Enhancing Security Measures in Electrical Control Systems: A Machine Learning Approach to Detecting and Preventing Intrusion Attacks

by Saif Qasaimeh

B.S. in Mathematics, May 1998, UAE University
B.S. In English Literature, August 2005, Yarmouk University
M.S. in Computer Science, August 2011, NEIU University

A Praxis submitted to

The faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

August 15, 2025

Praxis directed by

Andrea A. Molina
Professional Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of George Washington University certifies that Saif Qasaimeh has passed the Final Examination for the degree of Doctor of Engineering as of July 15, 2025. This is the final and approved form of the Praxis.

Enhancing Security Measures in Electrical Control Systems: A Machine Learning Approach to Detecting and Preventing Intrusion Attacks

Saif Qasaimeh

Praxis Research Committee:

Weina Dorsky, Professorial Lecturer in Engineering and Applied Science, Praxis Chair

Andrea A. Molina, Professional Lecturer in Engineering and Applied Science, Praxis Director

Ould Vadel, Professorial Lecturer in Engineering and Applied Science, Committee Member

© Copyright 2025 by Saif Qasaimeh
All rights reserved

Dedication

The author wishes to dedicate this praxis to my parents, whose unwavering support and endless sacrifices have laid the foundation for all of my achievements; to my wife, whose strength, patience, and encouragement have supported me through the most challenging times; and to my children, whose laughter and love have been endless sources of motivation and joy throughout the journey.

Acknowledgments

The author wishes to acknowledge Dr. Andrea Molina for her mentorship, thoughtful feedback, and steady support throughout the research process. Deep appreciation is also extended to my professors, whose knowledge, expertise, and exceptional technical insight have played a key role in shaping this academic journey. Finally, the author gratefully acknowledges his cousin, Dr. Ahmed Saleh Bataineh, for his recommendation and support at the outset of this endeavor.

Abstract of Praxis

Enhancing Security Measures in Electrical Control Systems: A Machine Learning Approach to Detecting and Preventing Intrusion Attacks

The electrical grid is one of the most crucial elements in industrial infrastructure since it plays a central role in energy supply to various sectors. Due to its importance and intrinsic vulnerabilities, the electrical grid has become one of the favorite targets for highly sophisticated cyberattacks. In particular, these have grown very much recently, since the first confirmed cyberattack on Ukraine's power grid (International Energy Agency, 2021).

Integrated Industrial Internet of Things devices (IIoT) that become part of electrical grids are predominantly sensors and control electronics. Though these devices are very much required to control and monitor electrical systems, they have less computational power and therefore are resource-constrained devices; thus, these devices are prone to cyber attack. With the rise of sophisticated cybersecurity attacks, traditional security mechanisms have failed to provide robust protection for these devices.

Besides, the available machine learning models based on generic IoT and network-related features are not good enough to meet the security requirements of industrial control systems (Neto et al., 2023). In this regard, this research will develop a machine learning-based intrusion detection system that uses industrial-specific features introduced in the CICAPTIIoT2024 dataset (Ghiasvand et al., 2024). The proposed models aim to enhance the accuracy and efficiency of detecting cyberattacks on electrical control systems using tailored features.

This work will compare the performance of different machine learning algorithms with those applied in related works. Comparative analysis will be performed in two

directions: comparing existing learning algorithms with newly implemented models and comparing models with industrial-specific features.

Table of Contents

Dedication **4**

Acknowledgments **5**

Abstract of Praxis **6**

List of Figures **12**

List of Tables **14**

List of Symbols **15**

List of Acronyms **16**

Chapter 1—Introduction **20**

1.1 Background 20

1.2 Research Motivation 21

1.3 Problem Statement 23

1.4 Thesis Statement 23

1.5 Research Objectives 24

1.6 Research Questions and Hypotheses 25

1.7 Scope of Research 26

1.8 Research Limitations 27

1.9 Organization of Praxis 27

Chapter 2—Literature Review **29**

2.1 Introduction 29

2.2 Cybersecurity Threats and Vulnerabilities in Electric Control Devices 30

2.3 APTs in Industrial Internet of Things (IIoT) - Attack Patterns, Impact Analysis, and Mitigation Strategies 31

2.3.1 APT Attacks and Impacts 33
2.3.2 APT Prevention and Detection 35
2.4 CIC IoT 2023 Dataset Related Works 35
2.5 CICAPTIIoT2024 Datasets Characteristic and Related Works 42
2.6 Features and Gap Analysis for CICIoT2023 and CICAPTIIoT2024 Datasets and Related Works. 46
2.6.1 Datasets Features and Gap Analysis 46
2.6.2 Datasets Related Work Summary 47
2.7 Novel Approaches and New Contributions 49
2.7.1 Exploring New ML Algorithms Leveraging IIoT-Specific Features 49
2.7.2 Evaluation of Module Performance Based on Selected Features 49
2.7.3 Utilize SMOTE to Address Class Imbalance 50
2.7.4 Explainable AI with Partial Dependence Plots: 50
2.8 Summary and Conclusion 50
Chapter 3—Methodology 53
3.1 Introduction 53
3.2 Datasets Analysis 54
3.2.1 CICIoT2023 dataset 55
3.2.2 CICAPTIIoT2024 dataset 56
3.3 Datasets Preprocessing 59
3.3.1 Data Cleaning 60
3.3.2 Data Transformation 62
3.3.3 Data Standardization 63

3.3.4 Class Imbalance Treatment 64
3.3.5 Feature Engineering 66
3.4 Machine Learning Algorithms and Classifier Modules. 67
3.5 Conclusion 69
Chapter 4—Results and Analysis 71
4.1 Introduction 71
4.2 Reviewing and Analyzing Research Questions and Hypotheses 72
4.3 Preprocessing and Data Balancing Outcomes 74
4.4 Feature Selection Analysis and Dimension Reduction Review 77
4.5 Training and Fine-Tuning of Machine Learning Algorithmic Module 83
4.5.1 Boosting Algorithms (CatBoost/LightGBM/XGBoost) 84
4.5.2 Regularized Regression Models (Elastic Net) 86
4.5.3 Bagging-based Ensemble Algorithm (Extremely Randomized Trees).. 88
4.6 Classification Modeling and Performance Evaluation 90
4.6.1 *Group One*: Training Outcomes for Balanced and Imbalanced Datasets 90
4.6.2 *Group Two*: Feature Selection Techniques, MI and RFECV, Applied Across Various ML Algorithms 92
4.6.3 *Group Three*: Evaluating Underexplored Classifiers with Emphasis on Industry-specific Features 95
4.6.4 *Group Four*: Cross-Dataset Comparison of Literature-Based Models.. 96
4.7 Model Behavior and Analytical Insights 99
4.8 Explainable AI and Interpretability 107

Chapter 5—Discussion and Conclusions 109

5.1 Discussion 109

5.2 Conclusions 109

5.3 Contributions to Body of Knowledge 111

5.4 Recommendations for Future Research 112

References 113

Appendix A 124

List of Figures

Figure 3-1. Attack Categories Distribution for CICIoT2023 55

Figure 3-2. Traffic Labeled Distribution for CICIoT2023 56

Figure 3-3. Tactic Group Names for CICAPTIIoT2024 58

Figure 3-4. Binary Label Distribution for CICAPTIIoT2024 58

Figure 3-5. Binary Label Counts for CICAPTIIoT2024 59

Figure 3-6. Data Preprocessing Stages 60

Figure 3-7. Data Validation and Cleanup Workflow 62

Figure 3-8. Balanced Datasets Distribution 65

Figure 3-9. Split-First approach Workflow 67

Figure 4-1. Types of Machine Learning 72

Figure 4-2. Preprocessing Pipeline Metrics 76

Figure 4-3. Feature Quality Assessment 77

Figure 4-4. MI Feature Importance 78

Figure 4-5. MI Features Grouping 79

Figure 4-6. RFECV Features Ranking 80

Figure 4-7. RFECV Features Grouping 81

Figure 4-8. Performance Metrics by Feature Selection Method 81

Figure 4-9. Processing Time & Memory Usage During Module Training 82

Figure 4-10. Entropy By Feature Selection Method 83

Figure 4-11. Module Evaluation via Logloss 86

Figure 4-12. AUC Learning Curves 88

Figure 4-13. AUC vs n_estimators Learning Curves 89

Figure 4-14. Original vs SMOTE Module Performance For 47 Features 91

Figure 4-15. Original vs SMOTE Module Performance For 52 Features 92

Figure 4-16. MI vs RFECV Top CICIoT2023 Features Performance..... 94

Figure 4-17. MI vs RFECV Top CICAPTIIoT2024 Features Performance..... 95

Figure 4-18. Comparative Evaluation of Models on CICIoT2023 and CICAPTIIoT2024..... 96

Figure 4-19. Model Performance Comparison Across Studies 98

Figure 4-20. Normality Test for Original and SMOTE Groups 99

Figure 4-21. Kruskal-Wallis Test Results for Group One 100

Figure 4-22. AUC Evaluation for CICIoT2023 101

Figure 4-23. AUC Evaluation for CICAPTIIoT2024 102

Figure 4-24. Group Three Residuals Distribution 103

Figure 4-25. Two-way ANOVA Analysis of Variance 103

Figure 4-26. Percentage Difference in Means 105

Figure 4-27. Mean Score Comparison with Literature Review Studies 106

Figure 4-28. Mean Score Comparison with Perfect Score 106

Figure 4-29. PDP Plots for *"ts"* and *"rst_count"*, Features 107

Figure 4-30. ICE Plot for *"ts"* 108

List of Tables

Table 2-1. Feature Comparison Across Different Data Sources 44

Table 2-2. Distinct Features CICIoT2023 46

Table 2-3. Distinct Features for CICAPTIIoT2024 47

Table 2-4. Summary of CICIoT2023 and CICAPTIIoT2024 Related Works 48

Table 2-5. Selected Algorithms for Machine Learning 49

Table 2-6. Selected Feature Selection Techniques 50

Table 3-1. Binay Classification Details 53

Table 3-2. Fit Regression Model Summary Result 67

Table 3-3. Proposed Machine Learning Algorithms 69

Table 4-1. Adaptation of ML Algorithms in Prior Studies 73

Table 4-2. Balanced vs Imbalanced Evaluation Results for 47 Features 91

Table 4-3. Balanced vs Imbalanced Evaluation Results for 52 Features 92

Table 4-4. MI vs RFECV Top CICIoT2023 Features 93

Table 4-5. MI vs RFECV Top CICAPTIIoT2024 Features 95

Table 4-6. Comparative Evaluation of Models on CICIoT2023 and CICAPTIIoT2024 . 96

Table 4-7. Cross-Dataset Model Performance Comparison 97

Table 4-8. Percent of Variance Explained (η^2) 104

List of Symbols

Symbol	Definition
η^2	eta-squared of effect size
SS_{factor}	Sum of squares for the effect of between-group variability
SS_{Total}	Total sum of squares
SS_{res}	Sum of the squared differences between observed and predicted values
SS_{tot}	Sum of the squared differences between observed values and the mean
p-value	probability value
DPKT	Python module for fast, simple packet parsing
K-Means	K-Means clustering
R^2_{adj}	Adjusted *R-squared*
R^2_{pred}	Predicted *R-squared*
R^2	*R-squared* in Regression
L1	Lasso in Elastic Net Regularization
L2	Ridge Elastic Net Regularization

List of Acronyms

Acronym	Definition
AI	Artificial Intelligence
ANOVA	Analysis of Variance
APTs	Advanced Persistent Threats
AUC	Area Under the Curve
BRFC	Balanced Random Forest Classifier
CatBoost	Categorical Boosting
C2	Command and Control
CFS	Correlation-based Feature Selection
CNN	Convolutional Neural Network
CoAP	Constrained Application Protocol
CSV	Comma-Separated Values
DDoS	Distributed Denial of Service
DNN	Deep Neural Network
DNP3	Distributed Network Protocol 3
DoS	Denial of Service
DT	Decision Tree
EDA	Exploratory Data Analysis
Extra Trees	Extremely Randomized Trees

Acronym	Definition
GB	Gradient Boosting
GSK	Gaining–Sharing Knowledge
GRU	Gated Recurrent Units
HDFS	Hadoop Distributed File System
HPC	High-Performance Computing
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICS	Industrial Control System
ICMP	Internet Control Message Protocol
IED	Intelligent Electronic Devices
IG	Information Gain
IGMP	Internet Group Management Protocol
IIoT	Industrial Internet of Things
IIDS	Intelligent Intrusion Detection System
IOBs	Indicators of Behavior
IOCs	Indicators of Compromise
IoT	Internet of Things
IRM	Importance Ranking Method
KNN	K-Nearest Neighbors

Acronym	Definition
LDA	Linear Discriminant Analysis
LightGBM	Light Gradient Boosting Machine
LIME	Local Interpretable Model-agnostic Explanations
LR	Logistic Regression
LSTM	Long Short-Term Memory
M2M	Machine-to-Machine
MDI	Mean Decrease in Impurity
MI	Mutual Information
ML	Machine Learning
MLP	Multi-Layer Perceptron
NSA	National Security Agency
PCA	Principal Component Analysis
PCAP	Packet Capture
PDPs	Partial Dependence Plot
PLCs	Programmable Logic Controllers
QDA	Quadratic Discriminant Analysis
RF	Random Forest
RFE	Recursive Feature Elimination
RFECV	Recursive Feature Elimination with Cross-Validation

Acronym	Definition
RNN	Recurrent Neural Networks
ROC Curve	Receiver Operating Characteristic Curve
ROS	Random Over Sampling
RTU	Remote Terminal Units
SCADA	Supervisory Control and Data Acquisition
SMOTE	Synthetic Minority Over-sampling Technique
SMOTE-NC	SMOTE for Nominal and Continuous
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TinyML	Tiny Machine Learning
UDP	User Datagram Protocol
xAI	Explainable AI
XGBoost	EXtreme Gradient Boosting

Chapter 1—Introduction

1.1 Background

Critical infrastructure in any modern city, with its interconnectedness, provides the backbone of essential services and enables the social entity to function smoothly. New methods of connectivity and services are being adopted to make the systems more efficient and resilient as technology advances. For example, digital technologies for sharing real-time information are crucial in managing electrical grids to optimize operations, grid stability, and resilience. According to the International Energy Agency, the potential annual savings from digitalization in the electricity sector could reach $80 billion by 2040 (International Energy Agency, 2021). These advancements do not come without challenges. Adopting new technologies introduces cybersecurity threats that have never been seen or imagined, making traditional cyber controls less effective and more predictive.

Historical incidents, such as the 2003 Northeast Blackout ("The 2003 Northeast Blackout: A Case Study for the Learner," 2015) and the 2005 Texas City refinery explosion (contributors, 2024), will highlight the potentially catastrophic outcomes of digital failures. The Northeast Blackout, which affected 50 million people across eight northeastern states, as well as parts of Canada, due to a software bug compounded by human error, reportedly caused economic losses ranging between $ 4 and $ 6 billion. The Texas City refinery explosion, which took 15 lives and $200 million in property damage, was similarly caused by faulty instrumentation and human error. These incidents show that even though the causes were not based on cyberattacks, the damage from digital failures could be catastrophic and life-threatening.

Digitalization of critical infrastructure, such as the electrical sector, is unavoidable. However, there is a catch: it is still an uphill task to balance introducing new technologies with assurances of secure, uninterrupted operational services. Throughout history, securing industrial control devices has not been sufficient to enable scalability of security or defence of infrastructure components. As a result, Artificial Intelligence (AI) and Machine Learning (ML) are drawing a lot of attention because of their sophisticated security solutions for Industrial Internet of Things (IIoT) devices. ML is giving tremendous advantages in terms of accuracy, capacity, and speed. However, ML models still require human intelligence for interpretation and decision-making.

The presence of big data provided by CICIoT2023 and CICAPTIIoT2024 has benefited the generation of high accuracy ML models based on various ML algorithms. The new industrial features in the CICAPTIIoT2024 datasets can potentially benefit the existing and newly created ML models, according to solid evidence. Different combinations of feature sets may also increase the accuracy of ML models' predictions. The ability to capitalize on such a rapidly developing area will, however, be essential to extend these improvements in security to electrical control systems in protecting critical infrastructures from newly evolving cyber threats.

1.2 Research Motivation

The economic consequences of cyberattacks on electrical systems could be staggering. In Lloyd's scenario ("Business Blackout," 2015), such attacks could have financial costs of $243 billion, with a slight increase in death rates due to the failure of health and safety systems. In the worst-case scenario, these costs could exceed $1 trillion.

This highlights the importance of implementing robust security measures to safeguard electrical infrastructure against cyber threats.

In response to this challenge, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) has proposed a budget of about $3.01 billion for the fiscal year 2025, of which $1.7 billion is clearly devoted to CISA's Cybersecurity Division to enhance the detection and response of cyberattacks against critical infrastructure (Edwards, 2024).

The financial implications of cyberattacks on critical infrastructure underscore the need to invest in complex and robust detection and prevention mechanisms; such measures are justified by the potential impact caused by the halting of critical infrastructure systems. As cyber threats evolve, so too must our defenses, which require the development of sophisticated security solutions.

There is strong demand for the availability of big data in this effort such as on the occasion of CICAPTIIoT2024 and more. These datasets bring features from industries that could enhance the performance of existing ML models and help create new models with better prediction accuracy. By using new technologies, researchers can create scalable and effective ML-based solutions for Industrial Control System (ICS) security that protect critical infrastructures against diverse cyber threats. This research direction helps to address limitations of the state-of-the-art and opens novel opportunities for enhancing ICS security in the digital age.

1.3 Problem Statement

Misconfigurations in electrical control systems resulting from cyberattacks can lead to grid system failures and increased operational costs. This is evidenced by infrastructure disruptions that cost utilities $17.8 million per company annually (Xu, 2020 & International Energy Agency, 2021)

Because of the Smart Grid implementation, the electrical grid is becoming more interconnected and interdependent (Business Blackout, 2015). This is increasing visibility and reliability in ICS while reducing costs. Smart devices may let you control things remotely, but that also leaves a lot more ways for hackers to cause issues. That creates new ways for something to go wrong. The average annual cost of a cyber-attack in the utilities sector is $17.8 million per company (International Energy Agency, 2021). Companies may underreport incidents due to liability and reputational damage. As the U.S. power grid transitions to an ICS-enabled infrastructure ("Cyber Threat and Vulnerability Analysis of the U.S. Electric Sector: Mission Support Center Analysis Report," 2016), the combined presence of legacy equipment, intelligent electronic devices and bidirectional communication results in exploitable vulnerabilities for these grid systems. Cyber-physical attacks targeting any of these systems could result in equipment malfunctioning, power disruptions, or even blackouts.

1.4 Thesis Statement

A machine learning solution with industrial-specific features in electrical control devices is crucial for enhancing system resilience against cybersecurity intrusion attacks.

The novel implementations of ML, which are embedded with industrial-specific features such as protocols used for device interconnection, are crucial in enhancing the system's resilience against cybersecurity intrusion attacks. In the operation of electrical control devices, such protocols are fundamental for exchanging real-time data and coordinating operations through complex industrial environments. By leveraging this industrial knowledge and data, ML models can more effectively identify anomalies and potential threats. Our approach will enable for a more robust defense mechanism tailored to the needs and vulnerabilities of an ICS. In other words, the goal of this research is to stop the operation of critical infrastructures from being disrupted by cyber attacks, which in turn can lead to huge economic and societal costs.

1.5 Research Objectives

1. *Develop Machine Learning Models with High Accuracy*: This work aims to develop new ML models with improved accuracy by incorporating industry-specific features. Unlike prior studies that focus on general IoT network traffic, the proposed method tailors its analysis to a specific industrial context.
2. *Evaluate Feature Selection and Performance*: Apply feature selection methods in a diversified approach to obtain a feature set with high correlation and then compare the performance of algorithms to combine these feature sets and establish the best approach.
3. *Address significant class imbalance*: Apply well-established techniques, such as Synthetic Minority Over-sampling Technique (SMOTE), to handle the substantial class imbalance present in the dataset. The goal is to improve the reliability of the

model's predictions by ensuring that minority classes are adequately represented during training.

4. *Apply Explainable AI (XAI) Techniques*: Use an explainable AI (xAI) method to gain insight into how the models make decisions and identify the features that most influence the outcomes.

1.6 Research Questions and Hypotheses

The following research questions and corresponding hypotheses are outlined to meet the research objectives and provide sufficient evidence to support the thesis:

RQ1: Can industry-specific features utilized in unexplored ML algorithms enable cyber intrusion detection models to achieve a high accuracy rate for binary classification in IIoT environments?

H1: Incorporating features related to IIoT, such as the Internet Group Management Protocol (IGMP) protocols, enables the model to achieve a binary classification accuracy rate of 99.99%.

RQ2: Can machine learning algorithms trained on industry-specific features achieve a higher binary classification accuracy rate in cyber intrusion detection compared to models trained on generic IoT features?

H2: Models trained on industry-specific features will achieve a binary classification accuracy rate of at least 99.99%, outperforming existing pre-trained models developed for generic IoT-based cyber intrusion detection.

RQ3: Can combining a permutation of machine learning models and feature selection techniques improve benchmark performance by decreasing the False Positive Rate (FPR) and increasing the Area Under the Curve (AUC) by 10%?

H3: A blend of diverse machine learning algorithms and feature selection techniques, such as Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV) techniques, will lead to models with improved FPR and the AUC by 10%

1.7 Scope of Research

This research focuses on intrusion detection using ML modules on industrial devices using the CICAPTIIoT2024 dataset. Previous studies have explored various ML algorithms on IoT devices using the CICIoT2023 datasets. As such, this study aims to train various machine learning classifiers using different feature selection techniques to identify impactful features and develop high-performance prediction modules.

The scope of this study is to integrate industry-specific features from the CICAPTIIoT2024 dataset and compare the performance of trained modules with these additional features against previously trained modules. Furthermore, the study tries to understand the correlation of industrial-specific features on the performance of prediction modules through the implementation and analysis of different feature selection methods. This research will also apply a new xAI technique to explain model predictions and the importance of features, such as Partial Dependence Plots (PDP).

1.8 Research Limitations

The first step to extending the work of this research in future studies is to acknowledge the limitations present and identify the gaps that need to be addressed for real-life implementation. The following points outline the critical limitations of this research:

- The CICAPTIIoT2024 and CICIoT2023 datasets have been collected using a simulated testbed. In these datasets, some nodes simulate attacking machines, while others are set up to be targets for various types of attacks.
- Although the CICAPTIIoT2024 dataset includes communication features for resource-constrained industrial devices, it does not provide features focused on industrial process control and monitoring, such as Distributed Network Protocol 3 (DNP3) and Modbus protocols.
- Collecting data from real-life electrical control devices is challenging for various reasons, including the proprietary nature and security concerns.
- Data collected from real-life devices installed in industrial fields has limitations regarding data size, the type of traffic being captured, and data format.

By addressing these limitations, future studies can be conducted to offer practical and real-world applications.

1.9 Organization of Praxis

This research is divided into five chapters, each with its own purpose that contributes to the overall framework of the research. Chapter 1 acts as an introduction, providing insight into the research topic. In addition, this chapter highlights the research objectives and outlines how these can be achieved through a presentation of the research

questions and hypotheses. In essence, this chapter provides the starting point of a study by establishing the foundation elements that guide the research.

Discussion of the Literature Review and Related Works is presented in Chapter 2. This chapter lays a broad foundation for the study, reviewing the existing literature and highlighting the areas where this research will build upon it. Hence, it highlights the gaps in current research and demonstrates how this study will add to the existing body of knowledge.

The third chapter focuses on the methodologies adopted in the research. It discusses the use of several ML algorithms and their application in developing predictive modeling components. It compares several methods of feature selection to build a more effective prediction model.

Chapter 4 discusses results. This chapter fully exploits the module performance results by comparing the evaluation metrics. It also contains a statistical analysis of outcomes for the different feature selection techniques to find if there is a significant difference among the selected feature sets.

Lastly, chapter 5 closes the research study by providing summary statements on the key additions made to the knowledge base. It identifies a set of features that yielded a high-performance prediction module. Additionally, this chapter evaluates the strength and soundness of the thesis statement by examining the outcomes of the research questions and their corresponding hypotheses. It also evaluates the limitations of the proposed approach and outlines directions for potential extensions and future in-depth research.

Chapter 2—Literature Review

2.1 Introduction

The opening statement by Cybersecurity and Infrastructure Security Agency (CISA) Director Jen Easterly before the House Select Committee in 2024 underscores the consequential and far-reaching cyber threats posed by state-sponsored actors, particularly from China, against U.S. critical infrastructure (Easterly, 2024). Easterly's warning that "Chinese cyber actors are burrowing deep into our critical infrastructure to be ready to launch destructive cyber-attacks in the event of a major crisis or conflict with the United States" highlights the urgency and potential consequences of these threats. Jen Easterly adds that cyber intrusions in sectors such as energy likely represent only "the tip of the iceberg," hinting at the hidden scale of these attacks.

With the dynamic complexity of intrusion techniques, traditional cybersecurity defenses have become increasingly insufficient. Cyber actors have adapted their methods to evade conventional detection systems, exposing the ineffectiveness of traditional approaches in preventing significant disruptions. As these threats continue to evolve, especially in electrical control systems, this chapter tries to review different types of cyberattacks and Advanced Persistent Threats (APTs) relevant to critical infrastructure. It will also explain the techniques cyber actors use to bypass the detection systems and examine the vulnerabilities they exploit.

This chapter examines recent research that leveraged the CICIoT2023 dataset to address security threats through ML models. By including the CICAPTIIoT2024 dataset, which incorporates industrial-specific features, this study seeks to enhance ML model performance in detecting sophisticated intrusion tactics. Comparisons will be made

between existing models developed using the CICIoT2023 dataset and new approaches utilizing CICAPTIIoT2024, evaluating whether these industrial features improve detection accuracy.

Finally, this chapter summarizes key findings, identifies gaps in the literature, and suggests future research opportunities for advancing cybersecurity in industrial environments. Such a review provides some grounds for developing resilient models with respect to detecting and mitigating advanced cyber threats against critical infrastructures. It contributes to the continually emerging body of research known as cybersecurity.

2.2 Cybersecurity Threats and Vulnerabilities in Electric Control Devices

Modern Electric control devices are extensively used to support electric distribution systems, facilitate power restoration, and maintain grid stability. While the enhanced functionality provided by intelligent technology improves efficiency and remote capabilities, it simultaneously increases exposure to cyber risks. Integrating Programmable Logic Controllers (PLCs) within these devices is a primary factor in their vulnerability to cyberattacks.

The added layers of complexity and connectivity in modern control devices introduce cybersecurity vulnerabilities absent in traditional systems. Significant vulnerabilities lie in the following key reasons:

- **Embedded Software and Firmware:** Smart electronics use complex embedded software and firmware that enable advanced functionality. However, firmware becomes an attack vector if a threat actor manipulates it. Firmware vulnerabilities can be exploited, potentially leading to system compromise and operational risks (Yuldashev & Solovev, 2022).

- **Reliance on IP-based and Wireless Protocols:** Many smart electronics rely on IP-based and wireless communication protocols, similar to those used in Supervisory Control and Data Acquisition (SCADA) systems. Without solid security configurations or encryption, these protocols are vulnerable to interception, spoofing, and manipulation, allowing attackers to gain access and alter device behavior (Hareesh et al., 2020).
- **Remote Access and Control Capabilities:** Smart electronics enable remote access and control of configuration changes and adjustments that don't need field technicians. Attackers, on the other hand, bypass authentication and can modify configurations—possibly altering settings in devices, thus causing failures or unintended operations of service (Mitigations for Security Vulnerabilities Found in Control System Networks, n.d.).
- **Dependence on Cloud Services and Centralized Control Systems:** Many smart electronics now interact with cloud platforms for data storage, monitoring, and control. This dependence expands the attack surface and exposes control devices to additional cyber risks.

