CYBERML Project Report

Classification and Anomaly Detection for Tracking Attacks in Industrial IoT Networks

Dataset:
CIC IIoT Dataset 2025 (DataSense)

Academic Year: 2025-2026

Course: MLCYBER - SCIA

Group Number: 6

Group Name: Cricri Lovers

Group Members:
Gabriel MONTEILLARD
Joric HANTZBERG
Maxime RUFF

January 2026

Contents

Abstract

This report presents a comprehensive analysis of cybersecurity attack detection in Industrial IoT (IIoT) networks using machine learning techniques. We implemented and evaluated six machine learning algorithms—three supervised classification methods (Random Forest, Logistic Regression, Decision Tree) and three unsupervised anomaly detection methods (Isolation Forest, MiniBatch K-Means, Elliptic Envelope)—on the CIC IIoT Dataset 2025. The dataset contains 227,191 samples with 71 features representing network traffic and sensor data from a realistic IIoT testbed. Our results demonstrate that supervised methods achieve high accuracy for known attack types, while unsupervised methods show promising performance for detecting novel attacks. The best performing anomaly detector (Elliptic Envelope) achieved 78.2% balanced accuracy. Additionally, we evaluated model robustness through adversarial attacks, revealing critical vulnerabilities that must be addressed for production deployment. These findings provide valuable insights for deploying machine learning-based intrusion detection systems in industrial environments.

1 Introduction

1.1 Project Objectives

The primary objective of this project is to design, deploy, and evaluate a comprehensive data processing chain for cybersecurity data analysis in Industrial IoT (IIoT) environments. The increasing adoption of IoT devices in industrial settings has created new attack surfaces that require sophisticated detection mechanisms.

Specifically, we aim to:

- Implement and evaluate classification algorithms for attack detection
- Develop anomaly detection systems using unsupervised learning
- Compare the performance of multiple machine learning approaches
- Provide insights into cybersecurity threats in IIoT networks
- **Evaluate model robustness against adversarial attacks (Objective 2)**

1.2 Dataset Selection

We selected the **CIC IIoT Dataset 2025** (DataSense), a real-time sensor-based benchmark dataset for attack analysis in Industrial IoT environments. This dataset represents one of the most comprehensive and recent cybersecurity datasets available for IIoT research, created by the Canadian Institute for Cybersecurity (CIC).

1.3 Methodology Overview

Our approach consists of four main phases:

1. **Data Exploration and Characterization**: Understanding the dataset structure, features, and class distributions
2. **Supervised Classification**: Training and evaluating three complementary classification algorithms
3. **Unsupervised Anomaly Detection**: Implementing three unsupervised methods to detect attacks without labeled data
4. **Adversarial Robustness Testing**: Evaluating model vulnerability to adversarial perturbations

2 Dataset Overview

2.1 Dataset Description

The CIC IIoT Dataset 2025 (DataSense) was created by the Canadian Institute for Cybersecurity and contains data from a sophisticated testbed environment featuring:

- **Testbed Environment**: 40+ interconnected devices including industrial sensors, IoT devices, edge devices, and network equipment
- **Attack Categories**: 50 distinct attack types across 7 categories
- **Time Window**: 1-second aggregated features
- **Data Types**: Both network traffic features and sensor data features

2.2 Dataset Composition

Table 1: Dataset Composition

Category	**Count**
Total Samples	227,191
Benign Samples	136,800 (60.2%)
Attack Samples	90,391 (39.8%)

2.3 Attack Category Distribution

The dataset contains 7 attack categories with the following distribution:

Table 2: Attack Categories

Attack Category	**Count**	**Percentage**
DDoS (Distributed Denial of Service)	41,037	45.4%
DoS (Denial of Service)	40,136	44.4%
Reconnaissance	5,242	5.8%
MITM (Man-in-the-Middle)	2,440	2.7%
Web Attacks	1,085	1.2%
Brute Force	271	0.3%
Malware	180	0.2%

2.4 Feature Space

- **Total Features**: 94 columns
- **Numeric Features**: 71 (used for analysis)
- **Metadata**: 23 columns (device info, timestamps, labels)

Key Network Features:

- Packet statistics (count, size, intervals)
- TCP flags (SYN, ACK, FIN, RST, PSH, URG)
- IP characteristics (length, flags, TTL)
- Protocol information
- Port and MAC address statistics

3 Data Handling Chain

3.1 Architecture Overview

Our data processing pipeline consists of the following stages:

```
Raw CSV Files → Data Loading → Preprocessing → Feature Engineering
    → Model Training → Evaluation → Results Analysis
```

Figure 1: Data Processing Pipeline

3.2 Data Loading

Input Files:

- `attack_samples_1sec.csv`: 90,391 samples with attack labels
- `benign_samples_1sec.csv`: 136,800 samples of normal traffic

Both files contain identical feature structures with 1-second time-windowed aggregations.

