# An Empirical Study on Learning Models and Data Augmentation for IoT Anomaly Detection

**Alireza Toghiani Khorasgani**

**A Thesis**

**in**

**The Department**

**of**

**Concordia Institute for Information Systems Engineering**

**Presented in Partial Fulfillment of the Requirements**

**for the Degree of**

**Master of Applied Science (Information Systems Security) at**

**Concordia University**

**Montréal, Québec, Canada**

**March 2025**

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Alireza Toghiani Khorasgani**

Entitled: **An Empirical Study on Learning Models and Data Augmentation for IoT Anomaly Detection**

and submitted in partial fulfillment of the requirements for the degree of

**Master of Applied Science (Information Systems Security)**

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
*Dr. Mohsen Ghafouri*

_____ Examiner
*Dr. Jun Yan*

_____ Supervisor
*Dr. Suryadipta Majumdar*

_____ Co-supervisor
*Dr. Paria Shirani*

Approved by _____
Dr. Chun Wang, Director
Department of Concordia Institute for Information Systems Engineering

_____ 2025

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

# Abstract

An Empirical Study on Learning Models and Data Augmentation for IoT Anomaly Detection

Alireza Toghiani Khorasgani

This thesis studies the application and impact of deep learning methods in anomaly detection, a critical area within security applications. While deep learning's popularity is driven by its perceived ability to manage complex patterns in large datasets and perform feature engineering inherently, this thesis questions these assumptions. By revisiting feature selection and data augmentation techniques, this research evaluates their effectiveness in improving the performance of deep-learning-based anomaly detection methods. Furthermore, it examines the impact of other essential factors such as model choice (both traditional machine learning and deep learning), data balancing, and hyperparameter tuning on anomaly detection performance.

From these investigations, the thesis reports that the common beliefs surrounding deep learning are not universally valid, highlighting the need for a framework to evaluate the usefulness of features and data for specific cases. To address this gap, a new framework is proposed, guiding data users and anomaly detection tools toward optimal configurations, including feature selection, model selection, hyperparameters, and data augmentation techniques. The effectiveness of this framework is demonstrated using two major IoT datasets, offering insights into improving anomaly detection systems through strategic and evidence-based approaches.

# Acknowledgments

I want to express my gratitude to Dr. Paria Shirani and Dr. Suryadipta Majumdar, who have served as my supervisors throughout my graduate studies. Their advice and original ideas have improved my work. Dr. Suryadipta Majumdar and Dr. Paria Shirani were open to new research ideas and welcomed them. The financial support I have received from my supervisors is appreciated.

I want to express my deepest gratitude to my family and friends for their unwavering love, support, and inspiration throughout this challenging journey. Their constant encouragement has been my source of strength and motivation, propelling me forward even in the face of adversity. Without their support, I could not have achieved this significant milestone.

I am particularly grateful for the role my friends played in supporting my mental health and well-being. Their presence, understanding, and encouragement were crucial in helping me navigate the ups and downs of this research journey. Through the most difficult times, they were there to listen, offer advice, and provide a much-needed sense of perspective and balance.

This journey has taught me the true value of friendship and support. It has shown me that even in the most challenging of circumstances, the love and encouragement of those closest to us can help us overcome any obstacle. I am fortunate to have such an incredible support system, and I know that I could not have reached this point without them. To my friends and family, I want to express my heartfelt gratitude and appreciation for being my pillars of strength throughout this incredible journey.

A special thanks to Arvin Asrari, Hirad Babayan, Yassaman Ommi (whose help with my LaTeX issues and challenges was invaluable), Matin Yousefabadi, Luna Ettihad, Mohammadreza Tayaranian, Anthony Andreoli, Hugo Kermabon, Anis Lounis and so many other friends whom I cannot

name due to space constraints but who have all played an essential role in my journey. Your unwavering support and kindness have meant the world to me, and I am forever grateful. And finally, I would also like to acknowledge Matcha Babayan, a beloved cat who provided much-needed emotional support during hard times back in December before my defense.

I am grateful to be a member of the CISR Lab under the leadership of Dr. Majumdar. I express my gratitude to my talented, innovative, and hardworking colleagues at the CISR Lab for their support and collaboration throughout the different phases of this thesis.

Finally, I am also grateful to the department and the university for providing me with the resources and opportunities that made this research possible. I appreciate your contributions to my academic and personal growth.

As I close this chapter, these words resonate deeply, reminding me of the resilience and growth this journey has brought me. Moving forward, I carry with me the strength, the lessons, and the support of those who stood by me, ready to face whatever comes next.