2.3 APTs in Industrial Internet of Things (IIoT) - Attack Patterns, Impact Analysis, and Mitigation Strategies

Since PLCs are critical electronic control devices, controlling electricity grids and SCADA systems, they are the desired targets of APTs. With their sophisticated nature, stealthiness, and disruptive capability, APTs are taken very seriously by attackers, including attackers of the nation-state type and organized groups of players.

This section explores the details of various attributes that distinguish APTs from common cybersecurity threats (Gan et al., 2023).

- **Stealthy Nature:** APTs mimic normal traffic behavior by maintaining volume and frequency within acceptable limits. They use standard protocols such as Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) and widely used tools like PowerShell and Task Scheduler, challenging detection and differentiation from legitimate activity (Singamaneni & Singh, 2024).
- **Sophisticated Techniques and Tactics:** APTs use the most sophisticated, multi-stage attack techniques: reconnaissance, initial access, Command-and-Control (C2), and lateral movement. These add layers of complexity, allowing attackers to evade detection and remain undetected within targeted systems for extended periods.
- **Targeted Systems:** APTs are generally targeted at high-value, critical targets and are not unleashed in the form of broad attacks. The main targets are government infrastructure and private organizations with a essential missions. Compromising these systems can lead to significant disruption, hinder normal operations, and result in political and financial consequences. For instance, the Colonial Pipeline ransomware attack showed just how serious these kinds of attacks can be: imposing chaos and massive economic damage (Easterly & Fanning, 2023).
- **Long-Term Persistence:** Long-term persistence helps the attackers achieve strategic goals while remaining unnoticed within the compromised system. Goals like data exfiltration usually require continuous monitoring and can also generate minimal immediate noise. The SolarWinds breach is one of the most prominent recent examples, lasting more than a year before being detected, well above the average

dwell time for cyberattacks reported by CrowdStrike in 2019 (Oladimeji & Kerner, 2023).

- **Adaptability:** APTs evolve rapidly, adapting their techniques to changes in technology and defenses. These attacks typically require highly skilled actors, such as nation-state groups or smaller entities like APT29 (also known as Cozy Bear, The Dukes, or Solar Storm). These actors are trained and equipped to execute sophisticated and prolonged operations (MITRE ATT&CK, n.d.).

2.3.1 APT Attacks and Impacts

APTs are sophisticated and potentially devastating cyberattacks launched by threat actors with specific objectives against critical infrastructure, governments, and major corporations. Some key characteristics of APT attacks include complexity, large scale, and long-term undected patterns in networks. The primary APT attackers are well-resourced and skilled threat actors often linked to nation-states (Greenberg, 2019).

NotPetya was launched in 2017 by a group named Sandworm, allegedly belonging to the Russian military intelligence. As Greenberg noted, it caused more than $10 billion worth of damage worldwide. Primarily directed towards Ukraine, NotPetya went on to target organizations worldwide and managed to hit some of the major companies, including Maersk, FedEx, and Merck. One of the most destructive impacts perpetrated by APT was illustrated by NotPetya, which needed more robust cybersecurity measures rather than simply stealing data, using highly destructive malware.

The attacks against Ukraine's power grid in 2015 and 2016 are also examples of the capabilities of APT actors. Attributed to Sandworm, these attacks exposed the capability to tareget critical infrastructure using complex techniques such as phishing,

malware deployment, and industrial control system exploitation to cause major power outages to tens of thousands. Those attacks were the first confirmed instances of hackers causing power outages, raising concerns about the vulnerability of critical infrastructure worldwide.

The 2017 WannaCry ransomware attack was pinned on North Korean hackers, showing that APTs could have a global reach. Greenberg mentioned that the attack infected over 200,000 computers using stolen National Security Agency (NSA) tools, causing widespread disruption to healthcare services, transportation systems, and businesses. Less complex than some of the other APTs, the WannaCry attacks highlighted the real dangers posed by unpatched systems and demonstrated how nation-state hacking tools are repurposed for cybercrime.

The Olympic Destroyer malware attack in 2018 on the Winter Olympics in Pyeongchang, South Korea, was a good example of how APT actors can be both highly sophisticated and capable of conducting sophisticated deception operations. Greenberg observed that Russian hackers allegedly leveraged various techniques to spread the malware and wipe data, shutting down the infected systems. The attackers embedded false flags to mislead investigators, demonstrating the complexity of APT operations.

In summary, APTs pose significant and dynamic challenges to the cybersecurity landscape. Given its persistence, sophistication, and potentially devastating impact, the consequences of an APT are something no organization or government can afford. Understanding the very nature of APTs, their tactics, and the need for comprehensive, proactive defense strategies to mitigate risks from such advanced cyber threats is essential.

2.3.2 APT Prevention and Detection

APTs are very difficult to detect and protect against because they are persistent and sophisticated in nature (What Are Advanced Persistent Threats?, n.d.). Building a defendable posture involves creating one based on deep visibility and analysis of the endpoints and across the entire network, assuming that attacks are in progress. Security teams must monitor patterns and changes that indicate long-term threats over time. Traditional endpoint security solutions are no longer sufficient, relying solely on known Indicators of Compromise (IOCs). Behavioral detection via Indicators of Behavior (IOBs) is required to catch those first hints of an attack.

The IOBs solution technique leverages the capabilities of analyzing vast volumes of data, spotting complex patterns, and adapting to moving targets. Threat detection can be automated since the technology remains busy monitoring network traffic and behavior and marks suspicious activities associated with APTs. ML models trained to an advanced level learn from new data and improve the quality of detection. Not only do they assist in decreasing false positives, but they also direct security teams towards real threats.

2.4 CIC IoT 2023 Dataset Related Works

(Yan & et. al. 2024) developed a module called DGConv-IDS that was trained and tested on three datasets, namely CICIoT2023, CICDDoS2019, and NSL-KDD. The news emphasizes how efficient the module's accuracy is within limited computational resources. The system uses a quick window combo of an autoencoder and a dynamic group convolution neural network to process recent data. It reduces the overhead and still offers sufficient detection accuracy. In addition, the dynamic group convolution strategy allows the model to adjust its complexity and hence, became suitable for the real-time

Distributed Denial of Service (DDoS) attack detection. This makes the detection more efficient without incurring any high cost which will help in improving the IoT security systems.

The study by (Patidar et al., 2024) proposes different security measures for the Internet of Things (IoT). However, it does not employ ML algorithms, or use any ML module. The study uses Apache Hadoop and Hive for scalable data processing using the CICIoT2023 dataset which consists of extensive data on IoT attacks. This study focuses on employing advanced analysis using data visualization tools such as Microsoft Power BI, TABLEAU and Excel for getting actionable insights from complex datasets. The most common attacks were DDoS; the most popular protocol targeted was Transmission Control Protocol (TCP), and the flags with the highest used in the DDoS-TCP-Flood scenario were SYN and ACK. The method is scalable to large datasets, and ultimately gives insights into IoT security, which add to developing greater security solutions.

(Yaras and Dener, 2024) proposed a combined model of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM). It brings a new hybrid deep learning algorithm for binary intrusion detection and multiclass classification. This attacks IDS model uses CNNs to extract features while the subsequently deployed LSTM has to capture the temporal interdependencies of the traffic data. This light model has shown high accuracy in detecting malicious attacks and can be used in real-time on resource-constrained IoT devices when tested on the famous CICIoT2023 and TON_IoT datasets. This study offers a scalable and efficient solution for enhancing security in IoT.

The study by (Huynh et al., 2023) focuses on DoS attack detection using the CICIoT2023 dataset and real-time data. It employs Tiny Machine Learning (TinyML)

frameworks, such as TensorFlow Lite Micro, to deploy the module on ESP32 microcontroller boards. Features extracted from real-time network traffic data captured during a De-Authentication DoS attack include frame length and packet inter-arrival time. Support Vector Machine (SVM) and Logistic Regression (LR) models achieved high accuracy (99.36% for SVM and 99.15% for LR) on real-time data. The Model types trained on the CICIoT2023 dataset showed slightly lower accuracy but greater versatility across attack types. The study advocates for future research on advanced microcontrollers and diverse deep neural network models.

The study by (Okey et al. 2024) on federated learning algorithms shows the effect of this approach on mitigating the risks associated with weak default security measures in IoT devices, which are particularly susceptible to increasingly malicious attacks. The proposed federated ensemble Intrusion Detection System (IDS) in this study, utilizing deep learning models such as CNN-GRU and LSTM-GRU, involves two federated learning algorithms, wFedAvg and wFedProx, which are applied to two datasets: CICIoT2023 and FLNET2023. Federated learning algorithms are designed to optimize model training across decentralized systems, preserving data privacy and avoiding the sharing of sensitive data. The generated models achieve high accuracy, up to 99.25% for CNN-GRU and 95.75% for LSTM-GRU across the datasets. The results demonstrate that the proposed models enhance IoT security while maintaining data privacy, outperforming several existing methods in accuracy and training efficiency.

Five machine learning modules were trained and evaluated by the study of Kumar et al. (2024) using CICIoT2023. The study employs a distinct approach by classifying attacks into three different groups. Each attack is labeled separately in the first

classification group to produce 34 labels (33 attack types + 1 benign). The second group of attacks is classified into seven categories, reducing the total classification labels to eight (seven attack types + 1 benign). The third grouping approach brings all attacks under one class, resulting in two classes (1 + 1). The study evaluates the performance of RF, Decision Tree (DT), AdaBoost, SVM, and LR in classifying these attacks. The results indicate that RF and DT achieve near-perfect accuracy, while SVM and LR perform less effectively, especially in multiclass scenarios.

The Principal Component Analysis (PCA) utilized in the study by (Pirtama et al., 2024), employing the CICIoT2023 dataset, demonstrates perceptible enhancement in network intrusion detection. PCA reduces 47 initial features by identifying essential features and analyzing correlations. Feature subsets of 5, 8, or 10 were extracted and used as inputs for the Random Forest (RF) algorithm. The study yielded excellent results, achieving an average accuracy of 99.2% across all feature sets. The highest accuracy of 99.19% was achieved by using five PCA-extracted features, hence proving very effective in enhancing IoT network attack detection.

The study by (Ramli and Thereza, 2023) focuses on developing lightweight ML-based intrusion detection models tailored for IoT devices. The research evaluates DT, K-Nearest Neighbors (KNN), RF, and Naïve Bayes algorithms, employing the CICIoT2023 dataset. With the assistance of the Chi-square algorithm for feature selection, both RF and DT models achieved perfect accuracy (100%), highlighting their efficacy in addressing IoT security challenges.

(Tseng et al., 2024) picks neural network models. The Deep Neural Network (DNN), Recurrent Neural Networks, CNN, LSTM, CNN + RNN, CNN + LSTM and

Transformer module have been used in the study for analysing the network traffic and segregate malicious behaviour. The approach in the research is binary and multi-class classification. Although the accuracy of Transformer model in binary classification was not as good as DNN and CNN + LSTM, it was better than them in multi-classification (99.40% accuracy).

The Gaining–Sharing Knowledge (GSK) optimization algorithm is used, and it reduces the size of the CICIoT2023 dataset by 62.45% (Gheni and Al-Yaseen, 2024). The above reduction is utilized along with the MLP Algorithm to produce a model with 97.46% accuracy multi-class classification and 99.26% in binary classification. The merging of GSK and MLP shows a very efficient and accurate method for improving network security and intrusion detection performance.

The Residual Memory Convolutional Neural Network (RMCNN) model improves multi-class intrusion detection (Liu et al., 2024). By interleaving CNN for extracting the spatial feature and Gated Recurrent Units (GRU) for modeling temporal feature. The model achieves an accuracy of 97.28%. The performance comparison of RMCNN with other models indicates promising outcomes for multi-class intrusion detection task.

The study examines the Extra Tree classifier used for feature selection (Khanday et al., 2024). The application of it on the CICIoT2023 dataset helps in improving the performance of ML models by effectively managing large datasets with many features. The LSTM component achieves a reliable 92%; nevertheless, the model, 1D-CNN, achieves an overwhelming 99.87%, suggesting that the full classifier would actually upgrade the model if it had optimization for better feature selection.

The study by (Modi, 2024) proposes a combination of feature selection technique based on different statistics to measure the variable relationships. This technique is applied to enhance performance of an EXtreme Gradient Boosting (XGBoost) model. Utilizing metrics such as Pearson correlation, Spearman correlation, Kendall correlation and Information Gain (IG), an overall accuracies of 99.993 % on the CICIDS2017 dataset and 97.64 % on the CICIoT2023 dataset has been obtained for feature selection. Including several parameters for unusual activity detection feature selection methods will show the efficacy of the techniques.

The study by (Becerra-Suarez et al., 2024) evaluates DNN, LSTM, and CNN models using the CICIoT2023 dataset. With its hyperparameters tuned, the CNN model gives the best performance of 99.10% in multi-class and 99.40% in binary classification. Additionally, it takes less time during training and inference for this CNN than for the compared models; thus, this proves its better effectiveness and efficiency.

The study by (Gupta et al., 2024) proposes a new hybrid voting classifier which is built along with LR, Adaptive Boosting (AdaBoost), Gradient Boosting (GB), and MLP. The study was assessed using the CICIoT2023 dataset. The hybrid model proved to be more effective than the individual classifiers as it was able to achieve 99.57%, 99.39% and 98.95% accuracy for binary, 8-class and 34-class classification respectively; thus, it is robust in its ability to detect IoT network anomalies.

(Le et al., 2023) propose combining multiple ML algorithms to enhance detection accuracy and interpretability. Tested on the CICIoT2023 and IoTID20 datasets, the developed modules outperform baseline models in terms of accuracy, Area Under the Curve (AUC), precision, recall, and F1 scores. Using a combination of counterfactual and

Local Interpretable Model-agnostic Explanations (LIME) techniques can make the model interpretable, leading to a transparent explanation, a clearer decision boundary, and less overlap among attack classes. The results reveal that this method can balance performance and interpretability for IoT security applications.

(Narayan et al., 2023) offer an approach to IoT environments using the CICIoT2023 dataset for an Intelligent Intrusion Detection System (IIDS). Combining the feature selection methods such as Correlation based Feature selection (CFS) and Importance ranking method (IRM) along with class balancing methods like Random over Sampling (ROS) and Balanced Random Forest Classifier (BRFC) will improve the detection performance of minority classes. When using the RF for application, the method improves the F1-score of under-represented classes by 7.9% in comparison to a significant base model. The research shows that intrusion detection can be enhanced using various techniques based upon feature selection and class balancing on imbalance IoT datasets which helps to effectively improve detection rate for attacks like Dictionary Brute Force, Browser Hijacking and in particular SQL Injection.

Various ML algorithms such as Naive Bayes, DT, RF, QDA, AdaBoost, MLP, and KNN were used in (Altrad, 2023) to classify IoT devices and detect attacks using the CICIoT2023 dataset. This study combines feature selection techniques for class balancing to improve detection performance. The outcomes indicate that DT and KNN obtained the best performance of about F-measure 0.979 and 0.973. The aim of this paper is to offer important insights which can be utilized to build effective security and protection analytics solutions for IoT environments.

(Phan et al., 2024) analyses the performance of 5 feature selection algorithms, that is, IG, Recursive Feature Elimination (RFE), RF, LR, and XGBoost together with 5 different ML models DT, RF, KNN, GB, and MLP to detect IoT Attacks using CICIoT2023 dataset. The findings reveal that RFE yields the highest accuracy in all models and feature selection schemes. The maximum accuracy of RFE is 99.57% with Random Forest using 30 features (RF+30f). The top-performing IoT attack classification method is identified as the combination of RFE and KNN using five features. This combination is identified for IoT attack classification based on getting good accuracy with fast time.

2.5 CICAPTIIoT2024 Datasets Characteristic and Related Works

The systems that comprise IIoT is defined as integration of ICS with enterprise systems and Internet, finally enabling digital transformation. The main features of IIoT systems include:

1. The performance of industrial and manufacturing systems is boosted through smart sensors, actuators and connected devices.
2. Connecting devices to networks to collect, share, and analyze data more efficiently.
3. Using real-time information and smart machines to help businesses make quicker decisions.
4. We are using these technologies for various sectors including manufacturing, energy, utility, oil and gas.

IIoT devices are applied in real life for predictive maintenance, improved field service, energy management, asset tracking and supply chain tracking. These devices are

used with almost every other industrial system for different purposes, which goes on to show that they are increasingly becoming important.

The importance and critical nature of IIoT devices in industrial operations make them very attractive targets for a wide range of cyber threats. This is not solely due to insufficient or immature security controls; it also results from the possible consequences of successful attacks on IIoT systems. The compromise of such systems could have far-reaching consequences for industrial operations and, potentially, national security.

The literature review shows that the most dangerous cyber attacks against IIoT devices are APTs. Advanced security threats can will escape or avoid detection. Due to their sophistication and stealthy character, they pose a significant challenge for traditional threat detection methods, leaving IIoT vulnerable to them. The CICAPTIIoT2024 dataset offers two types of data sources for detecting abnormal behavior caused by stealthy and persistent cyber-attacks: namely network logs and data provenance. The different sources of data have different levels of depth and viewpoints that enable malware detection.

Examining system logs and related system events with network logs and data provenance (Alam & Wang, 2021) shows that while the network logs and data provenance are important but have limited use for predicting abnormalities or differentiating malicious behavior in advanced cyberattacks. Table 2-1 compares various kinds of data sources.

Feature	Network Logs	System logs	System Events	Data Provenance
Purpose	Record network activity and communication	Capture process interactions with the operating system	Document system-wide activities and changes	Track origin, lineage, and history of data
Granularity	High-level network activity	Low-level process interactions	Mid-level system events	Detailed data flow and transformations
Accuracy	Relatively high	Very precise	Moderate	Can be highly accurate

Feature	Network Logs	System logs	System Events	Data Provenance
Security Relevance	Important for network security	Critical for process isolation	Useful for system-level threat detection	Essential for data integrity and trustworthiness
Use Cases	Network troubleshooting, detecting communication anomalies, DDoS, and unauthorized network access.	Essential for low-level behavioral analysis, detecting malware or rootkits, and sandboxing environments	System event correlation is useful for tracking specific application behavior or correlating high-level actions	Data trustworthiness, reproducibility, compliance, and understanding data lifecycle issues.
Strengths	Detect anomalies, unusual IPs or unauthorized connections	Captures stealthy behaviors like rootkits or malicious scripts that evade network detection	Provide a higher-level view and easier to parse than raw system logs	Tracks the lifecycle of data, helping identify unauthorized access and uncover lateral movement
Limitations	Encrypted communication can obscure malicious activity	High data volume makes real-time analysis challenging	May miss subtle or low-level actions performed by APTs.	High overhead for real-time tracking.
Reliability for APTs	Suitable for detecting external interactions, but insufficient alone to uncover internal activity	Effective for uncovering sophisticated and stealthy behaviors.	Suitable for detecting high-level behaviors, but less effective for stealthy or complex attacks	Very useful for detecting data theft and unauthorized changes but it depends on the implementation depth

Table 2-1: Feature Comparison Across Different Data Sources

We can conclude from the comparison that network logs and data provenance are more reliable for detecting APTs or traditional cyber threats due to their ability to capture detailed activities and trace stealthy activities within the system. This characteristic makes the CICAPTIIoT2024 dataset distinct from other captured datasets.

The network traffic in the CICAPTIIoT2024 includes IIoT-related features such as Constrained Application Protocol (CoAP) and Internet Group Management Protocol (IGMP). CoAP is a lightweight application protocol designed for resource-constrained devices in IoT environments operating on the User Datagram Protocol (UDP) and is commonly used for Machine-to-Machine (M2M) communication. IGMP is a network protocol device used to manage multicast group memberships, crucial for efficient communication in IIoT settings, especially when distributing the same data to multiple devices.

Although the SCADA system was part of the setup in the testbed environment, none of the SCADA-related communication protocols, such as Modbus or DNP3, were captured and included in the feature sets. These protocols allow SCADA (Jayasamraj, n.d.) systems to communicate between Master stations and remote terminals (PLC, Remote Terminal Units (RTU), Intelligent Electronic Devices (IED)), collect data, send commands, and monitor the status of industrial processes. Subsequent research could look at the elements of SCADA-related protocols that are crucial for the detection of APT, given the prominence of SCADA in industrial control.

(Priya et al., 2025) proposes a novel Weighted PCA-based Enhanced Deep Neural Network (WPCA_E-DNN) model for APT detection in IIoT environments. The model uses weighted PCA to extract the relevant features along with an improved Genetic Algorithm for tuning hyperparameters of the deep neural network. Testing of proposed methodology is done using CICAPTIIoT2024 dataset which outperforms contemporary methodologies by achieving an accuracy of 95.2%. The WPCA_E-DNN model could show effectiveness in detecting APT in IIoT systems.

(Narkedimilli et al., 2025) presents a lightweight and scalable solution that improves security in IoT networks. Their strategy integrates adaptive curriculum learning with explainable AI (xAI) techniques like LIME. The fundamental component of their system is a hybrid neural network that leverages GRU, LSTM, and self-attention layers supplemented with residual connections to capture temporal dependencies of IoT data. The model's accuracy and generalization are significantly enhanced through adaptive features pruning, staged learning and ensemble classification with RF and XGBoost. The

designed framework was evaluated on Edge-IIoT, CICAPTIIoT2024, and CIC-IoV-2024 datasets which gave an accuracy of over 97% with high precision and recall

2.6 Features and Gap Analysis for CICIoT2023 and CICAPTIIoT2024 Datasets and Related Works.

2.6.1 Datasets Features and Gap Analysis

The CICIoT2023 and CICAPTIIoT2024 datasets offer unique feature sets that are distinct from other developed datasets, although some features overlap. However, CICAPTIIoT2024 includes industrial-specific features such as CoAP and IGMP. The list of non-overlapping features that each dataset has is provided in Tables 2-2 and 2-3.

2.6.1.1 CICIoT2023 Distinct Features:

Feature Name	Description
LLC	Indicates if the link layer protocol is LLC

Table 2-2: Distinct Features for CICIoT2023

2.6.1.2 CICAPTIIoT2024 Distinct Features

Feature Name	Description
Source_Port	Port-related information for the source
Destination_Port	Port-related information about the destination.
Protocol_name	Protocol name instead of just the protocol type.
CoAP	A web transfer protocol designed for constrained devices and networks.
IGMP	IPv4 uses a network layer protocol for multicast.
DS_status	Likely related to the status of the Differentiated Services field in packets.
Fragments	It is related to fragmented packet handling.
Sequence_Number	The sequence tracking in protocols like TCP
Protocol_Version	The version of the protocol, such as IPv4 vs. IPv6.
flow_idle_time	Measures the time during which the flow is either idle or active.
flow_active_time	Measures the time during which the flow is either idle or active.
max_duration	The maximum value among all recorded durations
min_duration	The minimum value among all recorded durations.
sum_duration	The total sum of all durations in the dataset.
average_duration	The mean duration

std_duration	The standard deviation of durations

Table 2-3: Distinct Features for CICAPTIIoT2024

Chapter 4 highlights that certain features, including *source_port* and *destination_port* from Table 2-3, had a notable impact on the model's performance. This conclusion is based on their high importance scores derived from two separate feature selection techniques.

2.6.2 Datasets Related Work Summary

This section concisely describes and contextualizes the ML algorithms previously implemented on the CICIoT2023 and CICAPTIIoT2024 datasets. Table 2-4 delivers succinct insights into the applied feature selection techniques and pre-processing methods.

Study	ML Model	Feature Selection	preprocessing	Datasets
(Yan et al., 2024)	Dynamic Group Convolutional Networks	Autoencoders, Sliding window algorithm	OneHot coding, nearest neighbor imputation method	CICIoT2023, CICDDoS2019, NSL-KDD
(Patidar et al., 2024)	Power BI, Tableau, Excel	Hadoop Distributed File System (HDFS), HiveQL queries	NULL values removal,	CICIoT2023
(Yaras & Dener, 2024)	CNN, LSTM	Correlation method	Apache Spark	CICIoT2023, TON_IoT
(Huynh et al., 2023)	SVM, LR	Extracting relevant features from network traffic data using Wireshark, DPKT, or CICFlowMeter	Not specified	CICIoT2023, Real-time network traffic captured
(Pirtama et al., 2024)	RF	PCA	Not specified	CICIoT2023
(Okey et al., 2024)	CNN-GRU, LSTM-GRU	Not specified	Standardization	CICIoT2023, FLNET2023
(Kumar et al., 2024)	RF, DT, AdaBoost, SVM, LR	Not specified	Standardization	CICIoT2023

Study	ML Model	Feature Selection	preprocessing	Datasets
(Ramli & Thereza, 2023)	DT, KNN, RFt, Naïve Bayes	Chi-square algorithm	Data Cleansing	CICIoT2023
(Tseng et al., 2024)	DNN, CNN, RNN, LSTM, CNN + RNN, CNN + LSTM, Transformer	Not specified	Transformation, Standardization	CICIoT2023
(Gheni & Al-Yaseen, 2024)	MLP	Gaining-Sharing Knowledge (GSK), Autoencoders	Not specified	CICIoT2023
(Liu et al., 2024)	RMCNN	Not Specified	Not specified	CICIoT2023
(Khanday et al., 2024)	LSTM, 1D-CNN	Extra Tree classifier	Synthetic Minority Over-sampling Technique (SMOTE) , Standardization	CICIoT2023
(Modi, 2024)	XGBoost, RF, DT	Hybrid feature selection algorithm: Pearson & Spearman correlation coefficient, Kendall tau coefficient, IG	NULL values removal, Standardization, Transformation	CICIDS2017, CICIoT2023
(Becerra-Suarez et al., 2024)	DNN, LSTM, CNN	Extra Tree Classifier, features removal with only zero values	Data cleaning, Random sampling, Standardization	CICIoT2023
(Gupta et al., 2024)	LR, AdaBoost, GB, MLP, hybrid voting classifier (combining the above models)	Not specified	Minmax Scaler, Standard Scaler	CICIoT2023
(Le et al., 2023)	Ensemble learning technique: G B, RF, DT, Counterfactual explanations, LIME	Mean Decrease in Impurity (MDI)	Transformation	CICIoT2023, IoTID20
(Narayan et al., 2023)	RF	CFS, IRM	Standardization, ROS, BRFC	CICIoT2023
(Altrad, 2023)	Naive Bayes, DT, RF, QDA, AdaBoost, MLP, KNN	IG, Feature Weight	Data cleansing, Transformation	CICIoT2023
(Phan et al., 2024)	DT, RF, KNN, GB, MLP	IG, RFE, RF, LR, XGBoost	Standard scaling, Min-Max scaling, Transformation	CICIoT2023
(Priya et al., 2025)	WPCA_E-DNN	Weighted Principal Component Analysis (PCA)	Data cleansing, Standardization, Transformation, Data Sampling	CICAPTIIoT2024
(Narkedimilli et al., 2025)	Ensemble model (staged learning neural network model, RF, XGBoost)	Linear Discriminant Analysis (LDA), RF feature importance plots	Data Cleansing, Standardization	CICAPTIIoT2024, CIC-IoV-2024, EDGE-IIoT,

Table 2-4: Summary of CICIoT2023 and CICAPTIIoT2024 Related Works

2.7 Novel Approaches and New Contributions

2.7.1 Exploring New ML Algorithms Leveraging IIoT-Specific Features

This study aims to develop robust ML modules, distinct from those used in related work, that can effectively address key challenges in sparse datasets such as CICIoT2023 and CICAPTIIoT2024, particularly concerning the newly introduced features listed in Table 2-3. These models are designed to be tunable, allowing for control over both overfitting and underfitting, and to perform reliably when prediction and feature selection are required simultaneously, especially in complex datasets. Table 2-5 summarizes the candidate algorithms selected for this study, each chosen for its suitability to the data characteristics. While Elastic Net is less effective in managing non-linearity and class imbalance than other models, it has the advantage of integrating feature selection with prediction. Nonetheless, as analyzed in Chapter 4, a decline in performance was observed for Elastic Net relative to the other evaluated algorithms (Hui, 2023).