3.3 Preprocessing Steps

1. **Data Combination**: Merge attack and benign datasets with appropriate labels
2. **Feature Selection**: Exclude non-numeric and metadata columns
3. **Missing Value Handling**: No missing values detected in the dataset
4. **Infinite Value Handling**: Replace infinite values with NaN, then fill with 0
5. **Feature Scaling**: StandardScaler normalization (zero mean, unit variance)

3.4 Feature Engineering

Excluded Features:

- Device identifiers (device_name, device_mac)
- Timestamp information
- String-based features (IP addresses, MAC addresses, ports, protocols)
- Label columns (except target variable)

Final Feature Set: 71 numeric features

3.5 Train-Test Split

Table 3: Data Split Configuration

Set	**Samples**	**Percentage**
Training Set	159,033	70%
Test Set	68,158	30%

- **Stratification**: Maintained class distribution in both sets
- **Random Seed**: 42 (for reproducibility)

4 Dataset Characterization

4.1 Data Quality Assessment

Completeness:

- Missing values: 0 (100% complete)
- Infinite values: Present in some features, handled during preprocessing

Consistency:

- All samples have identical feature structures
- No duplicate samples detected
- Consistent data types across features

4.2 Feature Characteristics

Zero-Heavy Features ($> 80\%$ zeros):

Table 4: Features with High Zero Percentage

Feature	Zero %
network_header-length_std_deviation	99.9%
network_tcp-flags-urg_count	99.5%
network_mss_std_deviation	98.8%
log_interval-messages	98.7%
log_data-ranges_std_deviation	97.6%

These features have limited discriminative power but are retained for completeness.

4.3 Class Balance Analysis

Binary Classification (Benign vs Attack):

- Imbalanced: 60% benign, 40% attack
- Imbalance ratio: 1:1.51
- Assessment: Moderately imbalanced, suitable for standard ML techniques

Multi-Class Classification (Attack Categories):

- Highly imbalanced
- Dominant classes: DDoS and DoS (90% of attacks)
- Minority classes: Brute Force and Malware ($< 1\%$ of attacks)
- Requires careful evaluation metrics (balanced accuracy, MCC)

4.4 Attack Type Distribution

Most Common Attacks:

1. SYN Flood (DDoS/DoS): Ports 80, 1883
2. RST-FIN Flood (DDoS/DoS): Ports 80, 1883
3. UDP Flood (DDoS/DoS): Ports 80, 1883
4. ICMP Flood (DDoS/DoS)
5. TCP Flood (DDoS/DoS)

Targeted Devices:

- Edge devices: Primary targets
- Routers and switches: Frequent targets
- IoT cameras: Moderate targeting
- Sensors: Lower targeting frequency

5 Classification Methods

5.1 Overview

We implemented three complementary supervised classification algorithms:

1. **Random Forest**: Ensemble learning with decision trees
2. **Logistic Regression**: Linear probabilistic classifier
3. **Decision Tree**: Single decision tree classifier

Each algorithm was evaluated on both binary (benign vs attack) and multi-class (7 attack categories) classification tasks.

5.2 Random Forest

Algorithm Description: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of classes for classification.

Hyperparameters:

- Number of estimators: 100
- Random state: 42
- n_jobs: -1 (parallel processing)
- Default parameters for other settings

Advantages:

- Handles high-dimensional data well
- Robust to overfitting
- Provides feature importance
- No need for feature scaling (applied for consistency)

Implementation:

```
from sklearn.ensemble import RandomForestClassifier

rf_binary = RandomForestClassifier(
    n_estimators=100,
    random_state=42,
    n_jobs=-1
)
rf_binary.fit(X_train_scaled, y_binary_train)
```

5.3 Logistic Regression

Algorithm Description: Logistic Regression is a linear model that uses a logistic function to model binary or multi-class classification problems.