Through darkest nights, I'd nearly fall;
Some tore me down, stripped trust to thrall.
But voices true held firm and near,
They stirred my strength, my path made clear.
In time, I learned to rise and fight,
To claim my truth, to guard my right.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| IoT | Internet of Things |
| CIC | Canadian Institute for Cybersecurity |
| CNN | Convolutional Neural Networks |
| NN | Neural Network |
| AE | Auto Encoder |
| IF | Isolation Forest |
| RF | Random Forest |
| SMOTE | Synthetic Minority Oversampling Technique |
| MIC | Maximal Information Coefficient |
| G-mean | Geometric Mean |
| BERT | Bidirectional Encoder Representations from Transformers |
| XGBoost | eXtreme Gradient Boosting |
| DDoS | Distributed Denial of Service |
| DoS | Denial of Service |
| SFS | Sequential Forward Selection |
| SBS | Sequential Backward Selection |
| LRS | Plus L Minus R Selection |
| PSO | Particle Swarm Optimization |
| DL | Deep Learning |
| ML | Machine Learning |

| | |
|---|---|
| PCA | Principal Component Analysis |
| FPR | False Positive Rate |
| TNR | True Negative Rate |
| TPR | True Positive Rate |
| ROC | Receiver Operating Characteristic |
| AUC | Area Under the Curve |
| ARD | Automatic Relevance Determination |
| G-Mean | Geometric Mean |
| SKB | SelectKBest |
| CMD | Co-analyzed Malware Detection |

# Chapter 1

# Introduction

This chapter describes the context, motivations, problem statements, and contributions of this thesis, providing a comprehensive overview of this thesis in IoT anomaly detection.

## 1.1 Context and Motivation

Deep Learning (DL) techniques have become increasingly prevalent in anomaly detection applications (e.g., [1–5]). This widespread adoption is primarily driven by two common beliefs in the field: (i) deep learning's capability to manage complicated patterns within large datasets, and (ii) its perceived ability to eliminate the need for separate feature engineering since it is inherently handled within the model learning process. However, obtaining large-scale training datasets, which are traditionally considered essential for achieving better-performing anomaly detection models, remains one of the most significant challenges [6, 7].

Among many security applications, anomaly detection stands out as one of the biggest users of deep learning methods. The field has witnessed varied approaches to feature selection and data augmentation, with some studies advocating for deep learning without explicit feature selection [2, 4, 8, 9], while others incorporate it [10, 11]. This divergence in approaches raises important questions about the fundamental assumptions driving the field.

The primary motivation for this research stems from the increasing deployment of IoT devices

in critical infrastructure and everyday applications, where security breaches can have severe consequences. IoT is becoming ubiquitous, with its integration into various sectors, including healthcare, smart homes, and urban infrastructures. This includes diverse populations, such as infants and seniors, who are increasingly reliant on IoT devices for safety, monitoring, and healthcare. As these devices become more embedded in daily life, the need for robust security solutions becomes even more urgent to protect vulnerable citizens and smart societies. Recent studies have shown that IoT-based attacks have increased exponentially in recent years [12], with anomaly detection serving as a crucial defense mechanism. While deep learning approaches show promise, their application in IoT environments faces unique challenges due to resource constraints and diverse device characteristics [13]. Additionally, anomaly detection often encounters the problem of overfitting, which occurs when models capture irrelevant patterns or noise from the training data rather than learning generalizable relationships. This issue arises due to redundant or unnecessary features, as well as significant data imbalance—normal (benign) data typically far outweighs attack (anomalous) data. Consequently, models might achieve high accuracy on the majority class (normal data) while failing to reliably detect anomalies in new, unseen data. Addressing these challenges requires systematic feature selection and effective data balancing strategies to ensure robust anomaly detection performance across diverse applications. Several other factors contribute to overfitting in anomaly detection models. Insufficient training data, especially in IoT environments, limits model generalization, as rare attack instances may not be well-represented. Noisy data can mislead the model by introducing irrelevant or erroneous patterns, while overly complex models may memorize specific details of the training data rather than learning generalizable features. Additionally, data leakage, where information from outside the training set affects the model, can inflate performance during training but reduce its effectiveness on unseen data. Proper handling of these factors is crucial for improving model robustness and preventing overfitting.

Current approaches often apply deep learning techniques without systematic evaluation of feature selection and data augmentation strategies, potentially leading to suboptimal performance or unnecessary computational overhead. This research aims to bridge this critical gap by providing empirical evidence and practical guidelines for implementing effective anomaly detection in IoT environments.

## 1.2 Problem Statement

The proliferation of Internet of Things (IoT) devices has led to an exponential increase in network traffic and, consequently, security threats. This rapid expansion presents unique challenges for anomaly detection systems, particularly in the context of deep learning applications. This thesis identifies several critical challenges in the current landscape of IoT anomaly detection:

(1) **Feature Selection Complexity:**

- Current deep learning approaches often bypass explicit feature selection, assuming the neural networks will automatically learn relevant features
- This assumption remains untested across different IoT scenarios and data characteristics
- The lack of systematic feature selection may lead to:
  - Increased computational overhead
  - Reduced model interpretability
  - Potential degradation in detection accuracy
  - Difficulty in handling diverse IoT device characteristics