Algorithms	Category
Categorical Boosting (CatBoost)	Boosting Algorithms
Extremely Randomized Trees (Extra Trees)	Bagging-based Ensemble methods
XGBoost	Boosting Algorithms
Light Gradient Boosting Machine (LightGBM)	Boosting Algorithms
Elastic Net	Regularized Regression Models

Table 2-5: Selected Algorithms for Machine Learning

2.7.2 Evaluation of Module Performance Based on Selected Features

This study evaluates how the developed models perform when trained on different subsets of selected features from each dataset. The comparison highlights how various feature selection methods, distinct from those commonly used in related work, affect model behavior and accuracy. Table 2-6 outlines the selected feature selection techniques representing the two main categories: Filter Methods and Wrapper Methods (Jha, 2024).

Technique	Algorithm
Recursive Feature Elimination with Cross-Validation (RFECV)	Wrapper Method
Mutual Information (MI)	Filter Method

Table 2-6: Selected Feature Selection Techniques

2.7.3 Utilize SMOTE to Address Class Imbalance

This study adopts a systematic approach to understanding how balanced or imbalanced the target variable is, a factor often overlooked in related literature review studies. By analyzing entropy scores, we apply SMOTE techniques to correct any noticeable skewness in the class distribution, ensuring no single class dominates the dataset.

2.7.4 Explainable AI with Partial Dependence Plots:

This study employs less common xAI techniques, such as Partial Dependency Plots (PDPs), to better explain model predictions and understand which features truly matter. While PDPs are often used to interpret feature importance, they also help illustrate how individual features influence the target variable, providing clearer insight into the model's behavior.

2.8 Summary and Conclusion

Cybersecurity has faced numerous challenges recently, particularly with the increasing reliance on IoT and IIoT devices. Our research study aimed to explore these challenges and ascertain solutions to enhance the security of these interconnected systems.

Our literature review emphasized several crucial aspects of IoT and IIoT cybersecurity. The impact and consequences of cyberattacks against electric control devices and IoT/IIoT devices have been extensively documented in academic literature. It

is proven that addressing cyber threats, especially those considered advanced, such as APTs, using traditional methods is deemed insufficient and inadequate.

Unlike enterprise applications, where the primary security focus is on data protection, industrial systems prioritize the security of the process itself. SCADA-based instrumentation and alarms are commonly employed to control and monitor processes in industrial settings. These findings underscore the unique challenges posed by IoT and IIoT devices in industrial contexts and highlight the need for specialized security measures.

The review of existing research identified several key trends. Traditional security methods have been largely ineffective against the sophisticated cyber threats in IoT and IIoT environments. The traditional focus on data-centric security in enterprise applications does not translate well to the process-oriented nature of industrial systems.

The prevalence of SCADA systems in industrial settings presents both opportunities and vulnerabilities in the context of cybersecurity. These observations suggest that a paradigm shift is necessary in approaching cybersecurity for IoT and IIoT devices, particularly in industrial settings.

Recognizing the limitations of traditional security methods, the research aims to develop a novel solution. It proposes building a lightweight, robust ML module tailored explicitly to IIoT-related features. To optimize model performance, we plan to utilize the CICIoT2023 and CICAPTIIoT2024 datasets in conjunction with optimal feature selection techniques.

Additional data preprocessing methods will be employed to prepare the data for ML training and evaluation. The efficacy of the developed ML models will be assessed against a predefined performance matrix. This approach aims to harness the power of ML to detect malicious behavior within IoT and IIoT devices, potentially providing a more effective solution against evolving cyber threats.

Chapter 3—Methodology

3.1 Introduction

This chapter describes the methodology employed to answer the research questions and test the validity of the hypotheses stated in Chapter 1 associated with anomaly detection in network traffic exchanged among industrial components of electrical devices. Industrial protocols, including Supervisory Control and Data Acquisition (SCADA), exhibit unique behavioral patterns that largely deviate from those of general-purpose network protocols; hence, their deviations from normality are easily detectable, which is not always the case in Advanced Persistent Threat (APT)-based attacks. Expected behavior is baselined into three categories: 1) Historical Baseline, 2) Fixed functional patterns, peer baseline, similar functionality across devices; and 3) Partner Baseline, consistent communication feedback (Zhou et al., 2018).

Two enriched datasets, CICIoT2023 and CICAPTIIoT2024, are exploited to establish a strong behavioral baseline with attributes essential for industrial domains. While the datasets include several types of attacks, this work aims to develop a high-accuracy binary classification model that classifies all attacks into two distinct classes, Table 3-1.

Binary Classes	Description
Class 0	Normal Traffic (Benign)
Class 1	Attack (Malicious)

Table 3-1: Binay Classification Details

Through the Pegasus high-performance computing (HPC) system, a framework design process was optimized for the access and processing of large datasets in Python. Initially, the data must undergo a comprehensive process of cleaning, transformation, and

scaling. Additionally, we employ methods for selecting features and combining ML algorithms in both datasets for better accuracy and relevance in the model.

The performance and evaluation details of these modules along with the feature selection for model enhancement will be illustrated in Chapter 4.

3.2 Datasets Analysis

In the execution of this study, various datasets were carefully chosen to help produce quality results when developing Machine Learning (ML) models to distinguish the difference between malicious and benign behavior.

The Canadian Institute for Cybersecurity (CIC) has performed two different research projects to gather information from real and simulated IoT and IIoT devices. According to (Ghiasvand et al. (2024); Neto et al.,2023), the datasets produced by these studies comprise millions of records that truly reflect the realities and dynamics of actual attack scenarios, as well as the log data of various categories of cyberattack, in Comma-Separated Values (CSV) file format.

Although the CICAPTIIoT2024 has more features specifically related to industrial IoT devices, there are major overlaps in features in the two datasets. This overlap allows the evaluation of any performance benefit from the additional IIoT-related features in comparison to a more traditional network traffic-based dataset.

The complementary nature and overlapping features of these datasets provide a robust foundation for a comparative analysis of machine learning performance in detecting malicious behavior, particularly for understanding the benefits of incorporating IIoT-specific features.

The following two sections provide insights into the datasets and the distribution of the attack methods conducted against the IoT and IIoT devices.

3.2.1 CICIoT2023 dataset

Data from 105 diverse IoT devices, which mimicked a real-world IoT topology, were collected during multiple attack scenarios. Although the total duration of the entire test bed operation is not specified, 16 hours were dedicated to capturing benign data generated to establish a baseline for normal behavior.

During the active data collection period, 33 malicious attack types were executed in the IoT topology. Figure 3-1 illustrates the attack classification into seven primary categories: Denial of Service (DoS), Distributed Denial of Service (DDoS), reconnaissance, web-based, brute force, spoofing, and Mirai. Among these, DDoS attacks were the most prevalent, with over 30 million instances, while brute force attacks had the lowest occurrence.

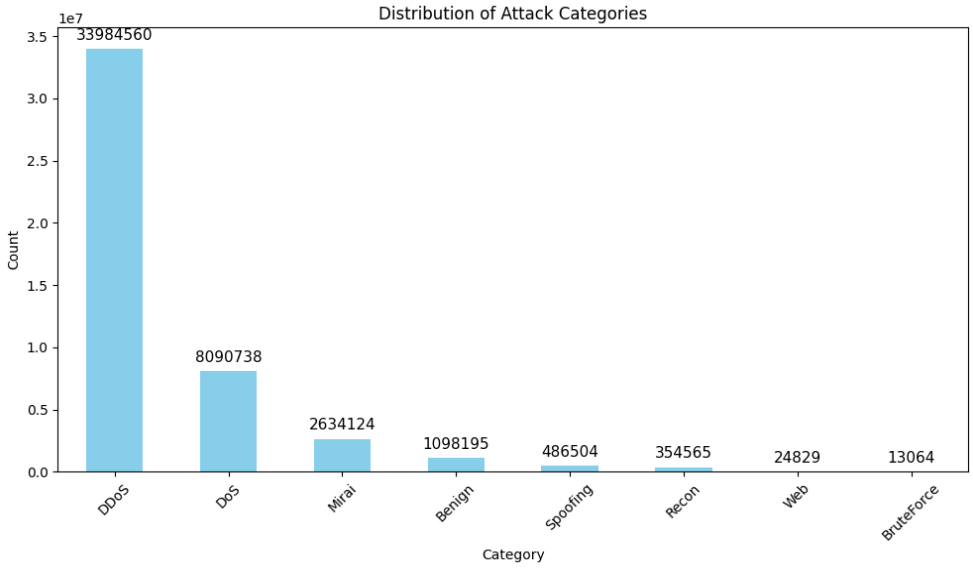

Figure 3-1: Attack Categories Distribution for CICIoT2023

This research focuses on collectively analyzing all attack types rather than individual classifications. As shown in Figure 3-2, all malicious attack categories were consolidated under a single label, "attack."

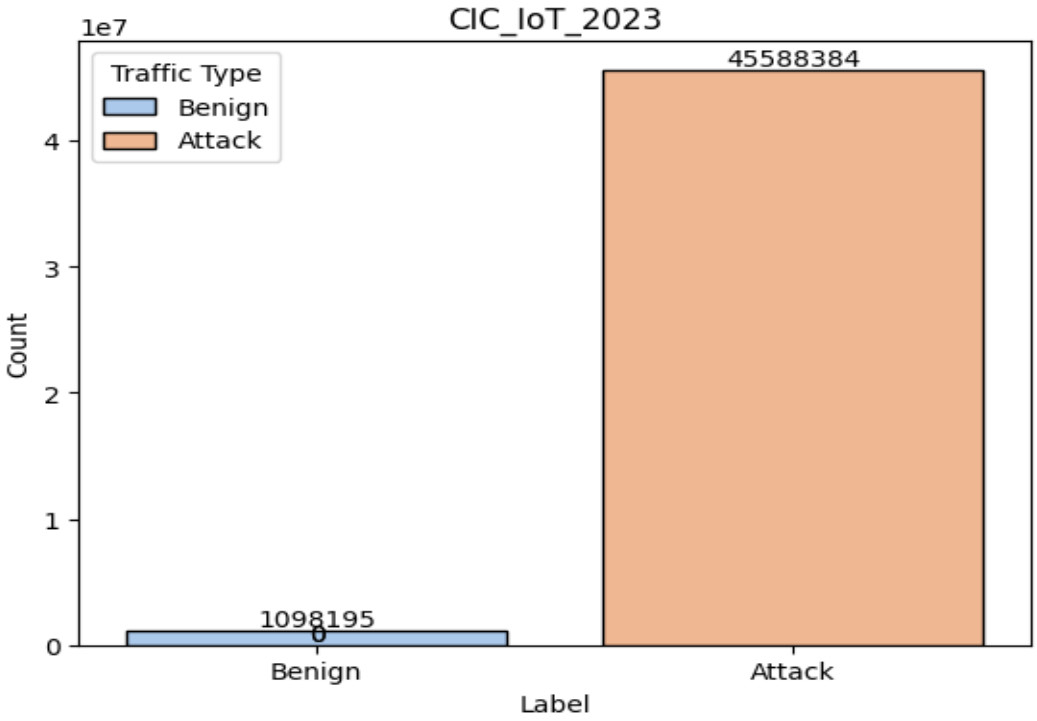

Figure 3-2: Traffic Labeled Distribution for CICIoT2023

All network traffic was captured using Wireshark during the attack duration and stored in Packet Capture (PCAP) format. The data processing phase converted the captured traffic into separate CSV files using the DPKT library, which resulted in a file size of 13 GB having 45,000,000 records. Forty-seven features were extracted from the captured network traffic to represent system behavior during the experiments.

3.2.2 CICAPTIIoT2024 dataset

Advanced Persistent Threats (APTs) are sophisticated attacks designed to harm targeted systems covertly over extended periods. Due to the inherent nature of Industrial

Internet of Things (IIoT) devices and their interconnectivity with other critical infrastructure components, these devices have become prime targets for APT attacks.

The primary goal of this dataset is to establish a robust behavioral foundation by collecting network logs during APT activities within a developed testbed. Despite the availability of network logs, some sophisticated APT activities remain undetected and indistinguishable from normal behavior.

What sets the CICAPTIIoT2024 dataset apart is its data provenance, which provides insights into system behavior and interactions among system entities. Two distinct datasets were captured during a customized attack that emulates the APT29 adversary. The datasets are stored in separate folders corresponding to two phases of the attack, which lasted 96 hours and 72 hours, respectively.

The provenance data is stored in CSV format, while network log files are available in CSV and PCAP formats. Although the data provenance files are highly valuable for detecting abnormalities in the presence of APT attacks, this study focuses exclusively on network log files, leaving provenance data analysis for future research.

The network log dataset comprises 70 features, ranging from network protocols used by IIoT devices to those employed in enterprise applications. The two CSV files, which contain approximately 21,599,219 records, total around 9.2 GB in size.

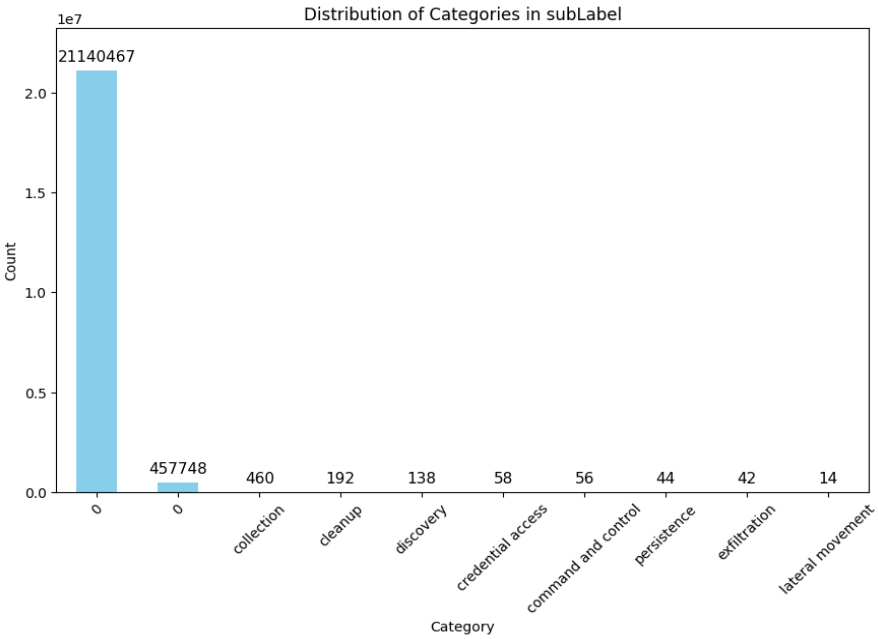

Figure 3-3: Tactic Group Names for CICAPTIIoT2024

During the two phases, APT attacks were categorized into eight tactic groups: Collection, Exfiltration, Command and Control (C2), Persistence, Discovery, Credential Access, Lateral Movement, and Defense Evasion. The dataset's sublabel column reflects these tactic group names as shown in Figure 3-3.

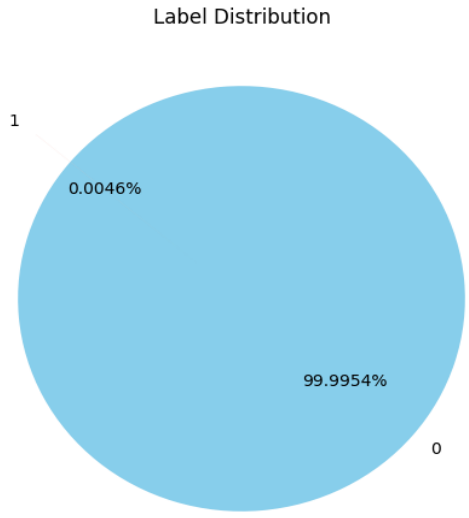

Figure 3-4: Binary Label Distribution for CICAPTIIoT2024

Although the dataset includes eight attack categories, Figure 3-4 illustrates a significant imbalance in label distribution between benign and attack-labeled records. As shown in Figure 3-5, the Label column classifies all records into two categories: benign and attack.

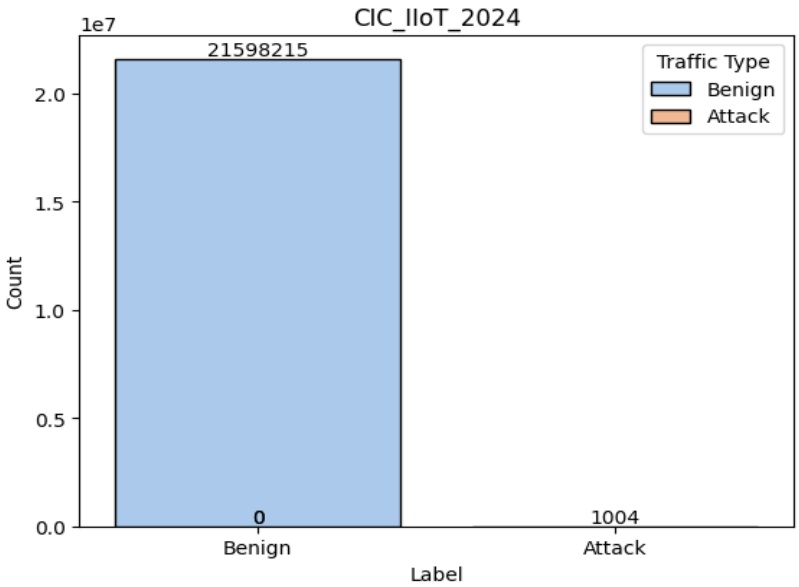

Figure 3-5: Binary Label Counts for CICAPTIIoT2024

3.3 Datasets Preprocessing

Data preprocessing will be comprehensively conducted using Python, leveraging well-established libraries designed for data cleaning and preparation. Key Python libraries, such as Pandas and NumPy, are recognized for their efficacy in data analysis and visualization, providing essential tools for achieving clarity and insight into complex datasets. The preprocessing workflow goes through several stages, during when the data undergoes various treatments to remove duplicates and unnecessary entries. While being a fundamental step involved in the successful working of ML processes, it is a challenging task in case of a large scale dataset.

In order to simplify handling big data, HPC environment with proper configuration is often used to perform parallel computing. Hadoop and Spark can be used to speed up processing and performance. Exploratory Data Analysis (EDA) stage will be done before starting the preprocessing stages. The purpose of this initial assessment is to understand the quality of the data, suitable preprocessing techniques, and suitable feature selection procedures. The phase where data is explored helps yield valuable information backed by visuals that aid in selecting the right ML algorithms for the given datasets.

As depicted in Figure 3-6, the dataset undergoes several preprocessing steps to prepare it for use in an ML model training.

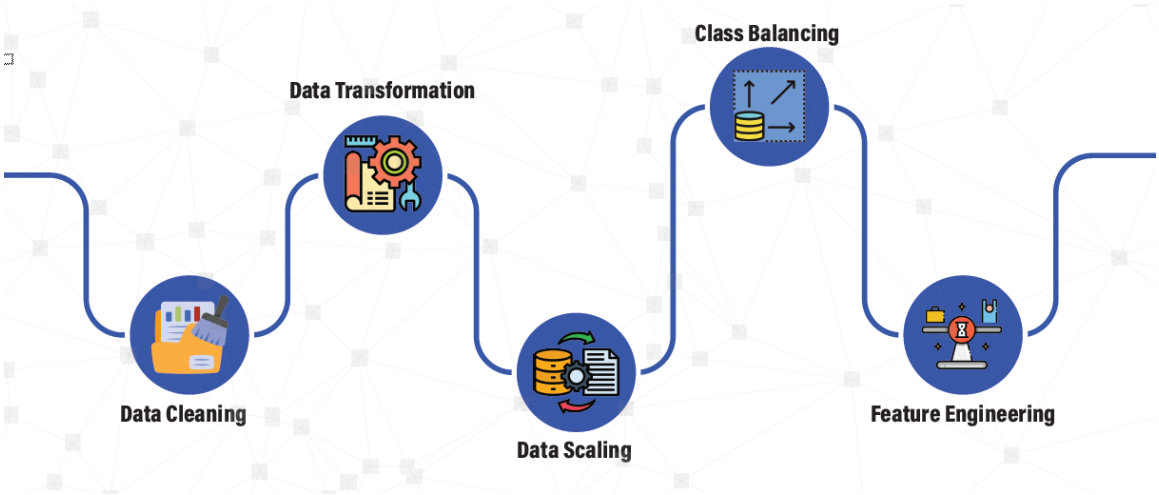

Figure 3-6: Data Preprocessing Stages

3.3.1 Data Cleaning

The steps of the data cleaning process, shown in Figure 3-7, help to assess the quality and integrity of the dataset which is useful for meaningful and effective analysis. The first step is to evaluate the influence of missing information on analysis and the choice of handling method. After checking, both datasets are found no null records,

meaning all the entries are filled up and updated (“Handling Missing Values”, 2024). Without the missing data, the dataset’s quality is maintained, which in turn helps in its analysis and modeling.

Duplicate removal is also a crucial step. This is particularly important in ML training where large data sets are involved. An excessive number of duplicate records takes up unnecessary computation resources and does not help boost the model's performance. Removing similar records will greatly enhance the performance and efficiency of ML models.

To prevent data sparsity and redundant feature columns, irrelevant features that might perplex the model and cause inferior performance were identified and removed from the datasets. Additionally, features with only one unique value were removed because they did not contribute to the model's predictive power. Elimination of these features decreases the dimensionality of the datasets, enhances model interpretability, reduces computational expenses, and accelerates training.

Another essential aspect of data cleaning is dealing with outliers. Outliers may have a big impact on the performance of ML models and the inferences from the data. However, for dealing with network traffic logs in big data environments, outliers may have to be retained. Outliers here may refer to unusual traffic that is relevant for malicious activity identification (Ianni & Masciari, 2023). The model, once trained, can make use of such anomalies to detect and analyze threats effectively despite the malicious entries being much less common than regular ones.

Finally, correcting inconsistent data formats is a good practice that results in a cleaner and more concise overall dataset. For instance, representing floating-point numbers with zero fractional parts (e.g., 10.0) as integers decreases the overall dataset size and improves computation efficiency by minimizing memory usage. These optimizations not only simplify data processing but also improve the overall ML pipeline.

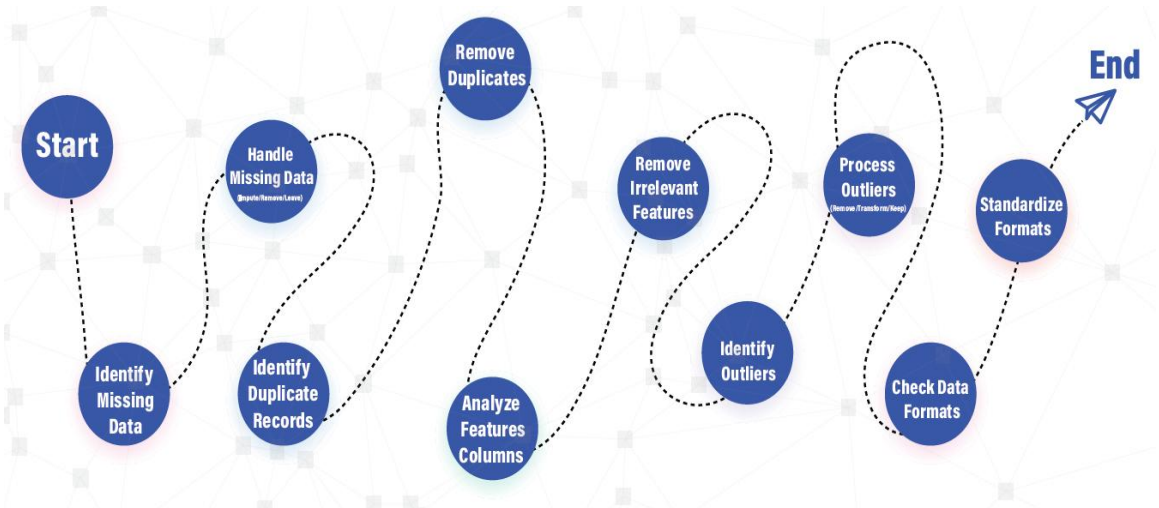

Figure 3-7: Data Validation and Cleanup Workflow

3.3.2 Data Transformation

Converting categorical data into numerical values is essential to preparing data for ML. Most ML algorithms are built to handle numerical inputs because they allow for the efficient execution of mathematical operations and optimizations. By representing categorical features as numbers, the data becomes clearer for algorithms to process, ensuring accurate and efficient model training.

The categorical features were already converted to numeric codes and there were no further required changes for those columns. This was time-saving and allowed for more attention to be paid to other areas of the data preparation process. On the other

hand, the label column which defines record as "malicious" or "benign" stores them in string format. Numerical inputs are necessary for most ML algorithms, so the label must be converted into numerical form before being used in training.

To address this, the label column was subjected to a binary encoding scheme. The categorical variables were transformed into binary; 0 for one category and 1 for the other category. With this, the label feature was prepared in the right form to train the machine learning model easily along with the other preprocessed features.

3.3.3 Data Standardization

There is no need for standardization of the ML algorithms utilized in this work. This was, however, helpful as our datasets have a high range of scales of numerical features. We could make sure that no feature will dominate the training process due to scale (Punyakeerthi, 2024).

A key benefit of standardization is that it ensures that no single feature has a large influence on the model due to scale. This process makes all features contribute equally during the training process itself, resulting in a powerful process. To do this we used the StandardScaler from Scikit-learn (Pedregosa et al., 2011). This was an appropriate choice for our datasets since they were not normally distributed. StandardScaler works efficiently on varied data unlike other scaling methods which rely on assumptions of normality.

The data size increased as a result of this standardization process. To solve the problem temporarily, we altered data-type to reduce size of the dataset by changing all float64 and int64 into float32 and int32, which take 4 bytes less memory.

This optimization improved memory efficiency noticeably and contributed to faster module execution. By reducing computational overhead, the preprocessing steps became more manageable and streamlined, ensuring the data preparation process remained efficient and scalable.

3.3.4 Class Imbalance Treatment

It is a big challenge in ML to deal with imbalanced data. Overlooking such an issue tends to produce poor prediction accuracy and bias towards the most dominant classes in a model. Therefore, handling a balanced dataset is crucial in developing practical and reliable models.

Merely attempting to balance and not selecting an effective balancing technique can have a negative impact, including generating incorrect output and undermining model performance. As such, balancing techniques must be carefully considered to maintain data integrity and generate correct output during development.

One helpful evaluation for checking for class distribution is calculating entropy, a measure of uncertainty or randomness in a dataset (SciPy API, 2008). A zero value for entropy signifies a dominant class, and balancing techniques must be adopted to enhance model performance. A higher value for entropy signifies a well-class distribution, and no balancing will have to be adopted in such cases.

Along with the data visualization, the entropy calculations for both datasets revealed values close to zero, indicating that one class significantly overshadowed the other. Therefore, this study relies on the data-level techniques during the data preprocessing phase to balance the datasets externally without relying on tuning the ML

module parameters. This will enable it to be combined with any ML algorithm later. However, the use of imbalanced learning as an ML module will be discussed further in the following sections.

Among external data balancing techniques (*7-Techniques-to-Handle-Class-Imbalance-in-Machine-Learning-Eb1297419ec9*, n.d.), the Synthetic Minority Over-sampling Technique (SMOTE) is often considered effective for handling significantly imbalanced datasets and less effective for highly dimensional datasets (SMOTE — Version 0.13.0, n.d.). SMOTE is a sophisticated algorithm used to address class imbalance by generating synthetic samples of the minority class, thereby preserving the characteristic patterns of the original data.

The entropy calculations were conducted again following the implementation of the SMOTE technique. As expected, the results were close to one. This outcome, as shown in Figure 3-8, indicates that SMOTE successfully balanced the data by increasing class diversity. By generating synthetic data from the minority class, the overall class distribution was more uniform, reducing the dominance of any single class.