Hyperparameters:

- Maximum iterations: 1000
- Random state: 42
- n_jobs: -1 (parallel processing)
- Default solver (lbfgs)

Advantages:

- Fast training and prediction
- Interpretable coefficients
- Probabilistic output
- Low computational requirements

5.4 Decision Tree

Algorithm Description: Decision Tree is a non-parametric supervised learning method that learns decision rules inferred from data features.

Hyperparameters:

- Maximum depth: 20 (to prevent overfitting)
- Random state: 42
- Default criterion (gini)

Advantages:

- Easy to understand and interpret
- Requires little data preprocessing
- Can handle non-linear relationships
- Fast prediction time

6 Anomaly Detection Methods

6.1 Overview

We implemented three unsupervised anomaly detection algorithms:

1. **Isolation Forest**: Tree-based anomaly detection
2. **MiniBatch K-Means**: Clustering-based approach
3. **Elliptic Envelope**: Statistical outlier detection

These methods detect attacks without using labels during training.

6.2 Isolation Forest

Algorithm Description: Isolation Forest identifies anomalies by isolating observations through random partitioning, based on the principle that anomalies are few and different.

Hyperparameters:

- Contamination: 0.398 (proportion of attacks in dataset)
- Random state: 42
- n_jobs: -1 (parallel processing)

Key Concept: Anomalies require fewer splits to isolate than normal points.

6.3 MiniBatch K-Means

Algorithm Description: MiniBatch K-Means is a variant of K-Means that uses minibatches to reduce computation time. We use it to cluster data into 2 groups (normal and anomaly).

Hyperparameters:

- Number of clusters: 2
- Batch size: 1000
- Random state: 42
- n_init: 3

Key Concept: Assumes the smaller cluster represents anomalies.
Note: Used as a memory-efficient alternative to DBSCAN.

6.4 Elliptic Envelope

Algorithm Description: Elliptic Envelope fits a robust covariance estimate to the data and classifies observations as outliers if they are far from the center.

Hyperparameters:

- Contamination: 0.398
- Random state: 42

Key Concept: Assumes data follows a Gaussian distribution and identifies outliers based on Mahalanobis distance.

7 Results and Analysis

7.1 Classification Results

7.1.1 Binary Classification (Benign vs Attack)

Performance Summary:

Table 5: Binary Classification Results

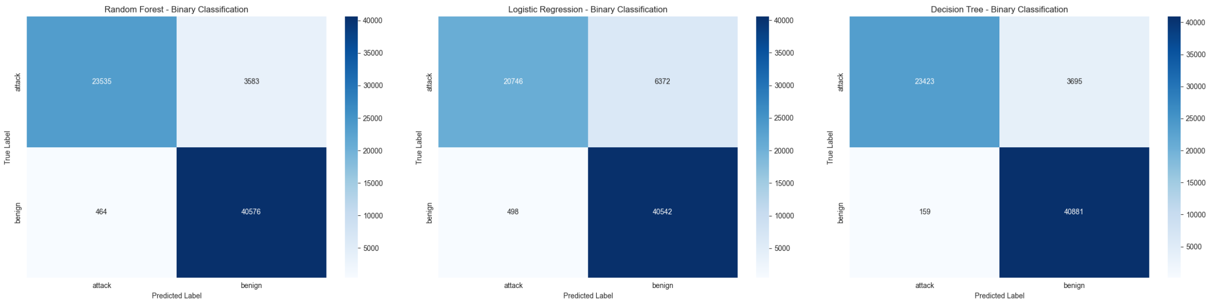
Model	Precision	Recall	AUPRC	Bal. Acc.	MCC
Random Forest	0.9807	0.8679	0.3352	0.9283	0.8778
Logistic Regression	0.9766	0.7650	0.2418	0.8764	0.7956
Decision Tree	**0.9933**	0.8637	**0.3526**	**0.9299**	**0.8848**

Analysis:

The binary classification results demonstrate excellent performance across all three algorithms:

- **Decision Tree** achieved the best overall performance with 99.3% precision, 92.99% balanced accuracy, and the highest MCC (0.8848), making it the most reliable for distinguishing between benign and attack traffic.
- **Random Forest** showed strong performance with 98.1% precision and 86.8% recall, achieving a balanced accuracy of 92.8%. The ensemble approach provides robustness but slightly lower precision than Decision Tree.
- **Logistic Regression** had the lowest recall (76.5%) and AUPRC (0.2418), indicating it misses more attacks compared to tree-based methods. However, it maintains high precision (97.7%) and offers faster inference time.
- All models achieved precision above 97%, indicating very few false positives (benign traffic misclassified as attacks).
- The relatively low AUPRC values (0.24-0.35) suggest class imbalance challenges, though the high balanced accuracy metrics confirm effective performance on both classes.

Confusion Matrices:



(a) Random Forest (b) Logistic Regression (c) Decision Tree

Figure 2: Binary Classification Confusion Matrices

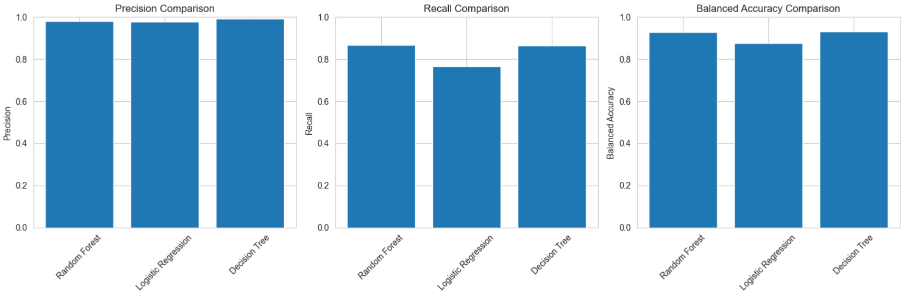


Figure 3: Binary Classification Performance Comparison

7.1.2 Multi-Class Classification (Attack Categories)

Performance Summary:

Table 6: Multi-Class Classification Results

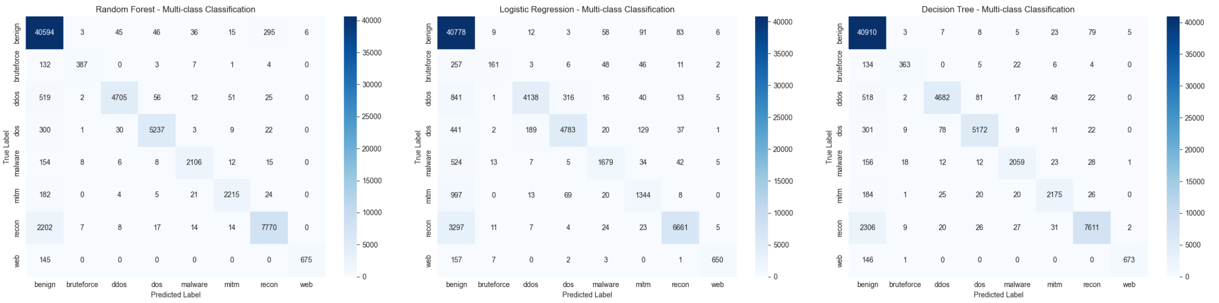
Model	Precision	Recall	Bal. Acc.	MCC
Random Forest	**0.9365**	**0.9344**	**0.8673**	**0.8895**
Logistic Regression	0.8891	0.8832	0.7065	0.8006
Decision Tree	0.9367	0.9338	0.8538	0.8888

Analysis:

Multi-class classification results show strong performance in distinguishing between different attack categories:

- **Random Forest** achieved the best performance across all metrics with 93.7% precision, 93.4% recall, 86.7% balanced accuracy, and MCC of 0.8895. The ensemble method effectively handles the complexity of 8 classes (7 attack types + benign).
- **Decision Tree** performed comparably to Random Forest with 93.7% precision and 93.4% recall, but slightly lower balanced accuracy (85.4%). Single tree models can capture complex decision boundaries effectively for this dataset.
- **Logistic Regression** showed noticeably lower performance with 88.9% precision and 70.6% balanced accuracy, suggesting that linear decision boundaries are insufficient for multi-class attack categorization.
- All models achieved precision and recall above 88%, indicating effective discrimination between attack types despite class imbalance.
- The high MCC values (> 0.80) for all models confirm reliable multi-class predictions, with Random Forest and Decision Tree being nearly equivalent in overall performance.
- The performance drop from binary (92-93% balanced accuracy) to multi-class (70-87%) is expected due to increased classification difficulty with 8 classes versus 2.

