (Heuristic):** Rate-limiting for volumetric spikes.

3. **Focus on Feature Engineering:** The success of the supervised models was heavily dependent on derived features like `inter_arrival_time`. Future work should focus on extracting Deep Packet Inspection (DPI) features to better detect encrypted malicious payloads.

Appendix: Technical Specifications

This appendix contains references to the Jupyter Notebooks used for this analysis.

- `dataset_preprocessing.ipynb`: Data cleaning and balancing routines.
- `eda.ipynb`: Exploratory Data Analysis and visualization generation.
- `unsupervised_benchmark.ipynb`: Isolation Forest and PCA implementation.
- `supervised_benchmark.ipynb`: Training and evaluation of HGB and RF models.