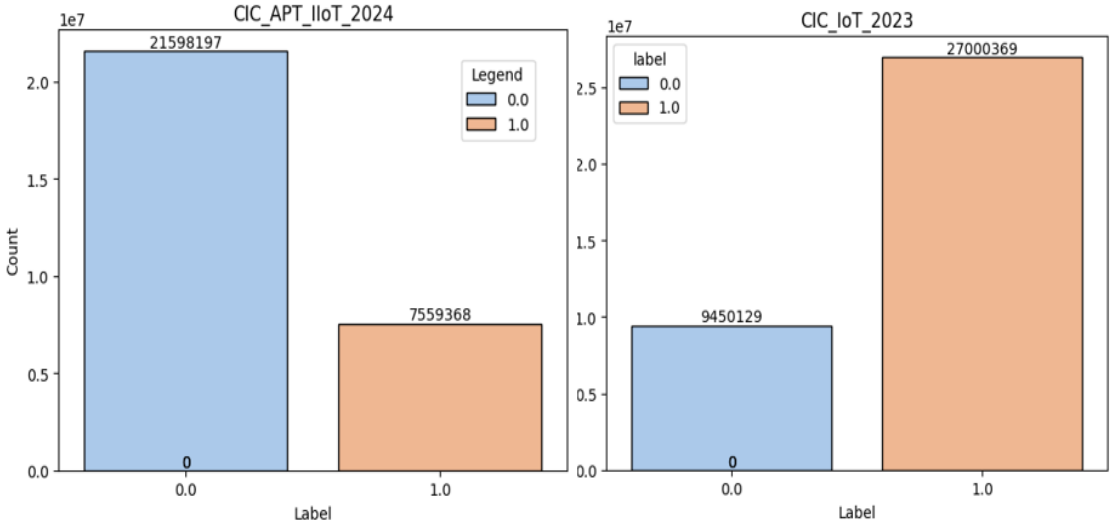

Figure 3-8: Balanced Datasets Distribution

While selecting SMOTE is adequate for handling significant imbalanced datasets, Future studies should explore SMOTE variant techniques such as SMOTE for Nominal and Continuous (SMOTE-NC), k-Means SMOTE, and others to assess their effects on model performance and further optimize the learning process (Bordia, 2022).

3.3.5 Feature Engineering

Although the selected ML algorithms are resilient regarding such dataset characteristics and already have feature importance functionality incorporated, trying distinct feature selection approaches in preprocessing is still unavoidable. This guarantees both optimal model performance and avoids unwarranted model complexity.

Different feature selection methods offer unique advantages and trade-offs. For instance, filter methods are computationally efficient but may overlook meaningful feature interactions ("Top 12 Dimensionality Reduction Techniques for Machine Learning," 2024). In contrast, while capable of capturing complex relationships, wrapper methods tend to be computationally expensive. Additionally, some techniques scale better for high-dimensional data than others (Brownlee, 2020).

Based on these factors, Recursive Feature Elimination with Cross-Validation (RFECV) and Mutual Information (MI) were selected. These methods are particularly effective for non-normal or nonlinear data, high-dimensional datasets, and scenarios where feature interactions play a critical role in prediction.

Before applying feature selection, we need to explore whether feature selection needs to be done before or after data splitting (“Should Feature Selection Be Done before Train-Test Split or After?”, 2019). Figure 3-9 demonstrates that the standard and

suggested method is to undertake feature selection after data splitting. Feature selection before data splitting could lead to information leakage from the test set, contrary to treating the test set as unseen data. The process of feature selection-first could result in optimistic performance and an overfitting module.

Given the large dataset and the need to handle numerous features while preventing data leakage, the split-first approach was adopted for both RFECV and MI. This ensures a robust and unbiased feature selection process, improving model reliability.

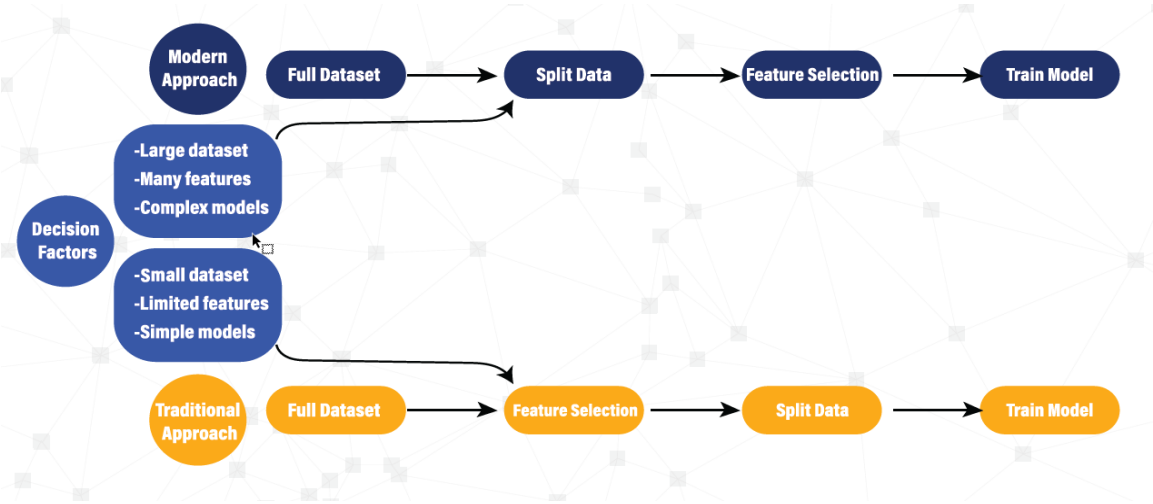

Figure 3-9: Split-First Approach Workflow

3.4 Machine Learning Algorithms and Classifier Modules.

Choosing the most appropriate ML algorithm is a steppingstone to optimal performance and accuracy in well-trained models. A classifier's performance highly relies on how well the algorithm matches the nature of the dataset. Knowledge of data shape and key features is vital in determining the most appropriate ML candidates. The choice ultimately results in building an enhanced classification model.

There is a need to verify data distribution to test for normality. The Anderson-Darling test, which Minitab supports, indicates if the data is normally distributed (The Anderson-Darling Statistic, n.d.). Upon applying such a test, all data points indicated substantial deviation from normality with an exponential distribution. The conclusion was established because a significant Anderson-Darling statistic confirms the dataset lacks a normal distribution.

Next, to evaluate the linearity between the predictors and the target variable, a Fit Regression Model was performed in Minitab using the $\boldsymbol{R^2}$ formula (Overview for Fit Regression Model and Linear Regression, n.d.). As illustrated in Table 3-2, the resulting $\boldsymbol{R^2}$ value was 0.01%, indicating that the model accounts for virtually none of the variability in the target variable. This suggests that the relationship between the predictors and the target classes is nonlinear, further reinforcing the need for ML algorithms capable of handling nonlinearity.

$$\boldsymbol{R^2} = 1 - \frac{\text{Sum of Squares of Residuals } (SS_{res})}{\text{Total Sum of Squares } (SS_{tot})}$$

Symbol	Value	Meaning
$\boldsymbol{S}$	0.0068151	Measures the average distance between observed values and the regression line
$\boldsymbol{R^2}$	0.01%	Represents the proportion of variance in the dependent variable explained by the model
$\boldsymbol{R^2_{adj}}$	0.01%	Adjust the R-squared for the number of predictors
$\boldsymbol{R^2_{pred}}$	0.01%	Estimate how well the model will predict new data

Table 3-2: Fit Regression Model Summary Result

Although SMOTE was applied during preprocessing to address the class imbalance, additional considerations were made when selecting ML algorithms. Since imbalanced-learn is incorporated as an ML module in this study, the chosen algorithms

should provide parameters such as class weights or built-in imbalance-handling capabilities. Moreover, built-in feature importance was also considered during the selection process. These mechanisms offer a fast, easy, and interpretable way to understand feature contributions. However, the built-in feature importance might have some limitations. Therefore, MI and RFECV were also used to make them more reliable.

The summary of the selected ML algorithms is presented in Table 3-3, providing a comparative analysis of their suitability based on the dataset characteristics and preprocessing requirements.

Algorithm	Assumes Normality	Handles Nonlinearity	Handles Imbalanced	Built-In Feature Importance
ExtraTrees	**No** *(Geurts et al., 2006)*	**Yes** *(Hastie et al., 2009)*	**Yes** *(Tunnicliffe, 2021)*	**Yes** *(ExtraTreesClassifier, n.d.)*
XGBoost	**No** *(Tyagi, 2018)*	**Yes** (Tyagi, 2018)	**Yes** *(XGBoost Parameters — Xgboost 3.1.0-Dev Documentation, n.d.)*	**Yes** *(XGBoost Parameters — Xgboost 3.1.0-Dev Documentation, n.d.)*
LightGBM	**No** *(Parameters — LightGBM 4.6.0.99 Documentation, n.d.)*	**Yes** *(LightGBM Feature Parameters, 2023)*	**Yes** *(LightGBM Feature Parameters, 2023)*	**Yes** *(Parameters — LightGBM 4.6.0.99 Documentation, n.d.)*
CatBoost	**No** *(CatBoost in Machine Learning, 2021)*	**Yes** *(Hancock & Khoshgoftaar, n.d.)*	**Yes** *(Common Parameters, n.d.)*	**Yes** *(Common Parameters, n.d.)*
ElasticNet	**No** (large sample size) *(Hastie et al., 2009)*	**Partially** (linear only) *(Hastie et al., 2009)*	**Yes** (using class weighting) *(ElasticNet, n.d.)*	**Yes** (non-zero coefficients) *(ElasticNet, n.d.)*

Table 3-3: Proposed Machine Learning Algorithms

3.5 Conclusion

To sum up, the methodology described in this chapter illustrates a systematic approach for detecting anomalies from network traffic of industrial devices using

sophisticated ML techniques. Utilizing the enriched CICIoT2023 and CICAPTIIoT2024 datasets, we establish robust baselines that take into account the particularities of relevant industrial protocols, as illustrated in Table 2-3, to distinguish between malicious and benign traffic effectively. For this study, ML models were chosen due to their appropriateness for binary classification, interpretability, and performance in a similar context. Algorithms such as ExtraTrees, XGBoost, LightGBM, and CatBoost were chosen because they can effectively handle imbalanced datasets and non-linear relationships in the feature space. These models are able to glean the highly complex interrelationships that exist in data. That is why they can detect the minute deviations that might point towards a cyber-attack. Conversely, Elastic Net was considered as a candidate but was ultimately deemed less effective for this particular application. Though suggesting features and handling multicollinearity, Elastic Net often fails at complex non-linear relationships present in the data. The important information that sophisticated models can exploit may be lost when relying on linear combinations of features. The diverse types of attacks and network behavior in the IIoT environment significantly limit the capability of Elastic Net; thus, the Elastic Net performance is expected to deviate from the other selected ML algorithms. As we move into Chapter 4, the comparative analysis of the chosen ML models will be discussed. This analysis will focus on performance metrics, accuracy, and effectiveness in detecting malicious network traffic. This analysis will provide insights into the strengths and weaknesses of each model, guiding future research and implementation strategies for enhancing cybersecurity in industrial environments.

Chapter 4—Results and Analysis

4.1 Introduction

In this chapter, we will examine the practical use of the adapted Machine Learning (ML) algorithms in detail, understand and exploit model performance. By evaluating the outcomes of these algorithms, the chapter aims at learning what can be inferred from the behavior of each model on the selected datasets.

It is important to understand how ML algorithms learn during the training phase. For this purpose, the results of the models will be contrasted against two benchmarks. One that comes from the results of the previous literature reviewed, and one constructed using the results from the training that uses the two selected datasets, that is, CICIoT2023 and CICATPIIoT2024.

One of the primary objectives of this chapter is to test the hypotheses using the findings of the analysis. So, this will be achieved through clearly answering the research questions and laying a logical foundation based on empirical findings.

In addition, the tools and techniques introduced in Chapter 3 will be revisited during the results analysis. This step is introduced to identify any potential gaps in methodology or to explore alternative approaches that could yield improved outcomes. The effectiveness of the methods employed ensures the robustness and validity of the study.

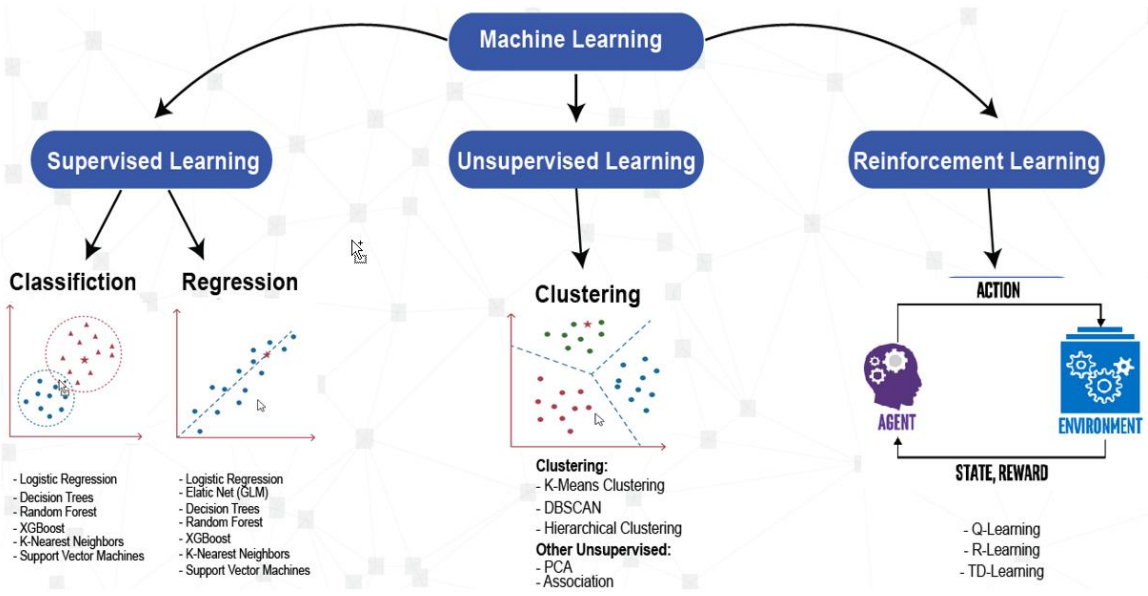

Figure 4-1: Types of Machine Learning

The ML algorithms selected for this study, categorized in Figure 4-1, fall within three subgroups of Supervised Learning: boosting models, regularized regression models, and bagging-based ensemble models. As this chapter gets into the minutest details of the results, it will also discuss why these models were selected. Special attention will be paid to how the characteristics of the datasets influenced model choice and why specific prediction modules achieved notably high accuracy when compared to observed outcomes.

4.2 Reviewing and Analyzing Research Questions and Hypotheses

This chapter provides the basis for answering the proposed research questions by describing the outcome of training ML models and reviewing the corresponding evaluation metrics. The conclusions drawn from these experiments not only justify the research approach but also determine if the proposed hypotheses must be rejected. All training models have been developed using appropriately selected datasets, supported by

extensive theoretical rationale and empirical validation, to ensure results are scientifically valid and practically meaningful.

To begin, we focus on the second research question, which investigates whether incorporating features extracted specifically from Industrial Internet of Things (IIoT) devices, particularly those related to lightweight, resource-constrained communication protocols, can significantly improve the ability of ML models to detect anomalous behavior. This investigation goes beyond existing literature by applying algorithms that have been rarely, if ever, studied in this context. To support this exploration, Table 4-1 presents a comparative analysis that distinguishes between algorithms previously adopted in related studies and those newly applied in this research. The goal is to evaluate whether IIoT-specific features, as outlined in Table 2-3, provide a measurable performance advantage in terms of accuracy and generalization when compared to the results reported in previous studies.

Algorithms	**Category**	**Previous Adaptation**
CatBoost	Gradient Boosting Algorithms	No
Extremely Randomized Trees (Extra Trees)	Bagging-based ensemble methods	No
XGBoost	Gradient Boosting Algorithms	Yes
LightGBM	Gradient Boosting Algorithms	No
Elastic Net	Regularized Regression Models	No

Table 4-1: Adaptation of ML Algorithms in Prior Studies

Closely related to the second question, the first research question explores performance variations when the same ML models are applied to two different types of datasets: 1) a general-purpose Internet of Things (IoT) dataset (CICIoT2023), and 2) an IIoT-specific dataset (CICAPTIIoT2024). This comparative analysis is critical for understanding whether the inclusion of industry-oriented features yields superior

detection capabilities. Although models like EXtreme Gradient Boosting (XGBoost) have been employed in previous work, this study introduces Categorical Boosting (CatBoost) and Light Gradient Boosting Machine (LightGBM), two related gradient boosting frameworks, to provide a more comprehensive evaluation. By assessing how both datasets perform using these models, we add to the credibility of the results and find out if certain characteristics are more helpful in detection.

The third research question seeks to determine the effectiveness of the feature selection techniques such as Mutual Information (MI) and Recursive Feature Elimination with Cross-Validation (RFECV) in improving performance metrics of the ML model. The techniques help in lowering the dimensionality which immensely reduces computations and yet keeps useful information which is necessary to obtain an accurate prediction. The goal of adding them is to demonstrate that a more strategic choice of features can yield enhanced model efficiency and a lower false positive rate, which is critical in real-world industrial systems where high accuracy is crucial.

4.3 Preprocessing and Data Balancing Outcomes

Preparing data for a ML model is both a time-intensive and crucial part of the development process. This task involves a sequence of interdependent steps, where the output of one phase becomes the input for the next. If data quality at any stage is compromised, the resulting impact can propagate through the pipeline, often leading to inconsistent or unreliable results in the final model. Data preparation must be done accurately and carefully due to this dependence.

To maintain the quality of every preprocessing stage, each must be continually monitored, and each output checked against its input. It guides us in differentiating between real gains and unintended noise (or loss) introduced by changes made in that step. If we do not validate the data at each stage, it becomes difficult to verify whether the final dataset can predict the original data or not.

In this regard, five quantitative measures are used to measure the effectiveness and quality of each phase in the preprocessing pipeline, which include file size, memory usage, number of features, processing time, and information loss (Shah, 2025). By taking these measurements together, you get a real idea of how the data looks across the pipeline, and whether we should be concerned by efficiency, complexity, or data quality.

The size of a file is noted to see how much storage is used by each transformation, like cleaning or normalizing. When the size changes, it could be that some unnecessary information has been removed or some synthetic data has been added. Processing time reveals the duration of any operation to execute while selecting a distributed computing process to determine its scalability to bigger datasets. Memory usage gives an insight into the hardware requirements of the pipeline which becomes very important with large data.

Information loss, while more abstract, is arguably the most significant metric. It refers to the reduction or degradation of valuable characteristics in the data, which can negatively impact the learning process. To quantify this loss, concepts from information theory, namely Entropy and MI, are applied (Berg et al., 2025). Entropy quantifies how uncertain or variable a single feature is thereby making it suitable for observing changes during general preprocessing. MI, meanwhile, assesses how the features depend on the target variable. This metric matters in feature engineering when selecting features with

the most predictive power for the target. Using entropy to early clean the data would help to see whether any transformation causes increment or decrement in entropy. This can in turn signal whether some useful variability has been lost or preserved.

In this study, the same preprocessing steps were applied to both the CICIoT2023 and CICAPTIIoT2024 datasets. For efficiency, metrics are reported only for CICAPTIIoT2024, under the assumption that the results are representative of both datasets due to their shared pipeline structure.

The impact of preprocessing on system resources is summarized in Figure 4-2. One of the most notable improvements occurred when the dataset format was converted from Comma-Separated Values (CSV) to Parquet.

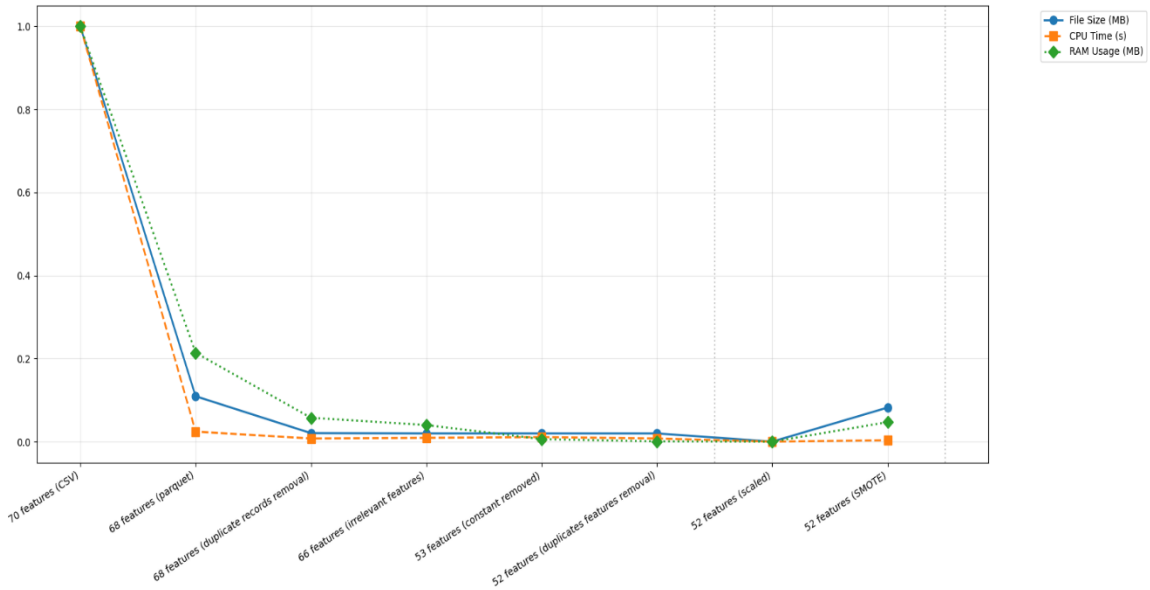

Figure 4-2: Preprocessing Pipeline Metrics

This change significantly reduced file size and improved overall resource utilization. Most of the subsequent preprocessing steps had little effect on processing time. However, a slight increase in memory usage was observed following the

implementation of Synthetic Minority Over-sampling Technique (SMOTE), a technique used to generate synthetic samples to address class imbalance.

Figure 4-3 depicts feature quality evolution. We found that feature quality improved significantly following the removal of a considerable number of constant-value features. This added entropy, meaning the features that were left have been more variable than previously. Higher Entropy is a good sign considering that cleaner data have more information. More information means that the learning capacity of ML models increases for better generalization and predictability.

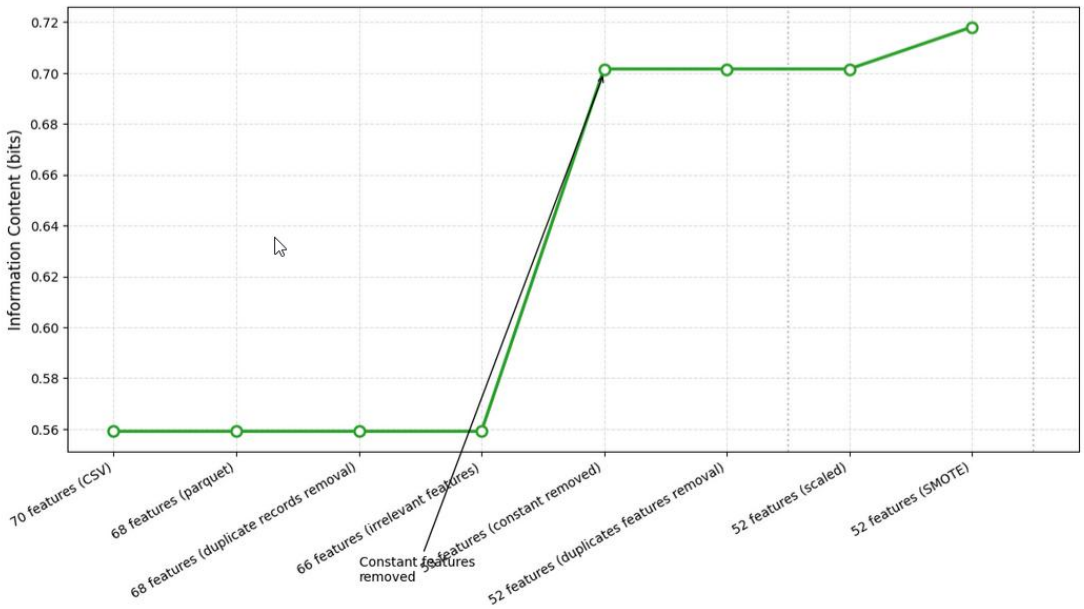

Figure 4-3: Feature Quality Assessment

4.4 Feature Selection Analysis and Dimension Reduction Review

When it comes to ML and data preprocessing, the number of features refers to the number of input variables inside a dataset. We can minimize the number of features through feature selection, while maintaining relevant features. It is an iterative process that starts with evaluating important features and recursively eliminating less important

features using cross-validation. The ultimate objective is to discover the most effective features over which it does not overfit but which do not add redundancy.

Among the techniques used for this task, MI and RFECV stand out as robust methods. Both are well-suited for handling datasets that exhibit non-normal distributions and complex, nonlinear relationships with the target variable. Since the same feature selection methods have been applied to two datasets (CICIoT2023 and CICAPTIIoT2024), this analysis will focus on the results derived from the CICAPTIIoT2024 dataset to avoid redundancy.

MI quantifies the dependency between each feature and the target variable by measuring how much knowing the feature reduces uncertainty about the target. A higher

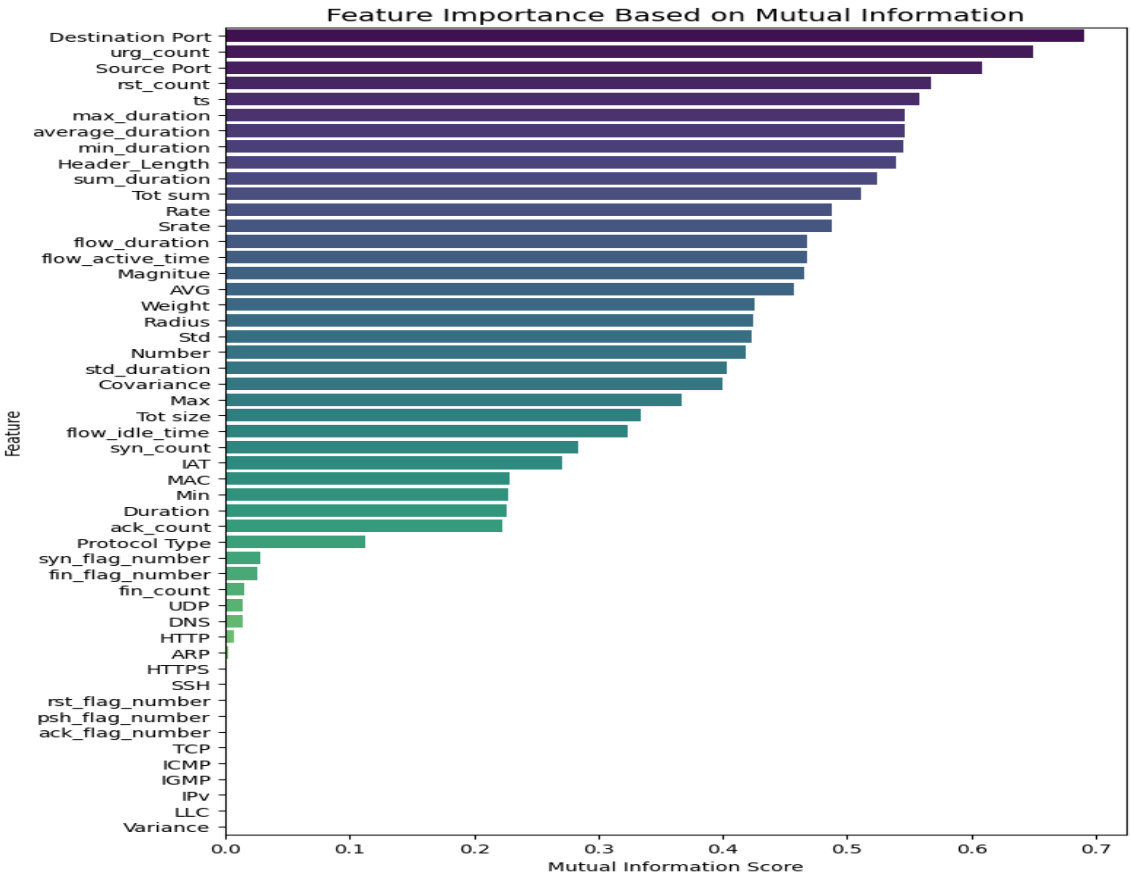

Figure 4-4: MI Feature Importance

MI score indicates a more substantial influence of the feature on predicting outcomes (Roy, 2022) (*Feature Selection Based on Mutual Information Gain for Classification and Regression*, 2021). This relationship is visually presented in Figure 4-4, which displays the MI-based feature importance scores.