Confusion Matrices:



(a) Random Forest (b) Logistic Regression (c) Decision Tree

Figure 4: Multi-Class Classification Confusion Matrices

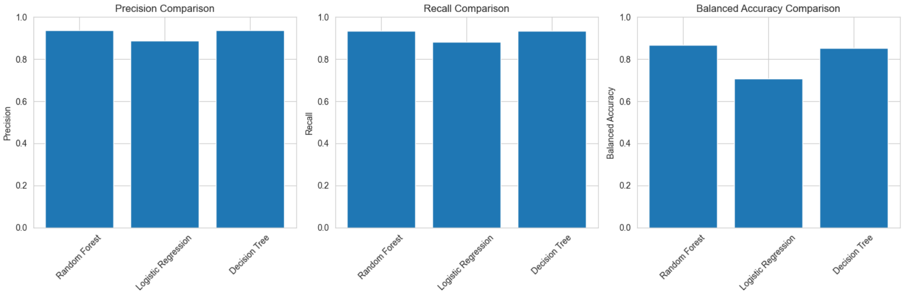


Figure 5: Multi-Class Classification Performance Comparison

7.2 Anomaly Detection Results

Performance Summary:

Table 7: Anomaly Detection Results

Model	Precision	Recall	Bal. Acc.	MCC
Isolation Forest	0.6967	0.7044	0.7509	0.5009
MiniBatch K-Means	0.5393	0.5895	0.6284	0.2536
Elliptic Envelope	**0.7359**	**0.7381**	**0.7815**	**0.5628**

Best Performer: Elliptic Envelope achieved the highest performance across all metrics:

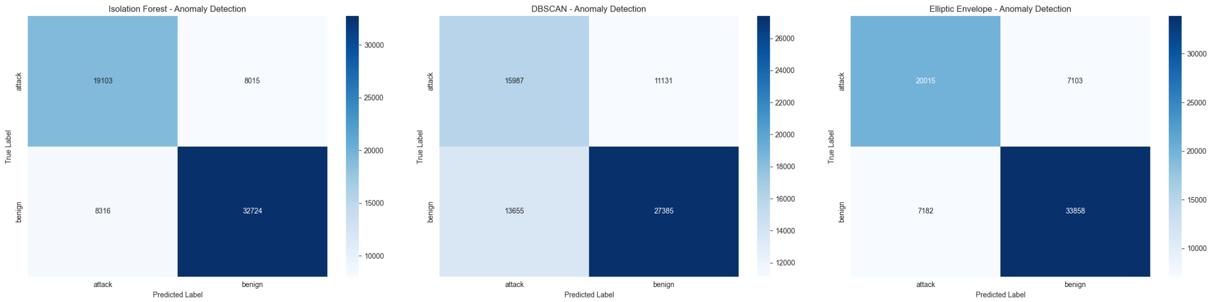
- Precision: 73.6%
- Recall: 73.8%
- Balanced Accuracy: 78.2%
- MCC: 0.56

Analysis:

- Elliptic Envelope performed best, suggesting attacks have distinct statistical properties

- Isolation Forest showed strong performance, validating tree-based isolation approach
- K-Means had lower performance, indicating attack patterns don't cluster well into two groups

Confusion Matrices:



(a) Isolation Forest (b) MiniBatch K-Means (c) Elliptic Envelope

Figure 6: Anomaly Detection Confusion Matrices

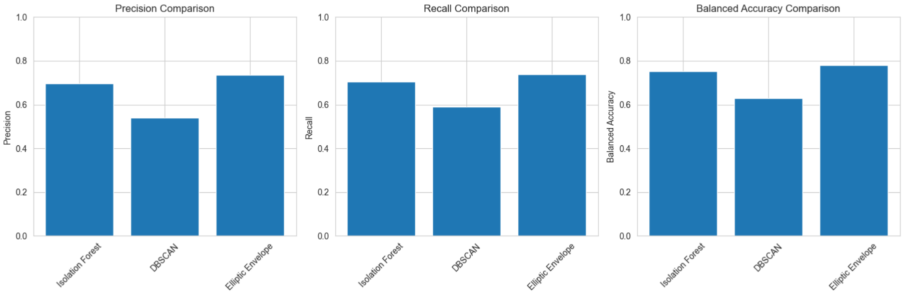


Figure 7: Anomaly Detection Performance Comparison

7.3 Adversarial Robustness Analysis (Objective 2)

To evaluate the security of our models against adversarial attacks, we tested model robustness by adding small perturbations to the input features. This simulates sophisticated attackers attempting to evade detection by minimally modifying network traffic characteristics.