Based on these results, features are categorized into three distinct groups: those with MI scores greater than zero (comprising 43 features), the top 20 features, and the top

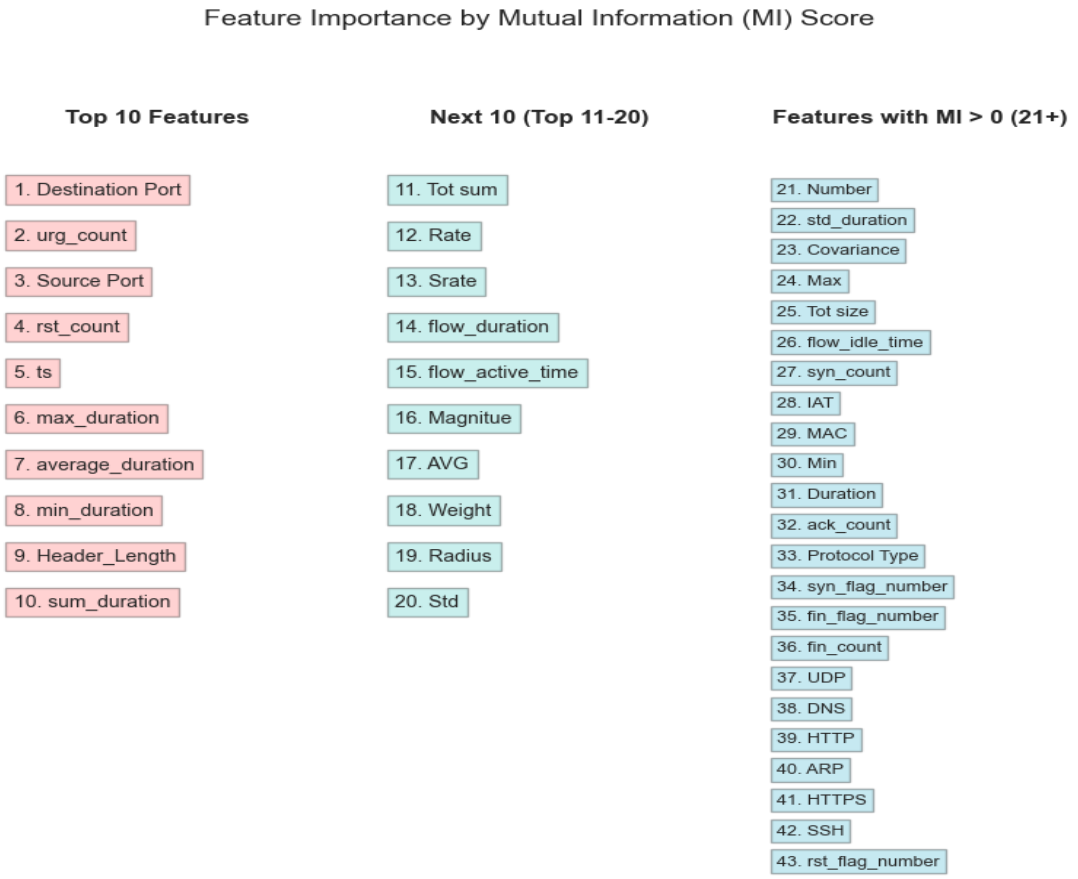

Figure 4-5: MI Features Grouping

10 features. Each group is used to train machine learning models separately, allowing performance comparisons across different feature subsets, as illustrated in Figure 4-5.

While RFECV does not assign explicit importance scores like MI, it offers a ranking mechanism where features are scored from most to least important (*RFE*, n.d.). The rank of one denotes the highest relevance, with larger values indicating decreasing importance. Figure 4-6 illustrates the RFECV rankings. Notably, within the same rank group, features are ordered by their contribution to model prediction, with the most impactful appearing first.

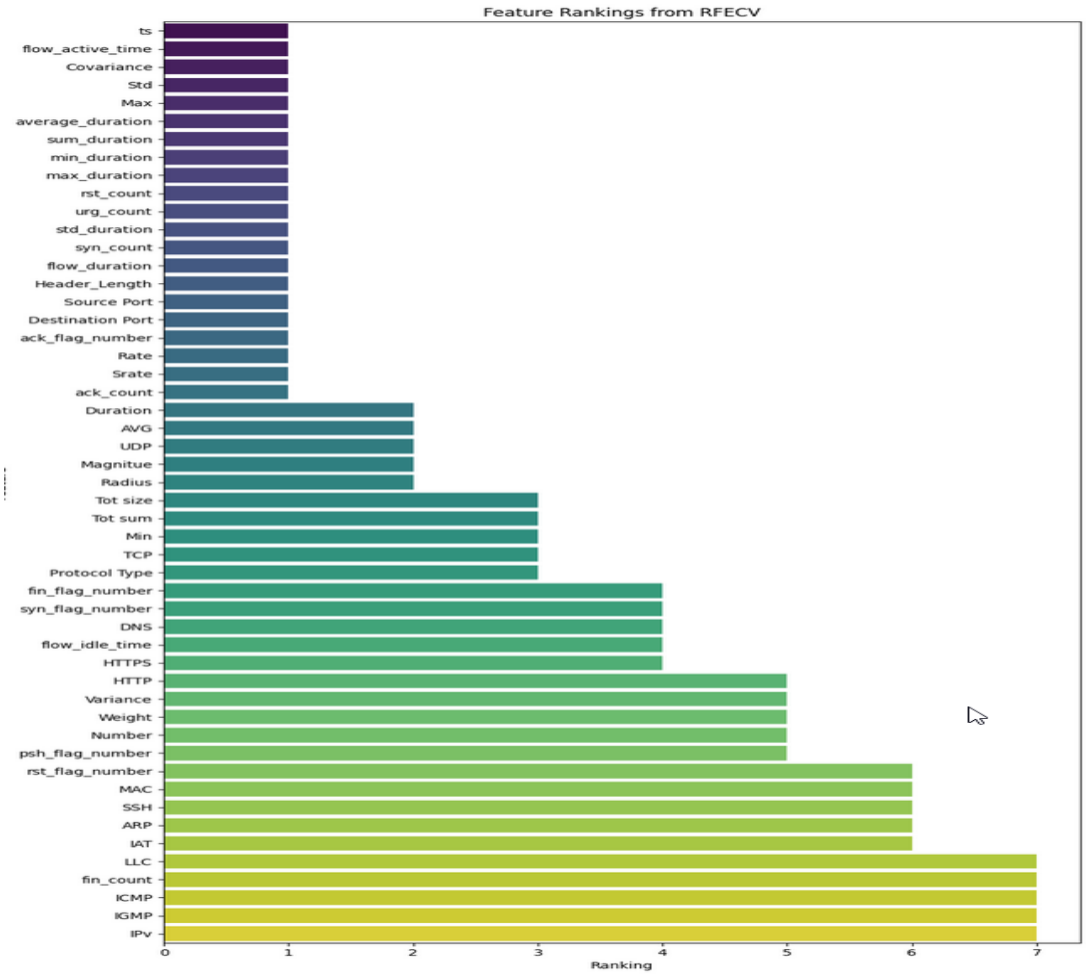

Figure 4-6: RFECV Features Ranking

Similar to the MI methodology, RFECV results are also divided into three groups, as depicted in Figure 4-7: features with rank values greater than 3, the top 20 ranked

features, and the top 10. This consistency in categorization supports a parallel comparison of both techniques across various feature sets and machine learning models.

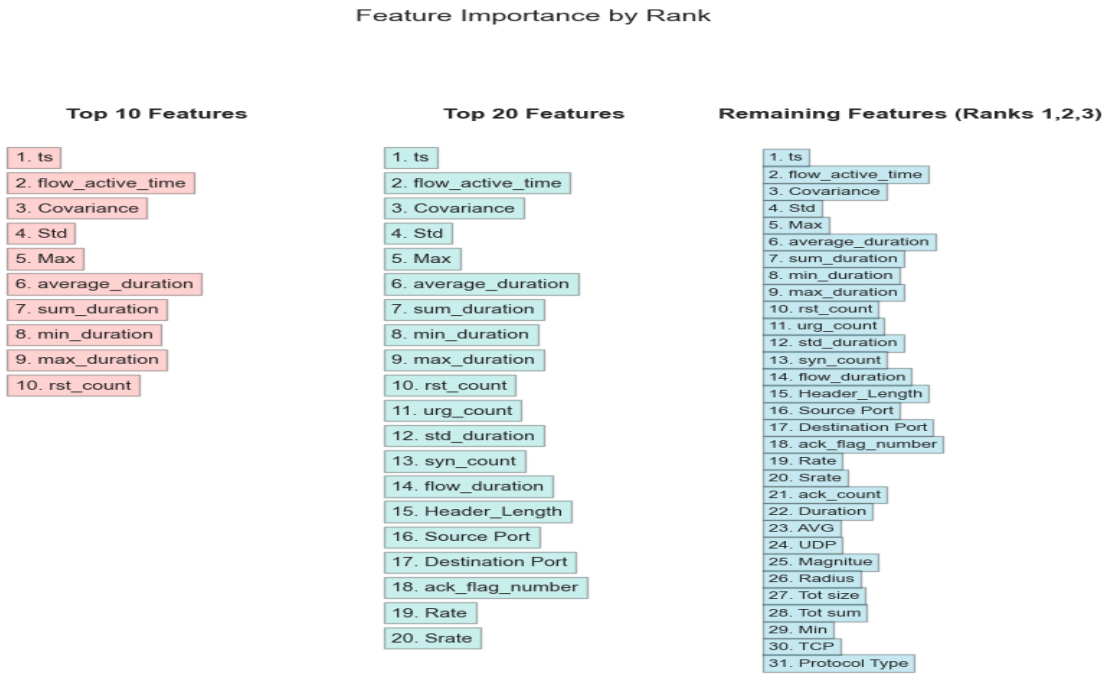

Figure 4-7: RFECV Features Grouping

Throughout the feature engineering process, it is essential to ensure that reducing the number of features does not negatively impact resource utilization. In fact, a linear reduction in resource usage is expected as feature sets become smaller. Figure 4-8 illustrates the relationship between feature dimensionality and factors such as file size,

Figure 4-8: Performance Metrics by Feature Selection Method

processing time, and memory usage during dataset loading and preprocessing for both MI and RFECV-based reductions.

Given that the datasets have been standardized and data types unified, it is reasonable to observe strong correlations between resource utilization and feature dimension reduction. This correlation supports the notion that the feature engineering process, along with prior preprocessing steps, did not introduce abnormalities or inconsistencies.

To complete the resource utilization analysis, both processing time and memory consumption were monitored during the training of models across the three feature groups. Figure 4-9 shows a significant improvement in memory usage with the smallest feature set.

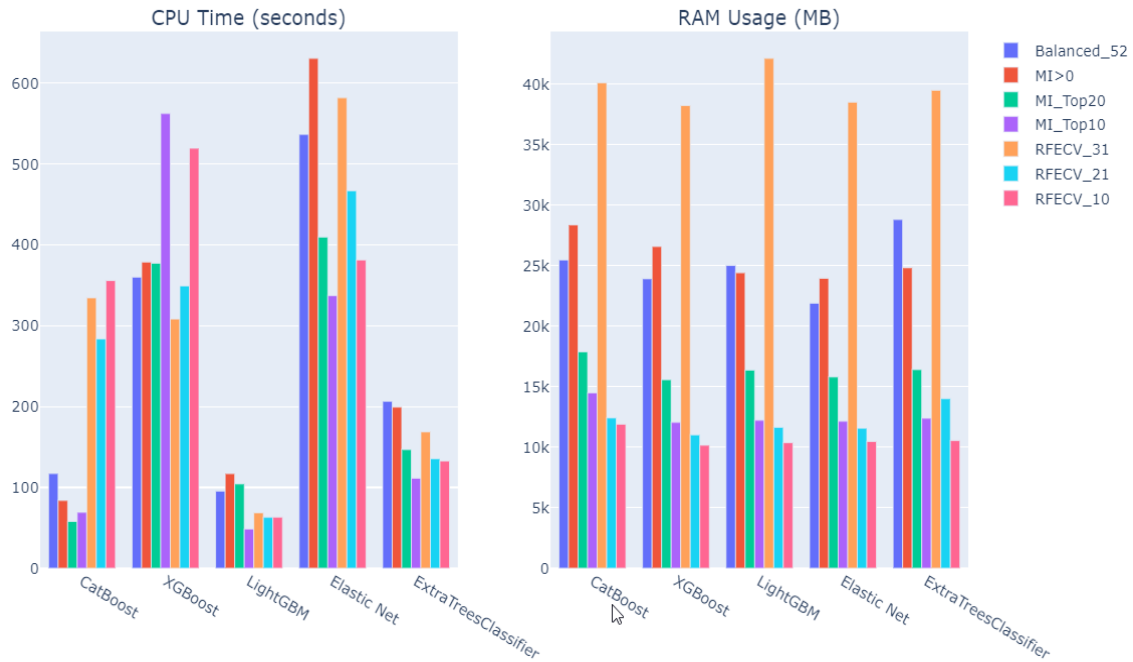

Figure 4-9: Processing Time & Memory Usage During Module Training

However, processing time did not exhibit a clear trend, suggesting that dimensionality reduction alone minimizes computation time. Interestingly, XGBoost and Elastic Net consistently emerged as the most computationally demanding algorithms, regardless of the number of features used in training.

Finally, to assess potential information loss following feature selection, entropy measurements were taken post-feature engineering, mirroring the earlier evaluations conducted after data preprocessing. Figure 4-10 presents the entropy values across different feature subsets and selection techniques. The findings not only confirm the

Figure 4-10: Entropy By Feature Selection Method

absence of information loss at this stage, but also suggests a potential information gain. This gain is likely due to the removal of low-importance features, which in turn increases the overall entropy, indicating a more informative and refined dataset.

4.5 Training and Fine-Tuning of Machine Learning Algorithmic Module

In the previous chapter, the focus was on selecting ML algorithms that best matched the characteristics of the two datasets utilized in this research. Care was taken in

handling nonlinear relations, non-normal data, and class imbalance, which were key factors in driving the algorithm selection process.

To ensure that each module ran optimally and with high accuracy during training, hyperparameter tuning was performed for each module. Additionally, the data was carefully divided to maintain the same class distribution in the training, evaluation, and testing sets to ensure a fair and balanced model assessment.

4.5.1 Boosting Algorithms (CatBoost/LightGBM/XGBoost)

Boosting algorithms, like LightGBM, CatBoost and XGBoost, are effective on datasets; they are popular machine and deep learning algorithms that often prove successful even on multidimensional datasets, such as complex network logs (CatBoost v. XGBoost v. LightGBM, n.d.). These algorithms deal with extensive feature sets and complex relationships while remaining scalable and efficient. Among the gradient boosting techniques that exist, the best implementations with respect to structured (tabular) data that use boosting methods are CatBoost, XGBoost and LightGBM.

Training on imbalanced datasets, like those often seen in cyber security or network analysis, needs adjustment of class weightage. Models like XGBoost and CatBoost support parameters *class_weight* or *scale_pos_weight* that help better model minority class. By weighing appropriately, it can make the model much more sensitive with no loss of performance.

The model is fit on the training set (*X_train, y_train*) and is evaluated frequently on the holdout set (*X_test, y_test*). With this configuration, we can make use of early stopping techniques during training, which stops early once performance on the

validation set stops improving. Early stopping helps reduce overfitting and aids hyper-parameter tuning, especially useful with *early_stopping_rounds* and similar parameters.

Several hyperparameters are important for model performance optimization. To avoid overfitting, parameters such as *learning_rate*, *depth* (CatBoost), *max_depth* (XGBoost and LightGBM), *l2_leaf_reg* (CatBoost), *min_child_weight* (XGBoost), and *min_data_in_leaf* or *num_leaves* (LightGBM) are commonly customized. Handling categorical features also varies among these algorithms: CatBoost offers first-class through *cat_features* and *one_hot_max_size*, LightGBM supports categorical features , while XGBoost requires manual encoding. Speed optimization is achieved by tuning settings like *iterations* (CatBoost), *colsample_bytree*, *subsample*, *n_estimators* (XGBoost and LightGBM), along with *feature_fraction* and *bagging_fraction*.

For tuning hyperparameters, we used specific configurations: LightGBM used *max_depth=6*, *learning_rate=0.01*, *num_leaves=127*, and *n_estimators=2000*; XGBoost used *max_depth=5*, *learning_rate=0.01*, and *n_estimators=1000*; CatBoost was configured with *depth=10*, *learning_rate=0.5*, *iterations=2000*, and *random_seed=42*. These choices were based on balancing training speed, overfitting control, and overall predictive performance.

As displayed in Figure 4-11, a log loss value was utilized during training for monitoring boosting-based algorithms' performance after tuning with hyperparameter optimization. Capturing log loss gives a normalized measure of the performance of the model, not just a measure of accuracy of the model. Log loss tells us how a model does when its probabilities are closer to a probability given by random or our most frequent class. If your log loss is low, you are confident and accurate.

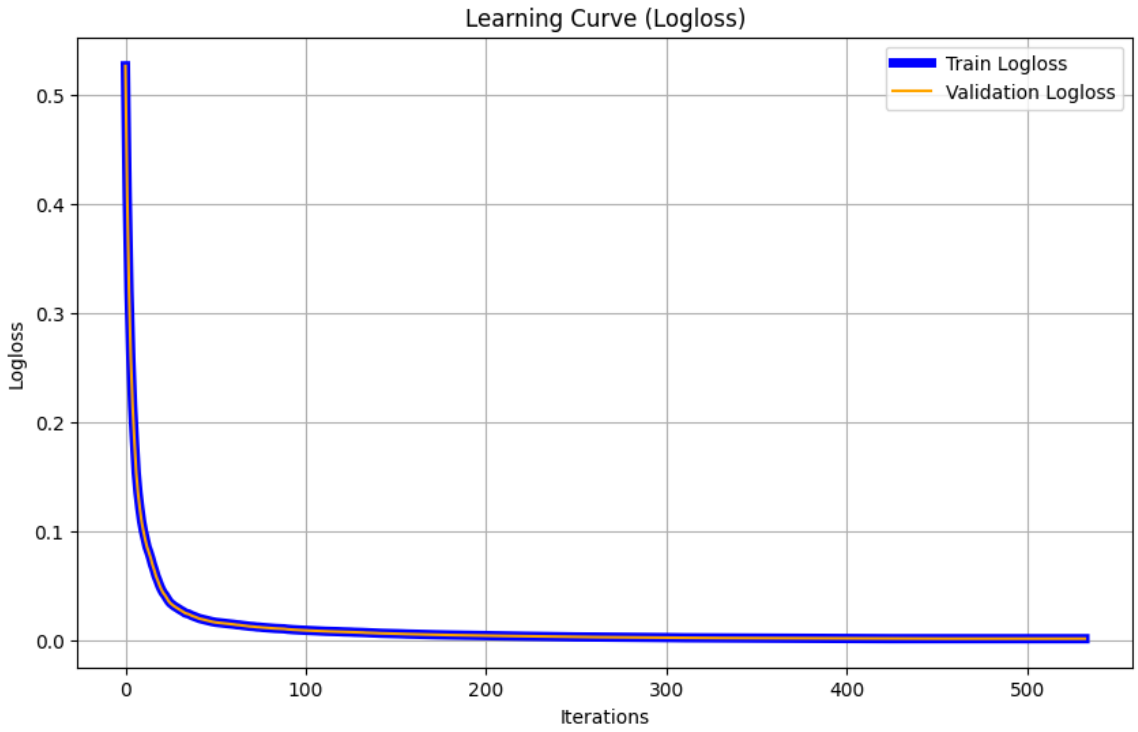

Figure 4-11: Module Evaluation via Logloss

4.5.2 Regularized Regression Models (Elastic Net)

Unlike more complex methods such as boosting algorithms, logistic regression is a basic model of ML. This model is quite straightforward, interpretable and is often used as a baseline for binary classification of data. Most of the boosting techniques perform better than logistic regression in capturing non-linear patterns. However, when it comes to smaller to medium-sized datasets, the decision boundary is approximately linear or it's not complex enough to require more complex classification algorithms. So, logistic regression may be preferred here. In these situations, tuning the regularization parameters of the model is needed for optimization.

When one uses regularization in regressions, the overall regression outputs becomes better and it does so by adding a penalty to the loss function so that overfitting

is not done. The penalty limits the complexity of the model and stops the algorithm from overemphasizing any feature. This is particularly useful in the context of log data, which may have an extensive and noisy feature space.

Elastic Net regularization combines both *L1* (Lasso) and *L2* (Ridge) penalties. The *L1* component encourages sparsity by driving some coefficients to zero, effectively performing feature selection. The *L2* component shrinks the remaining coefficients to reduce variance and handle multicollinearity. By blending these two regularization techniques, Elastic Net provides a balanced approach that adapts well to datasets with correlated or redundant features (LogisticRegression, n.d.).

In our implementation, the hyperparameter grid included *penalty="elasticnet"* to apply both *L1* and *L2* regularization, *C* as the inverse of regularization strength (smaller values increase regularization), and *l1_ratio* to balance *L1* (1.0) and *L2* (0.0) penalties. The *solver* must be set to "saga" to support Elastic Net, *max_iter* defines the number of iterations for convergence, and *class_weight="balanced"* helps address class imbalance. To evaluate these combinations, 5-fold cross-validation is used to ensure that the model's performance is tested across multiple subsets of the training data, which is more reliable than a single train/test split.

The Receiver Operating Characteristic Curve (ROC Curve) (*roc_auc*) is used as a scoring metric to assess model quality. Area Under the Curve (AUC) is particularly informative for imbalanced classification problems, where accuracy alone may be misleading. As shown in Figure 4-12, both the training and validation AUC curves rise steadily throughout training, indicating that the model is not only learning meaningful patterns from the training data but is also generalizing well to unseen samples. This

uptrend and correlation between the two curves suggest that the regularization strategy effectively guides the model to discriminate between classes without overfitting.

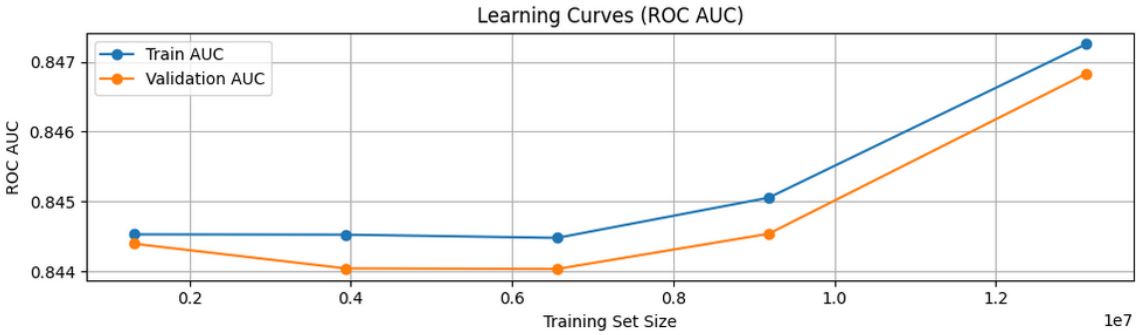

Figure 4-12: AUC Learning Curves

4.5.3 Bagging-based Ensemble Algorithm (Extremely Randomized Trees)

While Boosting algorithms build sequential models that learn from the mistakes of previous iterations, and Regularized Regression combines *L1* and *L2* penalties within a single objective function to prevent overfitting, Bagging-based methods take a different approach. In Bagging (Bootstrap Aggregating), independent and strongly randomized trees are trained on various subsets of the data, and their predictions are aggregated. This technique helps reduce variance while maintaining a balanced bias, resulting in more stable and generalized models (Geurts et al., 2006).

In the case of Extremely Randomized Trees (Extra Trees), the bagging component is optional and controlled by the bootstrap parameter. For this study, the bagging option is enabled (*bootstrap=True*) to improve generalization by decorrelating the trees. This configuration also helps reduce overfitting and increases training efficiency. Moreover, by training trees on different random subsets, the model becomes more robust to noise and less sensitive to minor fluctuations in the training data (ExtraTreesClassifier, n.d.).

Several hyper-parameters have been tuned to influence the model's bagging behavior. Setting *bootstrap=True* enables sampling with replacement. Setting *max_samples=0.5* ensures that each tree is trained on half the data, causing randomization in tree generation. In this case, a total of *n_estimators=200* trees have been used in to reduce variance with *random_state=42* ensuring the same sampled and the same tree that is built every time. The *max_depth=None* means the trees can grow as deep as they want, and the *class_weight='balanced'* helps with imbalanced classes. These settings work to improve model complexity and fairness together.

To see how the number of trees affects the performance of the model, models were trained with different values of *n_estimators*. The ROC AUC was noted for training validation for each configuration of each dataset. This analysis provides insights into how ensemble size impacts generalization.

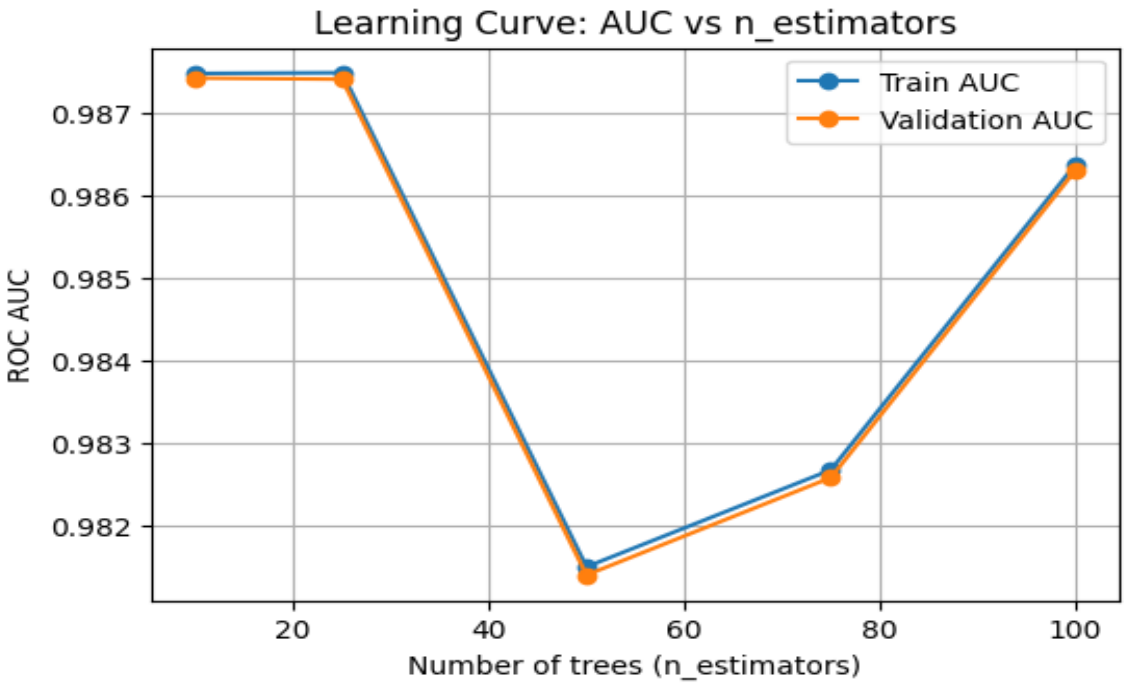

Figure 4-13: AUC vs n_estimators Learning Curves

Figure 4-13 contains the learning curves showing training and validation performance across forest sizes (*n_estimators*). With increase in the number of estimators, the training and validation AUC curves exhibit a strong collinearity which goes without much overfitting. It strengthens the case that bagging helps to reduce variance and improve predictive power.

4.6 Classification Modeling and Performance Evaluation

The outcomes of the evaluation metrics of the module training processes can be found in this section of the study. The effectiveness of these results for adequately answering the research questions is critically examined. In light of the above assessments, this study will assess whether there is sufficient evidence to reject or retain the hypotheses of the study.

The evaluation results are quite robust and consistent, showing the data preprocess, feature selection, hyperparameter tuning and similar process were properly executed. Thus, the entire design of the study can be trusted and so are the findings made through this study.

To support the study's research questions, the training results are organized into four groups to address each question systematically.

4.6.1 ***Group One*: Training Outcomes for Balanced and Imbalanced Datasets**

Tables 4-2 and 4-3 depict the first group, given the importance of considering training models with imbalanced data. This grouping helps evaluate how data imbalance impacts model performance and assessment, as illustrated in Figures 4-14 and 4-15.

Experiment	Metric	CatBoost	XGBoost	LightGBM	ElasticNet	ExtraTrees
Original 47	Precision	0.828	0.995	0.996	0.355	0.934
	Recall	0.999	0.691	0.614	0.978	0.960
	F1-Score	0.906	0.816	0.759	0.520	0.947
	Accuracy	0.992	0.999	0.985	0.930	0.996
SMOTE 47	Precision	0.983	0.989	0.992	0.824	0.985
	Recall	0.996	0.995	0.991	0.985	1.000
	F1-Score	0.990	0.992	0.992	0.898	0.992
	Accuracy	0.995	0.996	0.996	0.942	0.996

Table 4-2: Balanced vs Imbalanced Evaluation Results For 47 Features

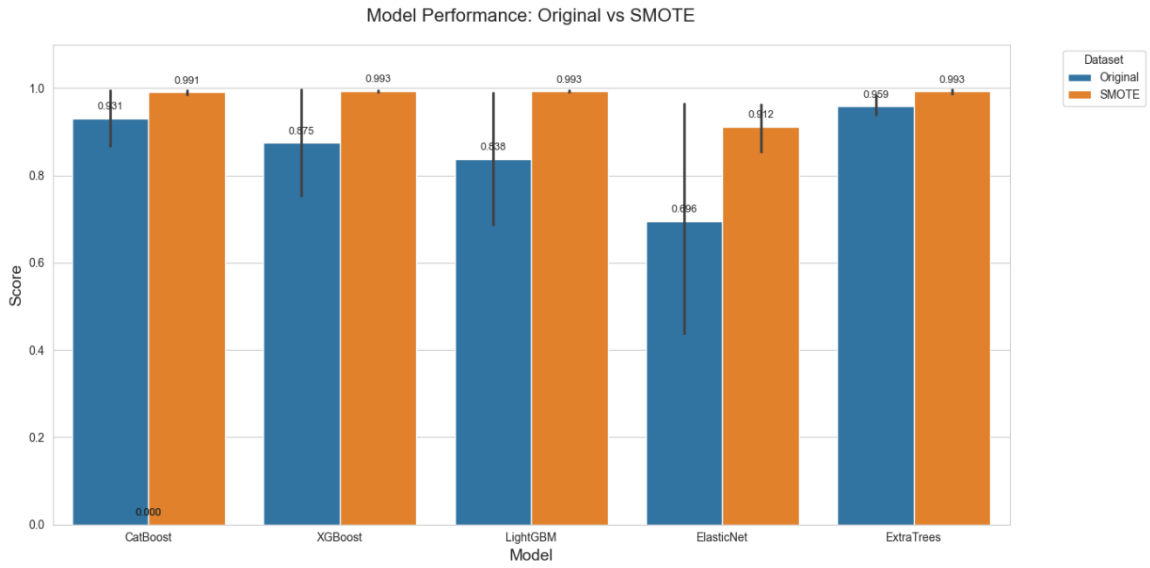

Figure 4-14: Original vs SMOTE Module Performance for 47 Features

Experiment	Metric	CatBoost	XGBoost	LightGBM	ElasticNet	ExtraTrees
Original 52	Precision	0.999	0.999	0.999	0.999	0.999
	Recall	0.999	1.000	0.999	0.908	0.999
	F1-Score	0.999	0.999	0.999	0.951	0.999
	Accuracy	0.999	0.999	0.999	0.908	0.999

SMOTE 52	Precision	1.000	1.000	0.999	0.997	1.000
	Recall	0.999	0.999	0.999	0.905	0.999
	F1-Score	0.999	1.000	0.999	0.949	0.999
	Accuracy	0.999	1.000	0.999	0.928	0.999

Table 4-3: Balanced vs Imbalanced Evaluation Results for 52 Features

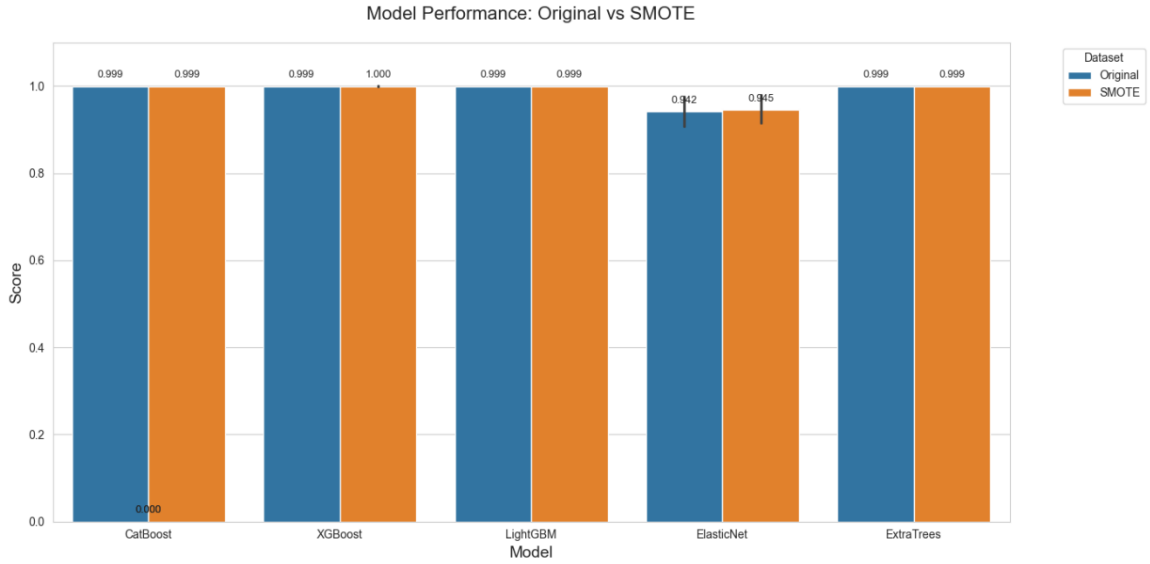

Figure 4-15: Original vs SMOTE Module Performance for 52 Features

4.6.2 ***Group Two*: Feature Selection Techniques, MI and RFECV, Applied Across Various ML Algorithms**

The second group evaluates the impact of combining two feature selection techniques. Tables 4-4 and 4-5 present the application of MI and RFECV across various machine learning models. The objective is to determine whether this hybrid approach improves performance compared to the baseline model trained on the complete feature set, as illustrated in Figures 4-16 and 4-17, thereby helping address RQ3.

Features	Models	Precision	Recall	F1-Score	Accuracy
RFECV 26	CatBoost	0.983	0.997	0.99	0.995
	XGBoost	0.971	0.992	0.981	0.99
	LightGBM	0.982	0.978	0.98	0.99
	ElasticNet	0.899	0.991	0.943	0.969
	ExtraTrees	0.5	0.983	0.663	0.993
RFECV 20	CatBoost	0.978	0.999	0.988	0.994
	XGBoost	0.988	0.996	0.992	0.996
	LightGBM	0.992	0.99	0.991	0.995
	ElasticNet	0.908	0.911	0.91	0.953
	ExtraTrees	0.986	1	0.993	0.996
RFECV 10	CatBoost	0.945	0.997	0.97	0.984
	XGBoost	0.967	0.99	0.978	0.989
	LightGBM	0.971	0.97	0.971	0.985
	ElasticNet	0.793	0.84	0.816	0.902
	ExtraTrees	0.966	0.998	0.982	0.99
MI 35	CatBoost	0.982	0.997	0.99	0.995
	XGBoost	0.989	0.995	0.992	0.996
	LightGBM	0.992	0.991	0.992	0.996
	ElasticNet	0.814	0.995	0.895	0.94
	ExtraTrees	0.985	1	0.992	0.996
MI 20	CatBoost	0.982	0.995	0.989	0.994
	XGBoost	0.986	0.995	0.99	0.995
	LightGBM	0.986	0.995	0.991	0.995
	ElasticNet	0.821	0.992	0.898	0.942
	ExtraTrees	0.984	1	0.992	0.996
MI 10	CatBoost	0.972	0.996	0.984	0.992
	XGBoost	0.977	0.995	0.986	0.993
	LightGBM	0.977	0.995	0.986	0.993
	ElasticNet	0.916	0.892	0.904	0.951
	ExtraTrees	0.978	0.999	0.988	0.994

Table 4-4: MI vs RFECV Top CICIoT2023 Features

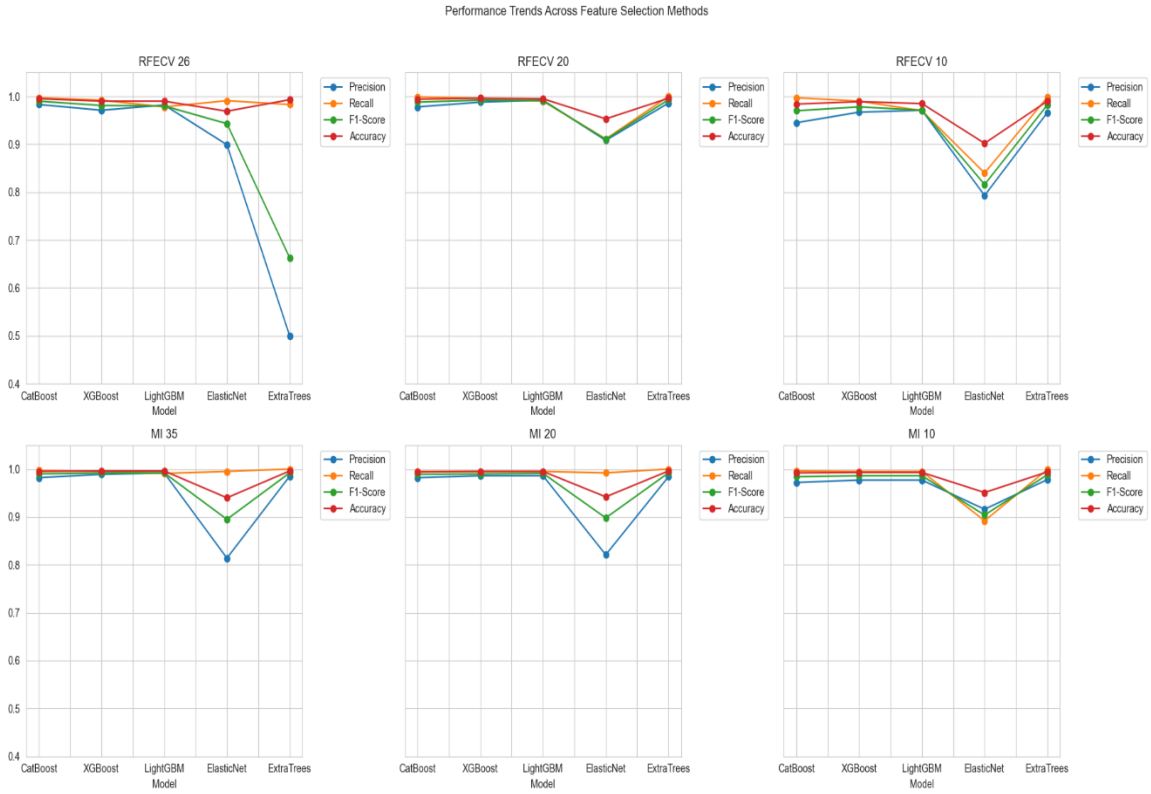

Figure 4-16: MI vs RFECV Top CICIoT2023 Features Performance

Features	Models	Precision	Recall	F1-Score	Accuracy
RFECV 32	CatBoost	1	0.999953	0.999976	0.999965
	XGBoost	1	0.999999	0.999999	0.999999
	LightGBM	1	0.999997	0.999999	0.999998
	ElasticNet	0.997131	0.951001	0.97352	0.961674
	ExtraTrees	1	0.999994	0.999997	0.999996
RFECV 20	CatBoost	0.999999	0.999961	0.99998	0.999971
	XGBoost	1	0.999999	0.999999	0.999999
	LightGBM	0.999998	0.999997	0.999998	0.999996
	ElasticNet	0.997746	0.816895	0.898308	0.862999
	ExtraTrees	1	0.999997	0.999998	0.999997
RFECV 10	CatBoost	0.999881	0.999492	0.999687	0.999536
	XGBoost	0.999999	0.999989	0.999994	0.999991
	LightGBM	0.999989	0.999975	0.999982	0.999973
	ElasticNet	0.991883	0.681316	0.807777	0.759808
	ExtraTrees	1	0.999993	0.999996	0.999995
MI 43	CatBoost	0.999996	0.999959	0.999977	0.999967
	XGBoost	1	0.999999	1	1
	LightGBM	0.999997	0.999996	0.999997	0.999995
	ElasticNet	0.997173	0.850167	0.917821	0.887228
	ExtraTrees	1	0.999995	0.999998	0.999997
MI 20	CatBoost	0.999995	0.999865	0.99993	0.999896
	XGBoost	1	0.999996	0.999998	0.999997

	LightGBM	0.999999	0.999996	0.999998	0.999996
	ElasticNet	0.99652	0.78931	0.880894	0.841892
	ExtraTrees	1	0.999996	0.999998	0.999997
MI 10	CatBoost	0.999998	0.99987	0.999934	0.999902
	XGBoost	0.999999	0.999965	0.999982	0.999973
	LightGBM	0.99999	0.999973	0.999981	0.999972
	ElasticNet	0.988891	0.690488	0.813178	0.764986
	ExtraTrees	1	0.999987	0.999993	0.99999

Table 4-5: MI vs RFECV Top CICAPTIIoT2024 Features

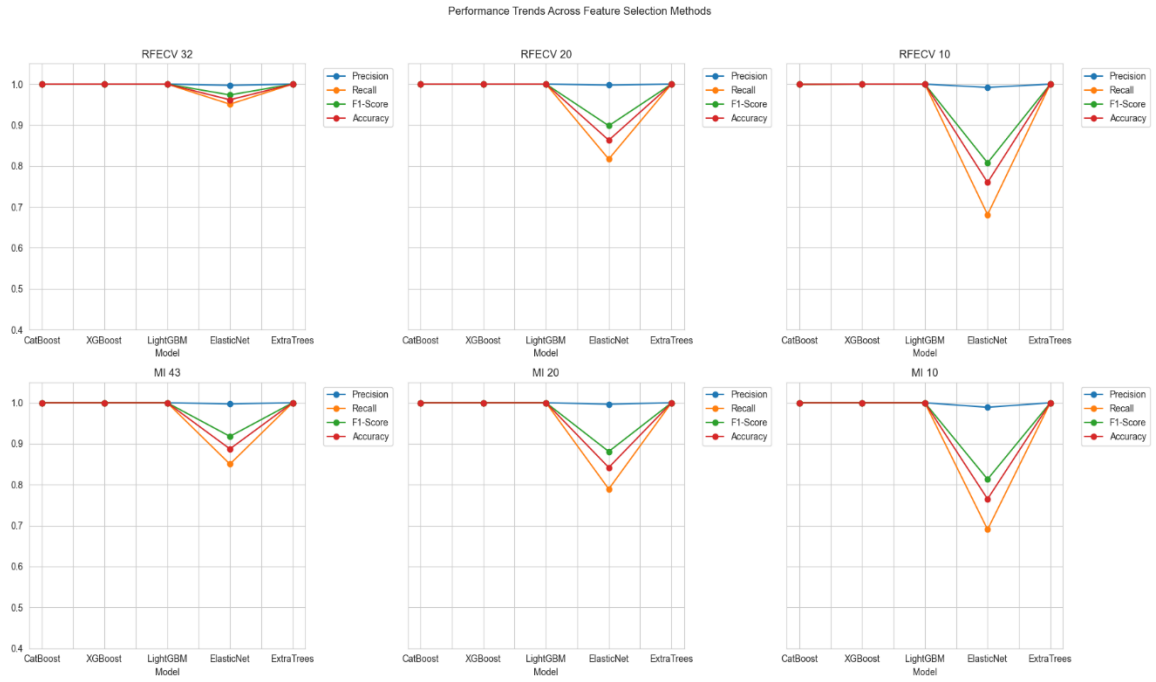

Figure 4-17: MI vs RFECV Top CICAPTIIoT2024 Features Performance

4.6.3 ***Group Three*: Evaluating Underexplored Classifiers with Emphasis on Industry-specific Features**

The third group, shown in Table 4-6, contributes to addressing RQ1 by incorporating performance metrics from relatively underexplored ML algorithms, such as CatBoost, LightGBM, ElasticNet and ExtraTrees, which are not commonly featured in prior studies. As illustrated in Figure 4-18, this group emphasizes the role of industry-specific features present in the CICAPTIIoT2024 dataset, in contrast to the more general IoT features found in the CICIoT2023 dataset.

Datasets	Models	Precision	Recall	F1-Score	Accuracy
CICIoT2023	CatBoost	0.982848	0.996455	0.989604	0.994568
	LightGBM	0.992422	0.99115	0.991786	0.99574
	ElasticNet	0.824177	0.985184	0.897517	0.94167
	ExtraTrees	0.984575	0.9998	0.992129	0.995887
CICAPTIIoT2024	CatBoost	1	0.999955	0.999978	0.999967
	LightGBM	0.999999	0.999993	0.999996	0.999994
	ElasticNet	0.99709	0.905811	0.949261	0.928272
	ExtraTrees	1	0.999996	0.999998	0.999997

Table 4-6: Comparative Evaluation of Models on CICIoT2023 and CICAPTIIoT2024

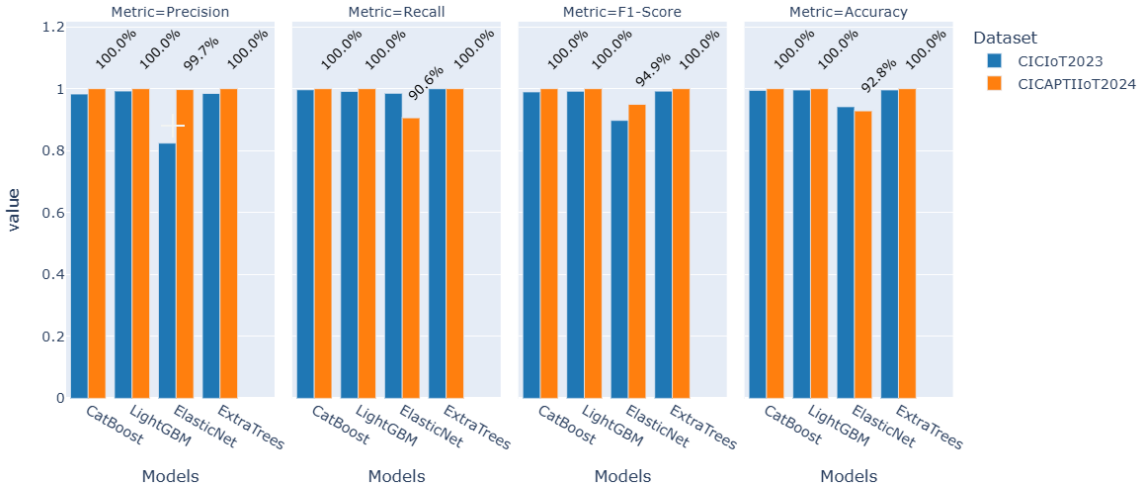

Figure 4-18: Comparative Evaluation of Models on CICIoT2023 and CICAPTIIoT2024

4.6.4 ***Group Four*: Cross-Dataset Comparison of Literature-Based Models**

The fourth group, represented in Table 4-7, contributes to answering RQ2 by comparing the performance of models previously examined in the literature. This comparison, as demonstrated in Figure 4-19, provides insights into the models' generalizability and adaptability when exposed to evolving threat landscapes.

Studies	Models	Accuracy	Precision	Recall	F-1 Score
(Yaras & Dener, 2024)	CNN, LSTM	99.995	99.995	99.95	99.995
(Huynh et al., 2023)	SVM	99.36	99.44	99.85	99.65
	LR	99.15	99.18	99.89	99.53
(Pirtama et al., 2024)	RF	09934	51.84	43.70	NA
(Okey et al., 2024)	CNN-GRU	98.13	95.93	98.95	NA
	LSTM-GRU	94.56	92	91.49	NA
(Kumar et al., 2024)	SVM	98.30	88.79	68.97	75.32
	DT	99.58	95.50	95.51	95.51
	RF	99.68	96.69	96.41	96.55

	AdaBoost	99.62	95.23	96.78	96.00
	LG	98.90	86.31	89.04	87.6
(Ramli & Thereza, 2023)	DT	100	100	100	100
	KNN	99.95	98.13	99.04	99.04
	RF	100	100	100	100
	Naïve Bayes	93.23	97.41	97.41	59.2
(Gheni & Al-Yaseen, 2024)	(MLP)	99.26	99	99	99
	(AE)	98.76	99	99	99
(Liu et al., 2024)	RMCNN	97.28	97.28	97.35	97.29
(Khanday et al., 2024)	LSTM	92	92	92	91
	1D-CNN	99.87	100	100	99
(Modi, 2024)	RF	97.631	95.45	95.12	95.28
	DT	96.528	95.39	96.52	95.95
	XGBoost	97.642	95.33	97.64	96.47
(Becerra-Suarez et al., 2024)	DNN	99.38	99.42	99.38	99.40
	LSTM	99.36	99.39	99.36	99.37
	CNN	99.40	99.43	99.40	99.41
(Gupta et al., 2024)	hybrid	99.39	79.0	90.89	83.01
(Le et al., 2023)	Blending	99.51	98.51	99.63	99.07
(Narayan et al., 2023)	RF	99.43	78.25	90.29	81.86
(Altrad, 2023)	Naïve Bayes	65.2	79.1	65.5	70.4
	DT	98.0	98.0	98.0	97.6
	RF	98.0	98.1	98.0	93.9
	QDA	14.3	02.0	14.3	03.6
	AdaBoost	13.1	43.8	13.1	14.1
	MLP	90.6	90.3	90.6	90.2
	KNN	97.4	97.4	97.3	97.3
(Phan et al., 2024)	DT	99.31	99.32	99.31	99.31
	RF	99.45	99.45	99.45	99.44
	KNN	98.86	98.82	98.86	98.80
	GB	99.43	99.43	99.43	99.42
	MLP	88.85	89.54	88.85	88.88
(Priya et al., 2025)	WPCA_E-DNN	95.2	93.1	92.7	92.6
(Narkedimilli et al., 2025)	Ensemble	98	99	95	96
This Study	CatBoost	99.99	100	99.99	99.99
	XGBoost	100	100	99.9	100
	LightGBM	99.99	99.99	99.99	99.99
	ElasticNet	92.80	99.70	90.50	94.90
	ExtraTrees	99.99	100	99.99	99.99

Table 4-7: Cross-Dataset Model Performance Comparison

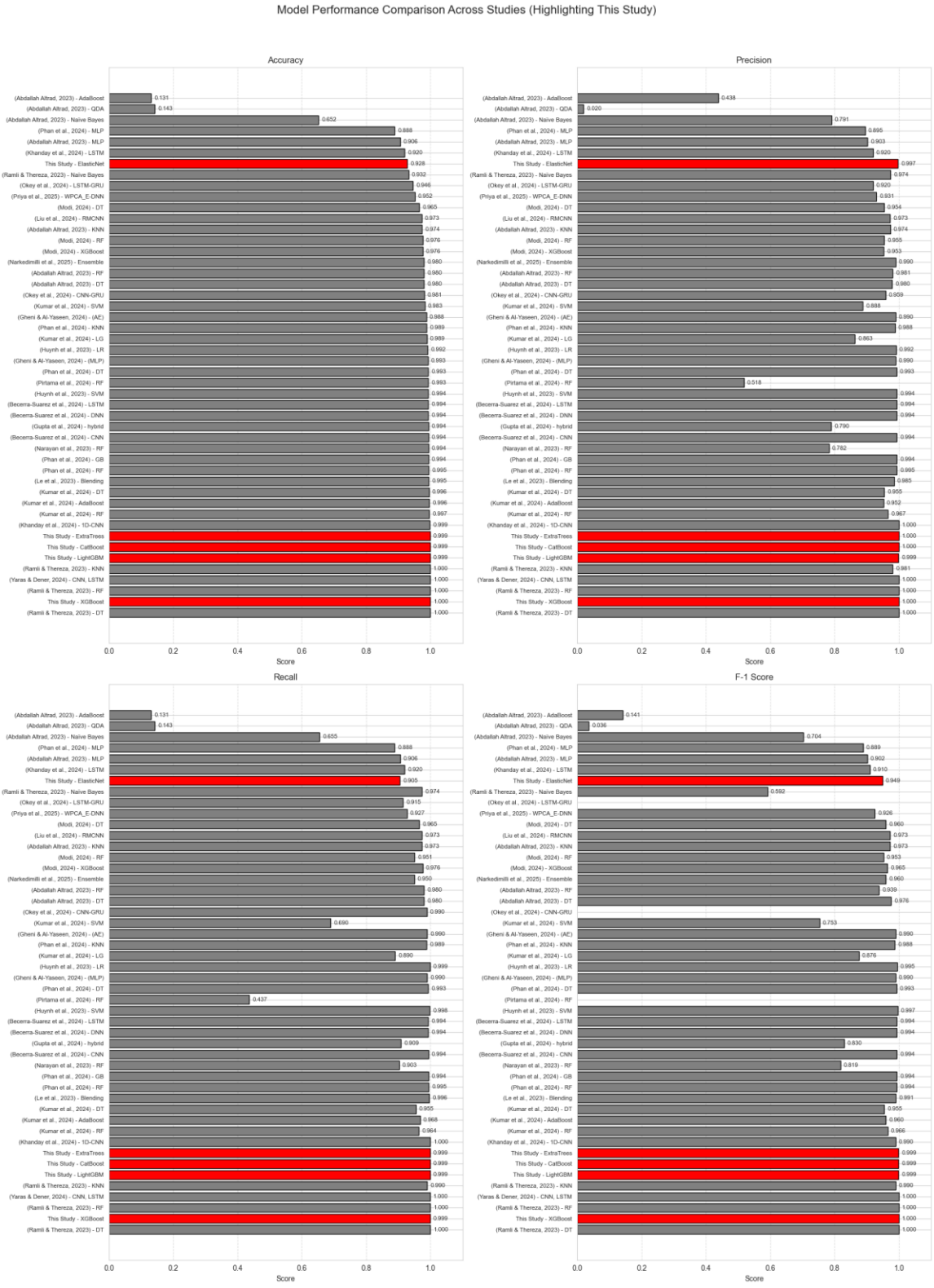

Figure 4-19: Model Performance Comparison Across Studies

4.7 Model Behavior and Analytical Insights

In the previous section, evaluation results were organized into four distinct groups to help address the study's research questions with a structured, evidence-based approach. ***Group One*** focused on comparing model performance metrics before and after applying the SMOTE technique to address class imbalance in both the CICIoT2023 and CICAPTIIoT2024 datasets. As shown in Figures 4-14 and 4-15, applying SMOTE significantly improved average performance in CICIoT2023, while the impact on CICAPTIIoT2024 was minimal.