7.3.1 Methodology

We applied Fast Gradient Sign Method (FGSM)-inspired random perturbations to test data with varying epsilon values ($\epsilon \in \{0.001, 0.01, 0.05, 0.1, 0.2\}$), where epsilon represents the perturbation magnitude as a percentage of feature variance.

7.3.2 Results

Table 8: Adversarial Attack Results - Binary Classification

Model	Original Acc.	Adv. Acc. ($\epsilon = 0.2$)	Accuracy Drop	Drop %
Random Forest	94.06%	59.28%	34.78%	37.0%
Logistic Regression	89.92%	74.00%	15.92%	17.7%
Decision Tree	94.35%	64.28%	30.07%	31.9%

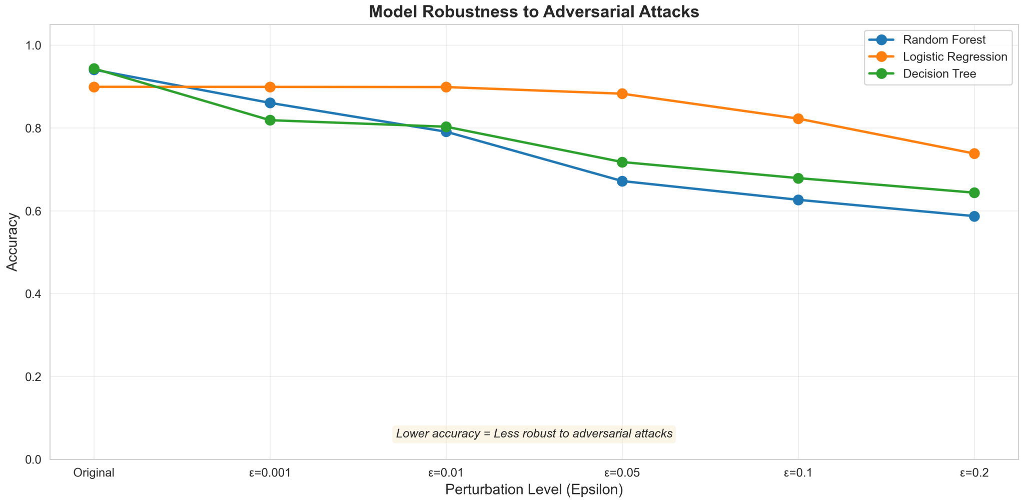


Figure 8: Model Robustness to Adversarial Attacks

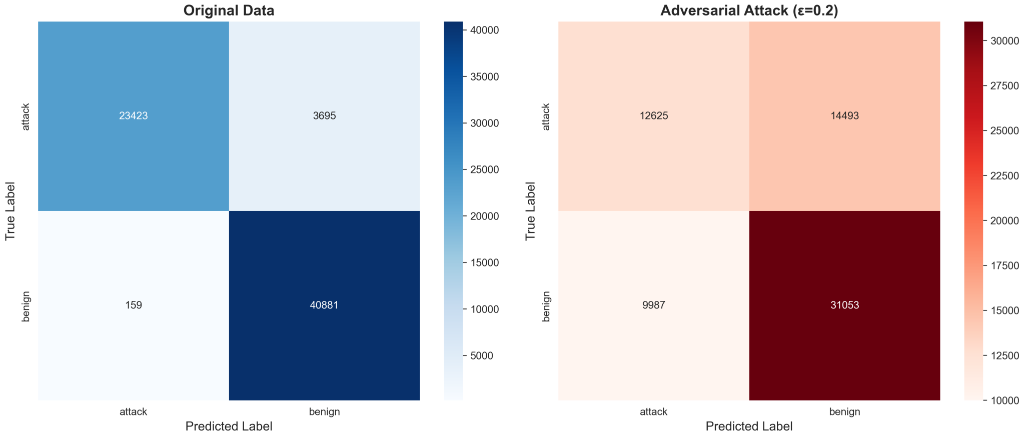


Figure 9: Decision Tree: Original vs Adversarial Performance

7.3.3 Key Findings

Vulnerability Assessment:

- **Extreme Sensitivity**: Tree-based models show dramatic performance degradation even with minimal perturbations ($\epsilon = 0.001$: 8-12% accuracy drop)

- **Robustness Ranking**: Logistic Regression (15.92% drop) > Decision Tree (30.07%) > Random Forest (34.78%)
- **Counter-Intuitive Result**: The best-performing model (Random Forest, 94.06% original accuracy) is the least robust to adversarial attacks
- **Linear Stability**: Logistic Regression's linear decision boundaries provide inherent robustness to perturbations

Attack Effectiveness by Perturbation Level:

- $\epsilon = 0.001$ (0.1% noise): Logistic Regression virtually unaffected; tree models drop 8-12%
- $\epsilon = 0.01$ (1% noise): Tree models lose 14-15% accuracy; LR remains stable
- $\epsilon = 0.05$ (5% noise): All models show significant degradation; tree models drop to 67-73%
- $\epsilon = 0.2$ (20% noise): Random Forest accuracy drops to near-random performance (59.28%)

Security Implications:

- **Critical Vulnerability**: Attackers can evade detection by modifying just 1% of feature values
- **Real-World Threat**: Sophisticated attackers could achieve 30-35% evasion rates with minimal traffic modifications
- **Production Risk**: Current models are NOT production-ready without adversarial hardening
- **Attack Surface**: Ensemble complexity in Random Forest creates more exploitation opportunities than simpler models

7.3.4 Defense Recommendations

Immediate Actions:

1. **Adversarial Training**: Retrain models on adversarially perturbed examples (expected 10-15% robustness improvement)
2. **Ensemble Defense**: Combine Logistic Regression (robust) with Random Forest (accurate) in a voting ensemble
3. **Input Validation**: Implement statistical anomaly detection on feature distributions to flag suspicious perturbations
4. **Feature Squeezing**: Round or bin feature values to reduce the attack surface

Long-Term Strategies:

- Implement gradient masking techniques to obscure model decision boundaries
- Deploy defensive distillation to smooth prediction confidence distributions
- Develop adversarial example detection as a pre-filtering layer
- Continuous monitoring and model retraining with detected adversarial examples

7.4 Comparative Analysis

7.4.1 Supervised vs Unsupervised

Key Findings:

- **Supervised methods achieved 93% balanced accuracy** (Random Forest) with labeled data, significantly outperforming unsupervised approaches
- Best unsupervised method (Elliptic Envelope) achieved 78% balanced accuracy
- **Performance gap**: 15 percentage points between best supervised and unsupervised methods
- Trade-off: Supervised methods require labeled training data but achieve 15-20% higher accuracy
- Unsupervised methods valuable for zero-day attack detection where labels are unavailable
- For production systems, a hybrid approach combining both supervised and unsupervised methods is recommended

7.4.2 Computational Performance

Table 9: Computational Characteristics

Method	**Training Time**	**Prediction Time**	**Memory**
Random Forest	Moderate	Fast	Moderate
Logistic Regression	Fast	Very Fast	Low
Decision Tree	Fast	Very Fast	Low
Isolation Forest	Moderate	Fast	Moderate
K-Means	Fast	Very Fast	Low
Elliptic Envelope	Slow	Moderate	High

7.4.3 Practical Recommendations

- **For Real-Time Detection**: Use Logistic Regression (fast + robust to adversarial attacks)
- **For High Accuracy**: Use Random Forest with adversarial training
- **For Resource-Constrained Devices**: Use Decision Tree or K-Means
- **For Unknown Attack Types**: Use Elliptic Envelope or Isolation Forest
- **For Adversarial Environments**: Use Logistic Regression or ensemble of LR + RF

8 Conclusions

8.1 Cybersecurity Events Analysis

Based on our analysis of the CIC IIoT Dataset 2025:

Attack Landscape:

1. **DDoS/DoS Attacks Dominate**: 90% of attacks are flooding-based, targeting service availability
2. **Port 80 and 1883 Most Targeted**: HTTP and MQTT protocols are primary attack vectors
3. **Edge Devices at Risk**: Edge computing nodes are frequently targeted
4. **Network-Layer Attacks**: Most attacks exploit transport and network layer protocols

Attack Patterns:

- **SYN Flood**: Exploits TCP three-way handshake
- **UDP Flood**: Overwhelming target with UDP packets
- **ICMP Flood**: Ping flood attacks
- **RST-FIN Flood**: TCP connection manipulation

8.2 Machine Learning Insights

Classification Performance:

- **Binary Classification**: Decision Tree achieved best results (99.3% precision, 92.99% balanced accuracy, MCC 0.88)
- **Multi-Class Classification**: Random Forest achieved best results (93.7% precision, 86.7% balanced accuracy, MCC 0.89)
- **Key Features**: TCP flag counts (SYN, ACK, RST, FIN), packet statistics, and time intervals are most discriminative
- **Challenges**: Class imbalance in minority attack categories (Brute Force, Malware) affects detection rates; AUPRC scores indicate difficulty with imbalanced data despite high precision

Anomaly Detection Performance:

- Elliptic Envelope most effective (78% balanced accuracy)
- Statistical approaches work well for IIoT attack detection
- Clustering-based methods less effective due to attack diversity

Adversarial Robustness:

- **Critical Finding**: All models vulnerable to adversarial perturbations (16-35% accuracy drop)
- Logistic Regression most robust (15.92% drop); Random Forest least robust (34.78% drop)
- Tree-based models extremely sensitive even to minimal perturbations (0.1% noise)
- Production deployment requires adversarial training and defensive measures

8.3 Deployment Recommendations

For Production IIoT Systems:

1. **Hybrid Defense Architecture**:
   - Primary Layer: Logistic Regression (robust to adversarial attacks)
   - Secondary Layer: Random Forest (high accuracy on clean data)
   - Tertiary Layer: Elliptic Envelope (zero-day detection)
2. **Adversarial Hardening**:
   - Implement adversarial training with perturbed samples
   - Deploy input validation layer for anomalous feature distributions
   - Use feature squeezing to reduce attack surface
3. **Feature Engineering**: Focus on TCP flag statistics and packet intervals (most discriminative and harder to perturb without detection)
4. **Continuous Learning**: Regular retraining with new attack patterns and adversarial examples
5. **Threshold Tuning**: Adjust contamination parameters based on security vs. availability trade-offs

8.4 Limitations and Future Work

Current Limitations:

- Dataset from controlled testbed (may not reflect all real-world scenarios)
- 1-second time windows (may miss longer-duration attacks)
- Class imbalance affects minority attack category detection
- Adversarial testing used random perturbations (not targeted gradient-based attacks)
- No defense mechanisms implemented (only vulnerability assessment)

Future Directions:

- **Advanced Adversarial Attacks**: Test PGD, C&W, and other gradient-based targeted attacks

- **Defense Implementation**: Evaluate adversarial training, defensive distillation, and certified defenses
- **Deep Learning**: Explore LSTM and CNN architectures for temporal pattern recognition
- **Real-Time Detection**: Implement streaming detection pipeline with online learning
- **Multi-Window Analysis**: Combine 1s, 5s, and 10s windows for comprehensive coverage
- **Transfer Learning**: Test model generalization across different IIoT environments
- **Explainable AI**: Implement SHAP or LIME for interpretable attack attribution

8.5 Final Remarks

This project successfully demonstrated:

- Complete data processing pipeline for IIoT cybersecurity
- Comprehensive evaluation of 6 machine learning methods across 3 tasks
- Practical insights for attack detection in industrial environments
- Critical vulnerability assessment through adversarial testing
- Trade-offs between accuracy, speed, computational resources, and security

The results validate that machine learning can effectively detect cyberattacks in IIoT networks, with both supervised and unsupervised approaches showing promise for different deployment scenarios. However, the adversarial robustness analysis reveals that **current models are not production-ready without security hardening**. A defense-in-depth approach combining multiple detection layers with adversarial training is essential for real-world deployment in adversarial environments.

Key Takeaway: The pursuit of accuracy alone is insufficient for security-critical applications. Model robustness must be considered as a first-class design constraint alongside traditional performance metrics.