To assess whether these differences are statistically significant, the Kruskal–Wallis test was used, following a normality check using Figure 4-20, which confirmed that the data did not meet the assumption of normality (Kim, 2014).

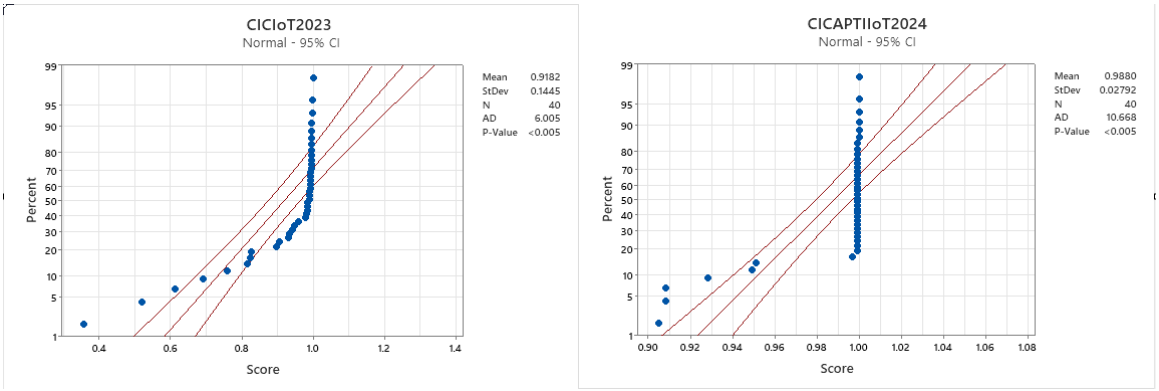

Figure 4-20: Normality Test for Original and SMOTE Groups

The test results, as presented in Figure 4-21, show a p-value below 0.05 for the CICIoT2023 dataset, indicating a statistically significant difference between the original and SMOTE-augmented versions. In contrast, the CICAPTIIoT2024 dataset showed no significant difference across most models, except for ElasticNet, where variation was observed.

Overall, the distribution plots and statistical test results underscore the negative impact of class imbalance on model performance, particularly in CICIoT2023, where imbalance was more pronounced compared to CICAPTIIoT2024.

Kruskal-Wallis Test: CICIoT2023

Descriptive Statistics

Experiment	N	Median	Mean Rank	Z-Value
Original	20	0.9405	16.4	-2.22
SMOTE	20	0.9920	24.6	2.22
Overall	40		20.5	

Test

Null hypothesis H_0: All medians are equal
Alternative hypothesis H_1: At least one median is different

Method	DF	H-Value	P-Value
Not adjusted for ties	1	4.92	0.027
Adjusted for ties	1	4.95	0.026

Kruskal-Wallis Test: CICAPTIIoT2024

Descriptive Statistics

Experiment	N	Median	Mean Rank	Z-Value
Original	20	0.999	19.2	-0.69
SMOTE	20	0.999	21.8	0.69
Overall	40		20.5	

Test

Null hypothesis H_0: All medians are equal
Alternative hypothesis H_1: At least one median is different

Method	DF	H-Value	P-Value
Not adjusted for ties	1	0.48	0.490
Adjusted for ties	1	0.69	0.406

Figure 4-21: Kruskal-Wallis Test Results for Group One

Although **Group One** is not designed to support a specific research question, it aims to establish a strong foundation for all machine learning algorithms by preserving data quality and improving overall model performance.

In ***Group Two***, a comparative analysis of the evaluation results across the two feature selection techniques, MI and RFECV, reveals consistent model performance, even when the number of selected features varies within each technique. This consistency becomes more evident in the CICAPTIIoT2024 dataset, highlighting the robustness of these selection methods.

However, as illustrated in Figures 4-16 and 4-17, ElasticNet shows a noticeable decline in performance, particularly as the number of selected features decreases. This

trend is consistent across both datasets and becomes more pronounced as feature dimensionality is reduced.

Figures 4-22 and 4-23 further support the observation that reducing feature sets does not yield substantial performance improvements across most models. The only notable exception appears in the ElasticNet model on the CICIot2023 dataset, where using the top 26 RFECV-selected features results in an improvement of over 1% in true positive rate compared to the baseline dataset (with balanced datasets). In contrast, the

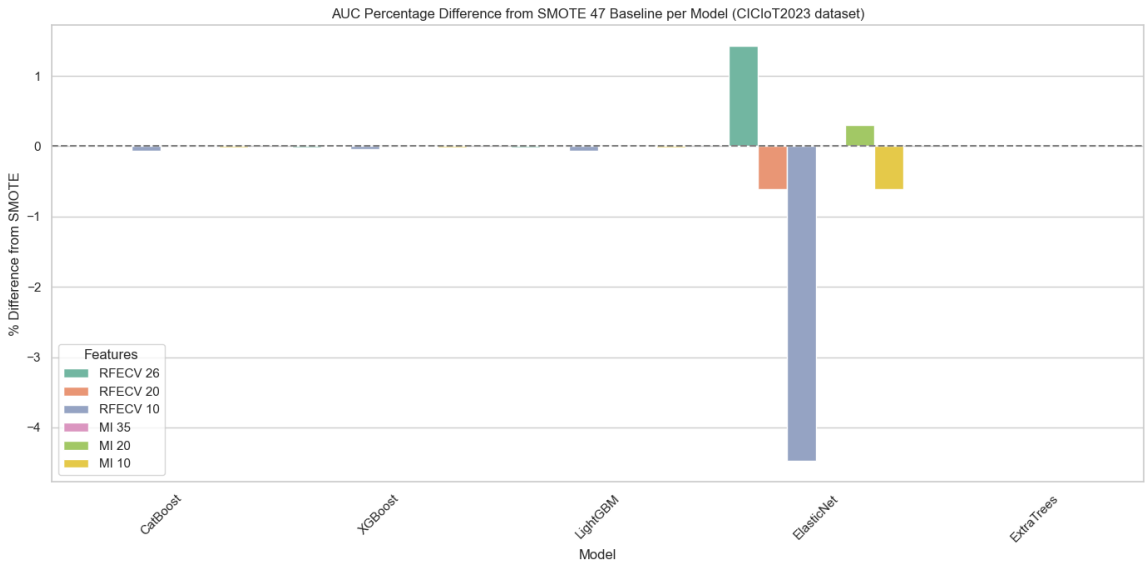

Figure 4-22: AUC Evaluation for CICIoT2023

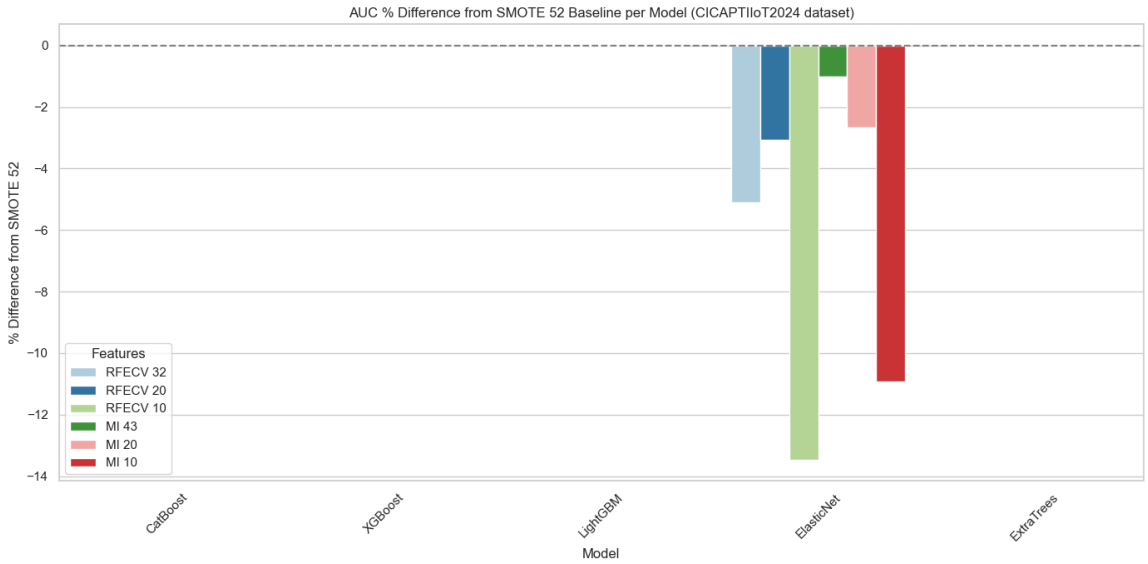

Figure 4-23: AUC Evaluation for CICAPTIIoT2024

same model behaves differently on the CCIAPTII2024 dataset, where performance drops significantly by more than 10% when the feature set is reduced to just 10 features using either MI or RFECV techniques.

The evaluation results in ***Group Three*** reveal mild differences in model performance across the two datasets, CICIoT2023 and CICAPTIIoT2024, as illustrated in Figure 4-18. To determine whether these differences are statistically significant, a two-way Analysis of Variance (ANOVA) was conducted (Chapman, 2018). Given that the residuals appear approximately normally distributed and the variances are relatively homogeneous across groups (Figure 4-24), the test remains appropriate, as ANOVA is generally robust to moderate deviations from normality.

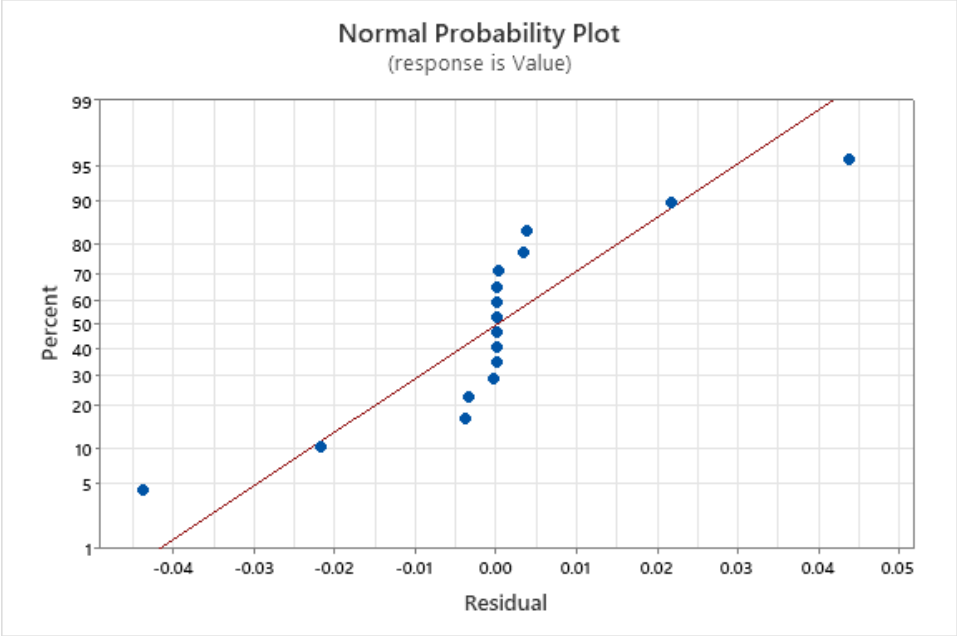

Figure 4-24: Group Three Residuals Distribution

As shown in Figure 4-25, the p-value for the DataSets factor indicates no significant difference in overall performance between the two datasets when averaged across all models. In contrast, the Models factor yielded a statistically significant result, suggesting that at least one model's performance differs notably across datasets. The interaction term between DataSets and Models was not significant, meaning the relative performance ranking of models remained stable across datasets.

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
DataSets	1	0.000008	0.000008	0.01	0.911
Models	3	0.011652	0.003884	6.42	0.016
DataSets*Models	3	0.000320	0.000107	0.18	0.909
Error	8	0.004839	0.000605		
Total	15	0.016820			

Figure 4-25: Two-way ANOVA Analysis of Variance

To further interpret these results, the proportion of variance explained by each factor was calculated using eta-squared ($\boldsymbol{\eta^2}$) (Bobbitt, 2020). As detailed in Table 4-8, model selection accounts for approximately 69% of the total performance variation, while dataset choice contributes a negligible 0.05%. These findings are consistent with the p-values and reinforce the conclusion that model architecture is the dominant factor influencing performance.

$$\boldsymbol{The\ formulas\ for}\ \eta^2 = \frac{\mathrm{SS}_{factor}}{\mathrm{SS}_{Total}}$$

The percentage is

Source	SS	η²	% Variance Explained
Datasets	0.000008	0.000008/0.01682 ≈ 0.000476	0.05%
Models	0.011652	0.011652/0.01682 ≈ 0.692455	69.25%
Datasets × Models	0.000320	0.000320/0.01682 ≈ 0.019031	1.90%
Error + Others	rest	–	28.80%

Table 4-8: Percent of Variance Explained ($\boldsymbol{\eta^2}$)

Figure 4-26 provides a visual summary of how each model's Precision, Recall, F1-Score, and Accuracy changed between the two datasets. Most models show a modest improvement of approximately 1% when applied to CICAPTIIoT2024. However, both the dataset (CICAPTIIoT2024) and the selected ML algorithms contributed to consistently high accuracy, exceeding 99.99% across all models, with the exception of ElasticNet as demonstrated in Appendix A, Table 1.

ElasticNet demonstrates a distinct pattern, exhibiting a noticeable decline in Recall and Accuracy, alongside a substantial improvement in Precision and F1-score.

This contrast suggests that while the dataset shift has a minimal overall effect, specific models may be more sensitive to underlying data characteristics.

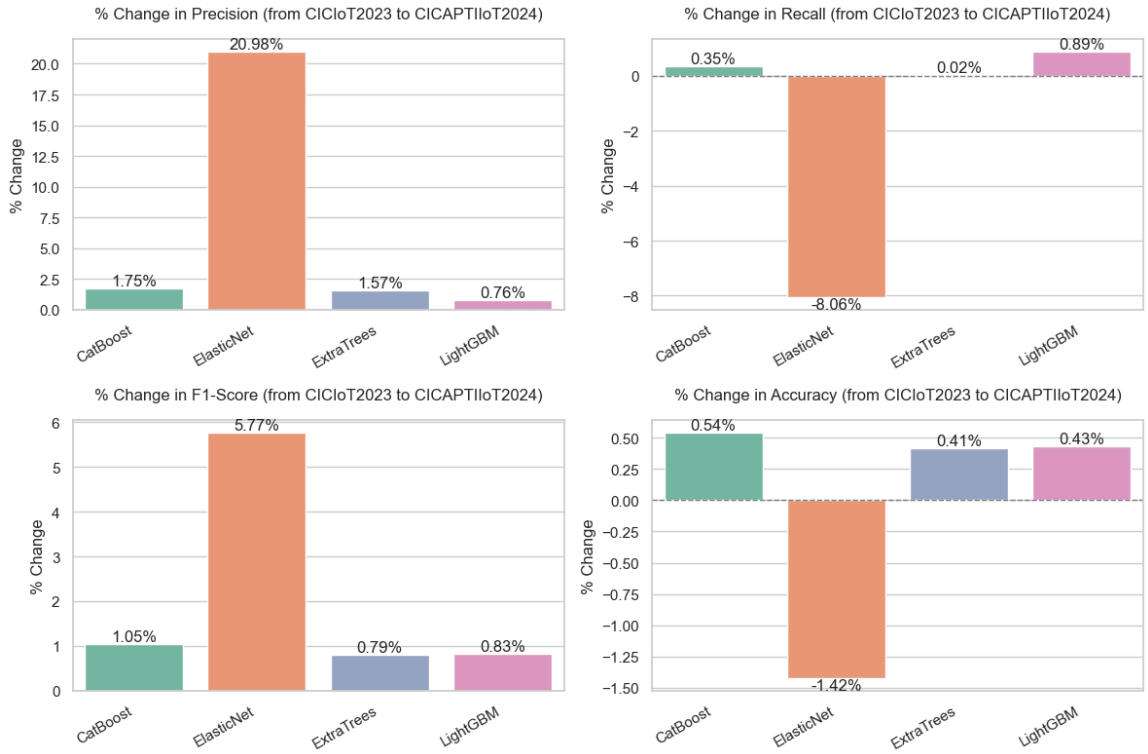

Figure 4-26: Percentage Difference in Means

In ***Group Four***, the average performance metrics of various machine learning algorithms reported in previous studies, as showed in Table 4-7, were compared with those trained on the CICAPTIIoT2024 dataset. As illustrated in Figure 4-27, the models trained using industry-specific features demonstrated a consistent performance improvement, averaging approximately 10% over those from earlier studies. Furthermore, the models exhibited high performance, with Precision and Accuracy exceeding 99.99%, and Recall and F1-Score reaching up to 99.99%, as illustrated in Figure 4-19.

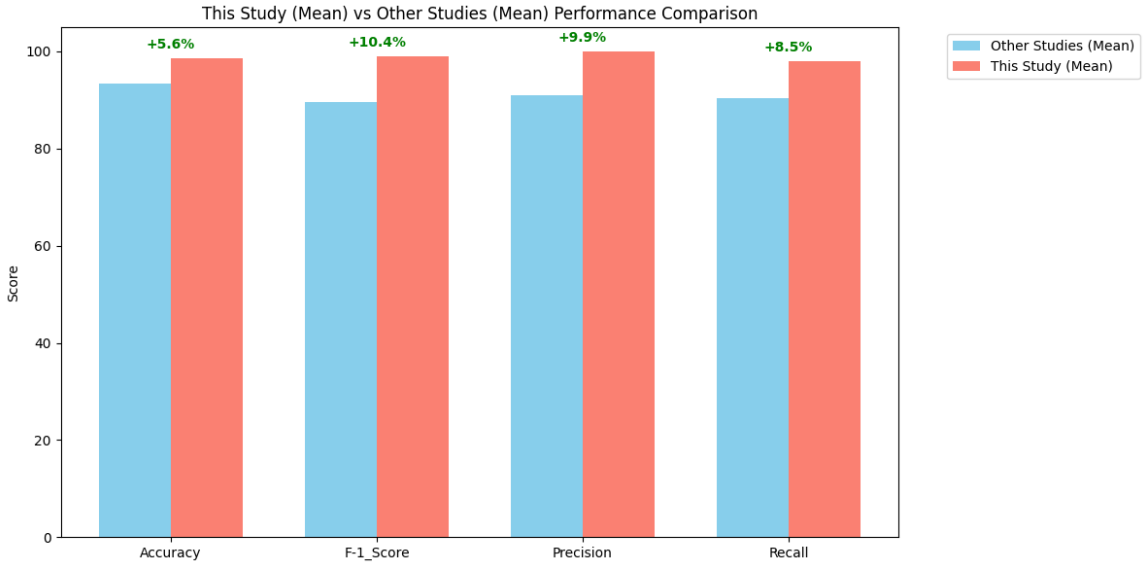

Figure 4-27: Mean Score Comparison with Literature Review Studies

To ensure a precise and meaningful evaluation, performance must be compared against the highest-performing model identified in the literature. Figure 4-28 highlights the study by (Ramli & Thereza, 2023), in which the DT and RF models achieved perfect scores of 100% across all key performance metrics. In contrast, the models developed in this study showed an average performance of approximately 1% lower than this benchmark.

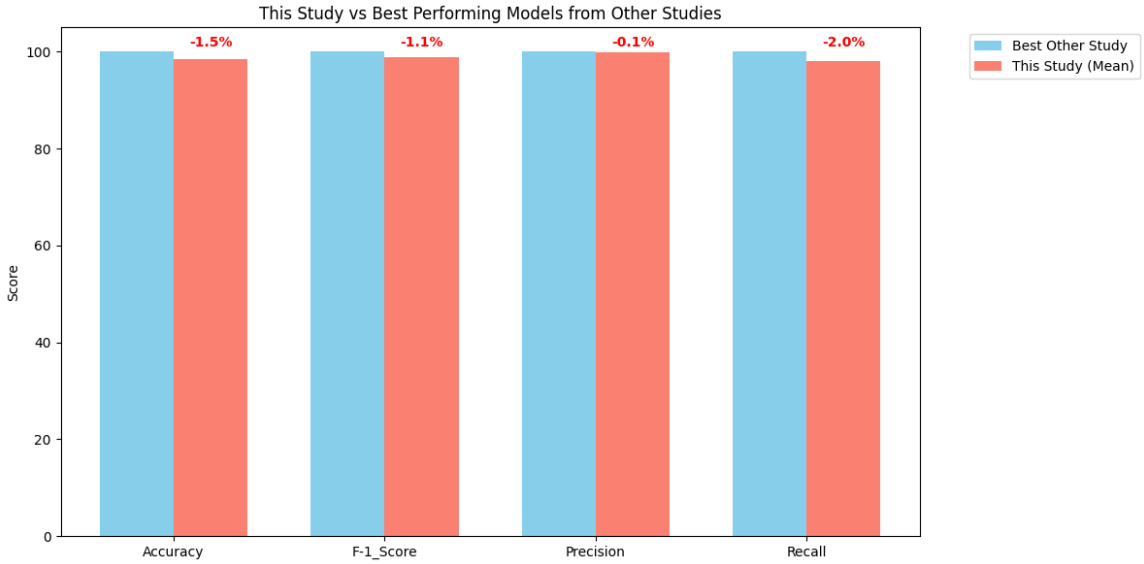

Figure 4-28: Mean Score Comparison with Perfect Score

4.8 Explainable AI and Interpretability

To verify if the performance and prediction of the models, particularly in safety-critical industrial IOT systems, exhibit bias or overfitting. Partial Dependence Plots (PDPs) are essential to help understand features effects on model predictions and thus enable responsible use of ML (Liu et al., 2020).

PDPs visualize whether the relationship between a feature and the target variable is linear, monotonic, or more complex. In this section, two of the top ten influential features, *"ts"* and *"rst_count"*, from the CICAPTIIoT2024 dataset were evaluated using PDPs to understand their effect on CatBoost model predictions.

As shown in Figure 4-29, *"ts"* has little impact until it reaches a threshold, after which the model's performance increases sharply, indicating that small changes in *"ts"* strongly influence predictions beyond this point. In contrast, *"rst_count"* shows an immediate steep rise, suggesting that increases from its minimum value significantly boost the model's confidence. These trends align with earlier feature selection results, confirming both features' strong positive contribution to prediction accuracy.

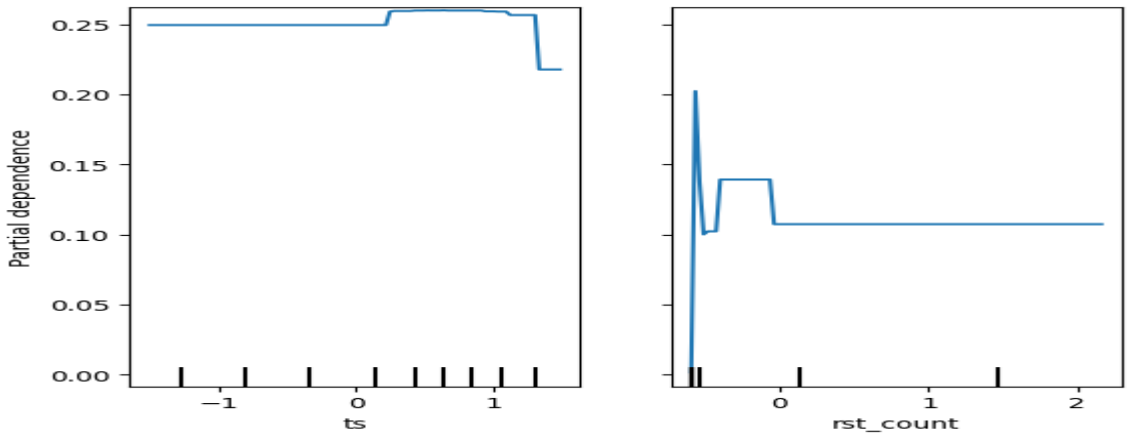

Figure 4-29: PDP Plots for *"ts"* and *"rst_count"*, Features

To uncover variations that PDPs may obscure through averaging, an Individual Conditional Expectation (ICE) plot was generated for "*ts*" (Figure 4-30) (Dev, 2024). The ICE plot reveals a flat response for "*ts*" values below 0, followed by a steep increase that mirrors the PDP line. A dense cluster of ICE lines intersects the PDP curve between 0 and 0.5, indicating consistently high predictions in this range across instances. This suggests that the model is particularly sensitive to "*ts*" in this interval, reinforcing its importance and highlighting a critical threshold where its influence becomes pronounced.

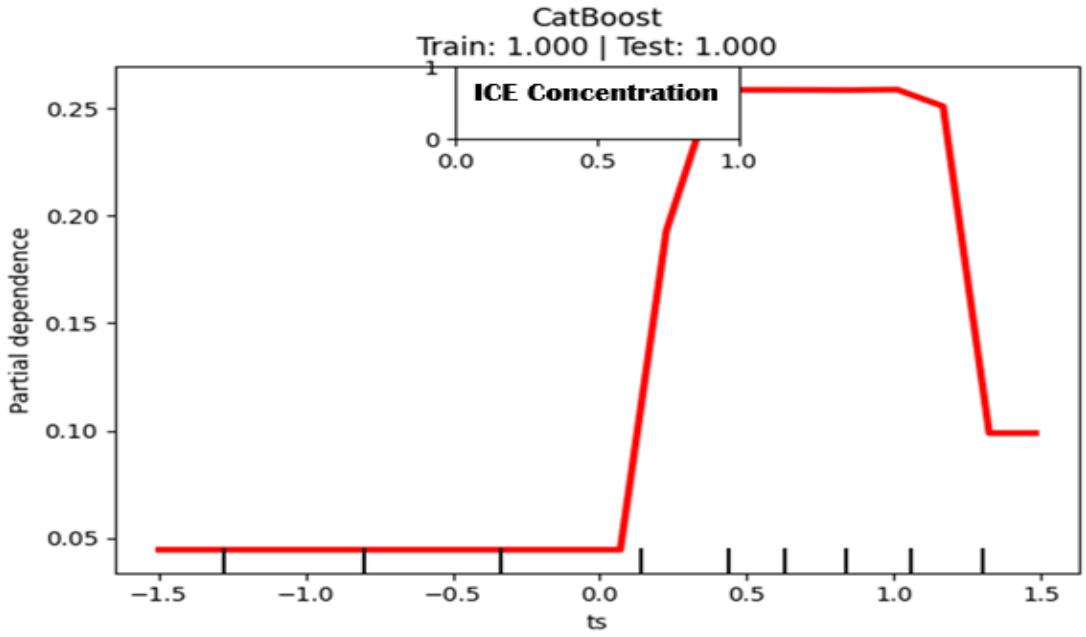

Figure 4-30: ICE Plot for *"ts"*

Because producing PDP and ICE plots is very resource intensive, using the same process for all features would be computationally infeasible. ICE plots specific to the “ts” feature gives insight into how the methodology can be adapted for the other features in order to improve interpretability.

Chapter 5—Discussion and Conclusions

5.1 Discussion

As Industrial Internet of Things (IIoT) are increasingly connected, they become critical for the automation and efficiency of important infrastructures. While interconnected physical devices can aid in optimizing operations, they tend to run on lightweight operating systems. This leaves little scope for integrated security which makes them vulnerable to cyberattacks. These cyberattacks impact essential services like the power distribution system. Since these systems support day-to-day operations, they must be secured, not simply optional. The study aims to develop Machine Learning (ML) models for industrial environments. This means having systems that can detect and counteract cyber threats quickly and efficiently, enabling greater resiliency of our critical infrastructure that we rely on every day.

5.2 Conclusions

Answering the research questions was the heart of this study. By carefully examining each question and linking it to empirical results, we could confidently accept or reject the related hypotheses and reinforce the thesis that ML with industry-specific features bolsters the resilience of electrical control systems against cyber threats.

With regard to RQ1, the results of the ANOVA and eta-squared (η^2) tests provided strong evidence that both the selected ML algorithms and industry-specific features contributed to the models' ability to achieve exceptionally high binary classification accuracy, reaching up to 99.99%. While the ML algorithms accounted for the majority of the performance improvement, the industry-specific features, such as those outlined in Table 2-3, played a supporting role in enhancing predictive capability.

Therefore, these two factors support the first hypothesis that adding industry-specific features to underutilized ML models can improve performance regarding intrusion detection in industrial settings.

For RQ2, the evaluation centered on comparing the performance of models trained on industry-specific features (CICAPTIIoT2024) with results from prior literature that utilized generic IoT datasets. As presented in Table 4-7, the models developed in this study consistently outperformed the average benchmarks reported in earlier studies, achieving approximately a 10% improvement across key performance metrics. Despite a marginal difference of about 1% when compared to the highest-performing models reported in the literature, the models in this study demonstrated exceptional outcomes, achieving over 99.99% in Precision and Accuracy, and reaching up to 99.99% in Recall and F1-Score. These findings strongly support the second hypothesis that machine learning models trained on industry-specific features offer superior predictive accuracy and greater generalizability for cyber intrusion detection in IIoT environments.

In addressing RQ3, the integration of top-scoring feature groups with various ML algorithms did not significantly improve performance. While the findings do not provide sufficient evidence to support the third hypothesis, the consistency observed in the False Positive Rate (FPR) reinforces confidence in the models' accuracy during the dimensionality reduction process. This stability is crucial for deploying machine learning models in resource-constrained industrial environments, where efficiency and reliability are critical.

In conclusion, the evidence gathered here affirms that tailoring ML toward the unique characteristics of industrial environments is vital. Although the third hypothesis

was not significantly supported, the exploration of research questions not only produced meaningful insights but also reinforced the coherence and validity of the central argument.

5.3 Contributions to Body of Knowledge

This study offers several contributions to the evolving field of ML-based cybersecurity in industrial environments, particularly in protecting electrical control systems through data-driven approaches:

1. **Industry-Specific Feature Analysis:** This work comprehensively examined features from the recently published industrial IoT dataset and evaluated whether these features can improve binary classification performance for cyber intrusion detection, especially compared to models trained on generic IoT datasets.

2. **Evaluation of Less-Common Machine Learning Algorithms:** The study focused on ML algorithms that were less frequently applied in previous studies. Models were combined with feature subsets obtained through various feature selection techniques to study their potential for improving detection performance.

3. **Handling of Highly Imbalanced Data:** The challenges of imbalanced datasets were addressed with special attention to the CICAPTIIoT2024 dataset. Balancing techniques such as SMOTE were carefully applied, and the integrity of the augmented datasets was validated through quantitative assessments to ensure consistent, trustworthy results after model training.

5.4 Recommendations for Future Research

Future work must close the gaps found in the CICAPTIIoT2024 datasets. Some features are set with fixed values and are not helpful for training models. This suggests a demand for even better-quality data acquired from actual industrial devices. Although availability is restricted by proprietary and security limitations, bringing more realism and diversity to simulated datasets is a promising way forward. Another primary emphasis is the collection of data from the environments where Supervisory Control and Data Acquisition (SCADA) protocols are currently being implemented. These communication protocols serve as a primary factor in determining the behavior of devices and establishing a stable baseline for normal industrial processes.

Additionally, future research is needed to rigorously examine and verify various techniques for handling class imbalance, along with clear measures to prevent data quality compromise and bias. Lastly, a more thorough investigation of explainable AI (xAI) is suggested. This would call for a fixed high-performance computing platform because analysis involves much computational effort

References

International Energy Agency (IEA) (2021). *Enhancing cyber resilience in electricity systems*, IEA, Paris https://www.iea.org/reports/enhancing-cyber-resilience-in-electricity-systems, Licence: CC BY 4.0

Ghiasvand, E., Ray, S., Iqbal, S., Dadkhah, S., & Ghorbani, A. A. (2024). CICAPT-IIOT: A provenance-based APT attack dataset for IIoT environment. *arXiv preprint arXiv:2407.11278*.

Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: *A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment*. Sensors 2023, 23, 5941. https://doi.org/10.3390/s23135941

The 2003 Northeast Blackout: A Case Study for the Learner. (2015). *George Mason University*. https://cip.gmu.edu/wp-content/uploads/2015/08/Blackout-Learner-Version.pdf

Contributors, W. (2024). Texas City refinery explosion. *Wikipedia*. https://en.wikipedia.org/wiki/Texas_City_refinery_explosion

Edwards, J. (2024). *Jen Easterly Breaks Down President's FY25 Budget for CISA at House Subcommittee Hearing*. https://executivegov.com/2024/05/jen-easterly-details-presidents-fy25-budget-request-for-cisa-before-house-subcommittee/

Business Blackout: The Insurance Implications of a Cyber Attack on the US Power Grid. (2015). *Lloyd's*. https://www.jbs.cam.ac.uk/wp-content/uploads/2020/08/crs-lloyds-business-blackout-scenario.pdf

Cyber threat and vulnerability analysis of the U.S. electric sector: Mission Support Center analysis report. (2016). *Idaho National Laboratory*

Yan Xu. (2020). A review of cyber security risks of power systems: from static to dynamic false data attacks. *Prot Control Mod Power Syst* **5**, 19. https://doi.org/10.1186/s41601-020-00164-w

Easterly, J. (2024). Opening statement by CISA Director Jen Easterly. *U.S. Department of Homeland Security*. https://www.cisa.gov/news-events/news/opening-statement-cisa-director-jen-easterly

Yuldashev, T., & Solovev, A. (2022). *Best Practices on Embedded Firmware Development: Typical Challenges and Solutions*. https://www.integrasources.com/blog/embedded-firmware-development-practices-challenges-solutions/

Hareesh, R., Kalluri, R., Mahendra, L., Kumar, R. K., & Bindhumadhava, B. S. (2020). Passive Security Monitoring for IEC-60870-5-104 based SCADA Systems. *International Journal of Industrial Control Systems Security (IJICSS)*, *3*(1). https://infonomics-society.org/wp-content/uploads/Passive-Security-Monitoring-for-IEC-60870-5-104-based-SCADA-Systems.pdf

Mitigations for Security Vulnerabilities Found in Control System Networks. (n.d.). https://www.cisa.gov/sites/default/files/2023-01/MitigationsForVulnerabilitiesCSNetsISA_S508C.pdf

Gan, C., Lin, J., Huang, D.-W., Zhu, Q., & Tian, L. (2023). Advanced Persistent Threats and Their Defense Methods in Industrial Internet of Things: A Survey. *Mathematics*, *11*(14), 3115. https://doi.org/10.3390/math11143115

Singamaneni, K., & Singh, S. (2024). *A Comprehensive Survey on Advanced Persistent Threat (APT) Detection Techniques*. *80*(2), 2675–2719. https://doi.org/10.1016/j.ijcip.2024.100598

Easterly, J., & Fanning, T. (2023). The Attack on Colonial Pipeline: What We've Learned & What We've Done Over the Past Two Years. *Cybersecurity and Infrastructure Security Agency (CISA)*.

Oladimeji, S., & Kerner, S. M. (2023). SolarWinds hack explained: Everything you need to know. *TechTarget*.

MITRE ATT&CK. (n.d.). https://attack.mitre.org/groups/G0016/

Greenberg, A. (2019). *Sandworm: A new era of cyberwar and the hunt for the Kremlin's most dangerous hackers*.

What are Advanced Persistent Threats? (n.d.). https://www.cybereason.com/fundamentals/what-are-advanced-persistent-threats

Yan, S., Han, H., Dong, X., & Xu, Z. (2024). Lightweight deep learning method based on group convolution: Detecting DDoS attacks in IoT environments. *2024 10th International Symposium on System Security, Safety, and Reliability (ISSSR)*, 181–189. https://doi.org/10.1109/ISSSR61934.2024.00027

Patidar, N., Zreiqat, S., Mahesh, S., & Woo, J. (2024). Cyberattack data analysis in IoT environments using big data. *arXiv*. https://doi.org/10.48550/arXiv.2406.10302

Huynh, N. S., De La Cruz, S., & Perez-Pons, A. (2023). Denial-of-Service (DoS) attack detection using edge machine learning. *2023 International Conference on Machine Learning and Applications (ICMLA)*, 1741–1745. https://doi.org/10.1109/ICMLA58977.2023.00264

Pirtama, A., Prasetia, Y., Saputra, R. I., & Winanto, E. A. (2024). Improvement attack detection on Internet of Things using Principal Component Analysis and Random Forest. *Media Journal of General Computer Science, 1*(1), 14–19. https://doi.org/10.62205/mjgcs.v1i1.8

Okey, O. D., Rodriguez, D. Z., & Kleinschmidt, J. H. (2024). Enhancing IoT intrusion detection with federated learning-based CNN-GRU and LSTM-GRU ensembles. *2024 19th International Symposium on Wireless Communication Systems (ISWCS)*, 1–6. https://doi.org/10.1109/ISWCS61526.2024.10639159

Kumar, A. G., Rastogi, A., & Ranga, V. (2024). Evaluation of different machine learning classifiers on new IoT dataset CICIoT2023. *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, 1–6. https://doi.org/10.1109/ISCS61804.2024.10581375

Ramli, K., & Thereza, N. (2023). Development of Intrusion Detection Models for IoT Networks Utilizing CICIoT2023 Dataset. *2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, 66–72. https://doi.org/10.1109/ICON-SONICS59898.2023.10435006

Tseng, S. -M., Wang, Y. -Q., & Wang, Y. -C. (2024). Multi-Class Intrusion Detection Based on Transformer for IoT Networks Using CIC-IoT-2023 Dataset. *Future Internet*, *16*(8), 284. https://doi.org/10.3390/fi16080284

Gheni, H. Q., & Al-Yaseen, W. L. (2024). Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset. *e-Prime - Advances in Electrical Engineering, Electronics and Energy, 9*, 100673. https://doi.org/10.1016/j.prime.2024.100673

Liu, Y., Guo, F., Zhao, Q., & Wu, C. (2024). An intrusion detection model based on a residual memory convolutional neural network with attention mechanism. *Journal of Physics: Conference Series, 2833*, 012009. https://doi.org/10.1088/1742-6596/2833/1/012009

Khanday, S. A., Fatima, H., & Rakesh, N. (2024). A novel data preprocessing model for lightweight sensory IoT intrusion detection. *International Journal of Mathematical, Engineering and Management Sciences, 9*(1), 188–204. https://doi.org/10.33889/IJMEMS.2024.9.1.010

Modi, P. (2024). Towards Efficient Machine Learning Method for IoT DDoS Attack Detection. arXiv preprint arXiv:2408.10267.

Becerra-Suarez, F. L., Tuesta-Monteza, V. A., Mejia-Cabrera, H. I., & Arcila-Diaz, J. (2024). Performance Evaluation of Deep Learning Models for Classifying Cybersecurity Attacks in IoT Networks. *Informatics*, *11*(2), 32. https://doi.org/10.3390/informatics11020032

Gupta, S., Kumar, S., Singh, K., & Saini, D. (2024). Unveiling anomalies: Harnessing machine learning for detection and insights. *Engineering Research Express, 6*(3), 035215. https://doi.org/10.1088/2631-8695/ad66b2

Le, T.-T.-H., Wardhani, R. W., Putranto, D. S. C., Jo, U., & Kim, H. (2023). Toward enhanced attack detection and explanation in intrusion detection system-based IoT environment data. *IEEE Access, 11*, 131661–131676. https://doi.org/10.1109/ACCESS.2023.3336678

Narayan, K. R., Mookherji, S., Odelu, V., Prasath, R., Turlapaty, A. C., & Das, A. K. (2023). IIDS: Design of intelligent intrusion detection system for Internet-of-

Things applications. *2023 IEEE 7th Conference on Information and Communication Technology (CICT)*, 1–6. https://doi.org/10.1109/CICT59886.2023.10455720

Altrad, A. (2023). IoTs traffic detection and analysis using machine learning for cybersecurity application. *2023 IEEE 5th Eurasia Conference on IoT, Communication and Engineering (ECICE)*, 78–83. https://doi.org/10.1109/ECICE59523.2023.10383018

Phan, V. A., Jerabek, J., & Malina, L. (2024). Comparison of multiple feature selection techniques for machine learning-based detection of IoT attacks. *Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES '24)*, 157, 1–10. https://doi.org/10.1145/3664476.3670440

Priya, M. K. V., Glory, H. A., & Sriram, V. S. S. (2025). Enhanced Deep Learning for IIoT Threat Intelligence: Revealing Advanced Persistent Threat Attack Patterns. *Communications in Computer and Information Science*, *2306*. https://doi.org/10.1007/978-981-97-9743-1_15

Alam, M. M., & Wang, W. (2021). A comprehensive survey on the state-of-the-art data provenance approaches for security enforcement. *Journal of Computer Security, 29*(4), 423–446.

Jayasamraj, J. (n.d.). SCADA communication & protocols. *National Power Training Institute (NPTI)*. https://jcboseust.ac.in/assets/electrical/images/notes/ssa_scada_communications_protocols.pdf

Zhou, X., Xu, Z., Wang, L., Chen, K., Chen, C., & Zhang, W. (2018). Behavior Based Anomaly Detection Model in SCADA System. *MATEC Web of Conferences*, *173*, 01011. https://doi.org/10.1051/matecconf/201817301011

7-techniques-to-handle-class-imbalance-in-machine-learning-eb1297419ec9. (n.d.). https://medium.com/@data.pilot/7-techniques-to-handle-class-imbalance-in-machine-learning-eb1297419ec9

Maldonado, S., López, J., & Vairetti, C. (2019). An alternative SMOTE oversampling strategy for high-dimensional datasets. *Applied Soft Computing*, *76*, 380–389. https://doi.org/10.1016/j.asoc.2018.12.024

Ianni, M., & Masciari, E. (2023). SCOUT: Security by computing OUTliers on activity logs. *Computers & Security*, *132*, 103355. https://doi.org/10.1016/j.cose.2023.103355

Handling Missing Values. (2024). *ML*. https://www.geeksforgeeks.org/ml-handling-missing-values/

Punyakeerthi, B. (2024). *Understanding Feature Scaling in Machine Learning*. https://medium.com/@punya8147_26846/understanding-feature-scaling-in-machine-learning-fe2ea8933b66

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

(2008). SciPy API. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html

SMOTE — Version 0.13.0. (n.d.). *https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html#imblearn.over_sampling.SMOTE*

Bordia, A. (2022). *Handling Imbalanced Data by Oversampling with SMOTE and its Variants*. https://medium.com/analytics-vidhya/handling-imbalanced-data-by-oversampling-with-smote-and-its-variants-23a4bf188eaf

Top 12 Dimensionality Reduction Techniques for Machine Learning. (2024). *Encord Blog*. https://encord.com/blog/dimentionality-reduction-techniques-machine-learning/

Should Feature Selection be done before Train-Test Split or after? (2019). *https://stackoverflow.com/questions/56308116/should-feature-selection-be-done-before-train-test-split-or-after*

Brownlee, J. (2020). How to Choose a Feature Selection Method For Machine Learning. *Data Preparation*. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

The Anderson-Darling statistic. (n.d.). https://support.minitab.com/en-us/minitab/help-and-how-to/statistics/basic-statistics/supporting-topics/normality/the-anderson-darling-statistic/

Overview for Fit Regression Model and Linear Regression. (n.d.). https://support.minitab.com/en-us/minitab/help-and-how-to/statistical-modeling/regression/how-to/fit-regression-model/before-you-start/overview/

Shah, S. (2025). *How Much RAM For Machine Learning? Expert Tips For Optimal Performance*. https://yetiai.com/how-much-ram-for-machine-learning/

Berg, T., Ordentlich, O., & Shayevitz, O. (2025). *Memory Complexity of Estimating Entropy and Mutual Information*. arXiv preprint arXiv10.1109/TIT.2025.3547871

Roy, R. (2022). *Mutual Information Score — Feature Selection*. https://bobrupakroy.medium.com/mutual-information-score-feature-selection-8eb19071664b

Feature Selection Based on Mutual Information Gain for Classification and Regression. (2021). https://guhanesvar.medium.com/feature-selection-based-on-mutual-information-gain-for-classification-and-regression-d0f86ea5262a

RFE. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html?utm_source=chatgpt.com

CatBoost v. XGBoost v. LightGBM. (n.d.). https://www.kaggle.com/code/nholloway/catboost-v-xgboost-v-lightgbm

LogisticRegression. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

ExtraTreesClassifier. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html

Narkedimilli, S., Makam, S., Sriram, A. V., Mallellu, S. P., Sathvik, M., & Prasad, R. R. V. (2025). *Enhancing IoT Network Security through Adaptive Curriculum Learning and XAI*. https://doi.org/10.48550/arXiv.2501.11618

Yaras, S., & Dener, M. (2024). IoT-Based Intrusion Detection System Using New Hybrid Deep Learning Algorithm. *Electronics*, *13*, 1053. https://doi.org/10.3390/electronics13061053

Kim, H. Y. (2014). Statistical notes for clinical researchers: Nonparametric statistical methods: 2. Nonparametric methods for comparing three or more groups and repeated measures. *Restorative Dentistry and Endodontics*, *39*(4), 329–332. https://doi.org/10.5395/RDE.2014.39.4.329

Chapman, S. J. (2018). Review of Discovering Statistics Using IBM SPSS Statistics, 4th Edition. *Journal of Political Science Education*, *14*(1), 145–147. https://doi.org/10.1080/15512169.2017.1366328

Bobbitt, Z. (2020). *What is Eta Squared? (Definition & Example)*. https://www.statology.org/eta-squared/

Liu, L., Kim, J. S., Aragam, B., & Ravikumar, P. (2020). https://www.davidinouye.com/publication/inouye-2020-automated/inouye-2020-automated.pdf?utm_source=chatgpt.com. *Automated Dependence Plots*.

Dev, V. (2024). *A Complete Guide to ICE Plots*. https://easyml.medium.com/a-complete-guide-to-ice-plots-037c8e5ac9f1

Hui, J. (2023). Machine Learning — Algorithm. https://jonathan-hui.medium.com/machine-learning-summary-algorithm-d75c64963800

Jha, N. (2024). Understanding Feature Selection Techniques in Machine Learning. https://medium.com/@nirajan_DataAnalyst/understanding-feature-selection-techniques-in-machine-learning-02e2642ef63e

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. *Data Mining,* Inference, and Prediction, Second Edition.

Tunnicliffe, D. (2021). Extra Trees, please. https://medium.com/data-science/extra-trees-please-cec916e24827

XGBoost Parameters — xgboost 3.1.0-dev documentation. (n.d.). https://xgboost.readthedocs.io/en/latest/parameter.html

Tyagi, A. (2018). What is XGBoost Algorithm? https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/?utm_source=chatgpt.com

Parameters — LightGBM 4.6.0.99 documentation. (n.d.). https://lightgbm.readthedocs.io/en/latest/Parameters.html?utm_source=chatgpt.com

CatBoost in Machine Learning. (2021). https://www.geeksforgeeks.org/catboost-ml/?utm_source=chatgpt.com

LightGBM Feature parameters. (2023). https://www.geeksforgeeks.org/lightgbm-feature-parameters/?utm_source=chatgpt.com

Hancock, J., & Khoshgoftaar, T. M. (n.d.). CatBoost for big data: an interdisciplinary review. SpringerOpen. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00369-8?utm_source=chatgpt.com

Common parameters. (n.d.). https://catboost.ai/docs/en/references/training-parameters/common

ElasticNet. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

Grammarly. (2025). Grammarly [AI writing assistant]. https://www.grammarly.com

Microsoft. (2025). Copilot (GPT-4) [Large language model]. https://copilot.microsoft.com/

Appendix A

# of features	ML	PR-AUC	F1-score	Recall	Precision	Accuracy	Confusion Matrix
Machine learning algorithms trained on imbalanced datasets							
52	CatBoost	0.8194	0.7845	0.8659		0.9999	3239661 56 22 142
	XGBoost	0.9984	0.9877	0.9817		1	3239716 1 3 161
	LightGBM	0.6795	0.6810	0.8720		1	3239604 113 21 143
	Elastic Net	0.980	0.95	0.91	1	0.91	5885876 593584 4 297
	ExtraTrees	1	1	1	1	1	647953 7 24 227
Machine learning algorithms trained on SMOTE-balanced datasets							
52	CatBoost	1	0.999	1		0.9999	3239892 145 0 1133598
	XGBoost	1	1	1		1	3240035 2 0 1133598
	LightGBM	1	1	1		1	3240014 23 3 1133595
	Elastic Net	0.980	0.93	0.93		0.93	7825557 813722 22839 3000908
	ExtraTrees	1	1	1	1	1	6479430 29 0 2267811
Features with RFECV rankings greater than 3							
32	CatBoost	1	0.9999	1	1	1	3239884 153 0 1133598
	XGBoost	1	1	1	1	1	3240033 4 0 1133598
	LightGBM	1	1	1	1	1	3240028 9 0 1133598
	Elastic Net	0.987	0.9306	0.9922	0.97	.9617	3081278 158759 8867 1124731
	ExtraTrees	1	1	1	1	1	6479423 36 0 2267811
Features ranked in the top 20 by RFECV							
20	CatBoost	1	0.9999	1	1	1	3239912 125 4 1133594

	XGBoost	1	1	1	1	1	3240033 0	4 1133598
	LightGBM	1	1	1	1	1	3240026 5	11 1133593
	Elastic Net	0.950	0.87	0.86	0.91	0.86	7057380 15942	1581899 3007805
	ExtraTrees	1	1	1	1	1	6479437 0	22 2267811
Features ranked in the top 10 by RFECV								
10	CatBoost	1	0.9991	0.9997	1	0.9995	3238392 386	1645 1133212
	XGBoost	1	1	1	1	1	3240000 3	37 1133595
	LightGBM	1	0.9999	1	1	1	3239956 37	81 1133561
	Elastic Net	0.848	0.77	0.76	0.87	0.76	5886076 48166	2753203 2975581
	ExtraTrees	1	1	1	1	1	6479414 1	45 2267810
Mutual Information-selected features with scores above zero								
43	CatBoost	1	0.9999	1	1	1	3239905 14	132 1133584
	XGBoost	1	1	1	1	1	3240035 0	2 1133598
	LightGBM	1	1	1	1	1	3240024 9	13 1133589
	Elastic Net	0.970	0.89	0.89	0.92	0.89	7344833 20822	1294446 3002925
	ExtraTrees	1	1	1	1	1	6479429 0	30 2267811
Top 20 features selected by Mutual Information								
20	CatBoost	1	0.9998	1	1	0.9999	3239599 17	438 1133581
	XGBoost	1	1	1	1	1	3240023 0	14 1133598
	LightGBM	1	1	1	1	1	3240023 2	14 1133596
	Elastic Net	0.954	0.85	0.84	0.90	0.84	6819068 23810	1820211 2999937
	ExtraTrees	1	1	1	1	1	6479432 0	27 2267811
Top 10 features selected by Mutual Information								

10	CatBoost	0.9999	0.9998	1	1	0.9999	3239616	421
							6	1133592
	XGBoost	1	0.9999	1	1	1	3239924	113
							3	1133595
	LightGBM	0.9999	0.9999	1	1	1	3239949	88
							34	1133564
	Elastic Net	0.873	0.78	0.76	0.87	0.76	5965316	2673963
							67012	2956735
	ExtraTrees	1	1	1	1	1	6479374	85
							0	2267811

Table 1: Machine learning performance metrics using features selected by MI and RFECV on the CICAPTIIoT2024 dataset

# of features	ML	PR-AUC	F1-score	Recall	Precision	Accuracy	Confusion Matrix
Machine learning algorithms trained on imbalanced datasets							
47	CatBoost	1	0.9957	0.9916	0.99	0.99	164617 121 34175 4015869
	XGBoost	1	0.9937	0.9998	0.99	0.9878	113854 50884 609 40494835
	LightGBM	1	0.9921	0.9999	0.98	0.9848	101081 63657 454 4049590
	Elastic Net	0.983	0.94	0.93	0.97	0.93	322084 7369 586049 7514062
	ExtraTrees	0.996	1	1	1	1	316141 13312 22417 8077694
Machine learning algorithms trained on SMOTE-balanced datasets							
47	CatBoost	0.9999	0.9963	0.9939	0.99	0.9939	1413550 5029 24669 4024327
	XGBoost	1	0.9973	0.9963	1	0.9956	1411870 6709 14996 4034000
	LightGBM	1	0.9971	0.9973	1	0.9957	1406025 12554 10736 4038260
	Elastic Net	0.983	0.94	0.94	0.95		3724048 56004 794455 10005693
	ExtraTrees	1	0.9973	0.9946	1	0.996	2834471 568 44407 8055704
Features with RFECV rankings greater than 3							
26	CatBoost	0.9999	0.9964	0.9938	0.99	0.9946	1414420 4159 25104 4023892
	XGBoost	0.9998	0.9935	0.9898	0.99	0.9890	1406929 11650 41491 4007505
	LightGBM	0.9998	0.9930	0.9938	0.99	0.9897	1386904 31675 24903 4024093
	Elastic Net	0.9970	0.9785	0.9610	0.97	0.9687	1405232 13347 157901 3891095
	ExtraTrees	0.9999	0.9965	0.9932	0.99	0.9949	55108 969 55108 8045003
Features ranked in the top 20 by RFECV							
20	CatBoost	0.9999	0.9958	0.9920	0.99	0.9938	1416950 1629 32389 4016607

	XGBoost	0.9999	0.9971	0.9958	1	.9957	1412204 16940	6375 4032056
	LightGBM	1	0.9969	0.9972	1	0.9954	1404946 11342	13633 4037654
	Elastic Net	0.977	0.9684	0.9678	0.95	0.95	3445181 347747	334871 10452401
	ExtraTrees	1	0.9974	0.9950	1	0.996	2834188 41300	851 8058811
Features ranked in the top 10 by RFECV								
10	CatBoost	0.9993	0.9891	0.9795	0.98	0.9839	1413790 83014	4789 3965982
	XGBoost	0.9996	0.9923	0.9883	0.99	0.9887	1404193 47519	14386 4001477
	LightGBM	0.9994	0.9897	0.9900	0.98	0.9847	1375372 40387	43207 4008609
	Elastic Net	0.939	0.9329	0.9231	0.90	0.90	3176457 830168	603595 9969980
	ExtraTrees	0.999	0.9936	0.9880	0.99	0.990	2829605 99517	5434 8000594
Mutual Information-selected features with scores above zero								
35	CatBoost	1	0.9963	0.9937	0.99	0.9946	1414345 25432	4234 4023564
	XGBoost	1	0.9973	0.9963	1	0.9960	1411702 14999	6877 4033997
	LightGBM	1	0.9970	0.9972	1	0.9956	1405875 11191	12704 4037805
	Elastic Net	0.983	0.94	0.94	0.94	0.9829	3759301 858700	20751 9941448
	ExtraTrees	1	0.9973	0.9947	1	0.996	2834419 43801	620 8056310
Top 20 features selected by Mutual Information								
20	CatBoost	0.9999	0.9960	0.9937	0.99	0.9934	1411804 25418	6775 4023578
	XGBoost	0.9999	0.9966	0.9951	0.99	0.9944	1411280 19996	7299 4029000
	LightGBM	0.9999	0.9968	0.9952	0.99	0.9953	1412057 19392	6522 4029604
	Elastic Net	0.986	0.94	0.94	0.95	0.96	3750679 818250	29373 9981898
	ExtraTrees	1	0.9971	0.9944	1	0.996	2833832 46614	1207 8053497
Top 10 features selected by Mutual Information								

10	CatBoost	0.9997	0.9943	0.9900	0.99	0.9910	1413177 40400	5402 4008596
	XGBoost	0.9950	0.9950	0.9918	0.99	0.9917	1411827 33388	6752 4015608
	LightGBM	0.9998	0.9951	0.9920	0.99	0.9928	1411940 32537	6639 4016459
	Elastic Net	0.977	0.95	0.95	0.95	0.9767	3373004 309689	407048 10490459
	ExtraTrees	1	0.9959	0.9922	0.99	0.994	2832958 63868	2081 8036243

Table 2: Machine learning performance metrics using features selected by MI and RFECV on the CICIoT2023 dataset