(2) **Data Augmentation Challenges:**

- The prevailing belief that larger datasets invariably improve model performance overlooks crucial factors:
  - Data quality and relevance
  - Computational costs of processing enlarged datasets
  - Potential introduction of noise or bias
  - Impact on model generalization
- Current approaches lack systematic methods to evaluate when and how data augmentation should be applied
- The relationship between data quantity and model performance remains poorly understood in IoT contexts

(3) **Configuration Optimization Difficulties:**

- Existing systems often treat feature selection, data augmentation, and model hyperparameters as independent concerns
- The interdependencies between these elements are not well understood or addressed
- Current approaches lack:
  - Systematic methods for joint optimization
  - Clear guidelines for practitioners
  - Adaptability to different IoT scenarios
  - Efficient ways to balance accuracy and computational resources

These challenges lead us to two fundamental research questions:

- *Is avoiding feature selection always useful?*: While avoiding feature selection brings more automation and convenience to the anomaly detection process, it may not always result in optimal model performance. The relationship between feature selection and model effectiveness needs systematic investigation.
- *Is augmenting data always useful?*: Although adding more data can enhance model performance in some cases, the impact of data augmentation isn't uniformly positive. There are scenarios where augmented data might negatively affect model performance, necessitating a more nuanced approach.

## 1.3 Research Objectives

To address these challenges, we establish the following research objectives:

(1) Conduct a systematic investigation of feature selection's impact on deep learning and machine learning based anomaly detection models

(2) Evaluate the relationship between data augmentation and model performance across different scenarios

(3) Develop a framework that can optimize the configuration of anomaly detection systems

(4) Validate the effectiveness of our approach using real-world IoT datasets

## 1.4 Thesis Contributions

This thesis considers a security context and address the above-mentioned two questions to provide a guideline for existing anomaly detection tools on how to decide on feature selection and data augmentation along with several other critical configurations (e.g., hyperparameters, balanced data, models) that impact their performance (both in accuracy and efficiency). Specifically, this thesis first examines the impact of different combinations of feature selections, data balancing, and other factors on various models' performance. Next, it selects the best combinations, analyze the impact of data augmentation on the performance of the selected models, and suggest whether to augment the data through the use of data complexity measurements. Then, it builds a framework, namely, AMETIS (named after **A**thena and **Metis**, the symbols of deep and strategic decision-making), that can suggest the best scenarios for a given dataset. Finally, using two public IoT datasets (CICIoT2023 [14] and IoT-23 [15]), it evaluates the effectiveness of the proposed framework in assisting the existing anomaly detection tools.

The main contributions of this thesis are as follows:

- As per our knowledge, we are the first to study the wide applicability of two common beliefs (i.e., big need of augmented data and no need of feature engineering) on deep learning methods for anomaly detection and show that those beliefs are not always applicable for the performance of existing anomaly detection approaches.

- Based on the key findings of our study, we propose a framework that aims to assist existing anomaly detection approaches in choosing on features and data. The proposed framework provides several DL/ML models along with different feature selection methods in a flexible manner, where a user can simply choose any combinations to train and test their desired models on their own dataset and examines different accuracy metrics to decide whether a given dataset is helpful for data augmentation.

- We evaluate our proposed framework using two large IoT datasets (i.e., CICIoT2023 with over 100 million network flow records and IoT-23 with approximately 20 million captured packets), six deep/machine learning techniques (including BERT and autoencoder), three major feature selection methods (i.e., filter, wrapper, and embedded) along with ten different evaluation setups depicting various combinations of techniques applied on anomaly detection to demonstrate its effectiveness in choosing the best combination of features and augmented data.

- The source code of our framework, along with evaluation setups and documentation, is publicly available[1].

These contributions collectively address the challenges identified in our problem statement by providing both theoretical insights and practical tools for improving IoT anomaly detection systems. This thesis bridges the gap between theoretical understanding and practical implementation, offering concrete solutions for practitioners while advancing the academic state-of-the-art.

## 1.5 Impact and Societal Benefits

This research provides several key benefits to both the academic community and industry practitioners:

- **Resource Optimization:** By challenging the conventional belief that more data and features always lead to better performance, this work helps organizations optimize their computational resources and reduce energy consumption in IoT deployments [16].

- **Enhanced Security:** The AMETIS framework provides practitioners with evidence-based guidelines for implementing more effective anomaly detection systems, potentially reducing the risk of security breaches in critical IoT infrastructure [17].

- **Accessibility:** Through the public release of our framework and comprehensive documentation, we enable smaller organizations and researchers to implement sophisticated anomaly detection systems without extensive trial and error [18].

[1] https://github.com/OCyberLab/Ametis

The insights and tools developed through this research directly contribute to more efficient and effective security measures for IoT ecosystems, which are increasingly integral to critical infrastructure, healthcare systems, and smart city initiatives [12].

## 1.6 Prior Publications

Our work [18] about studying feature selections and data augmentation techniques for anomaly detection published in the IEEE Conference on Communications and Network Security Conference.

> **A. Toghiani Khorasgani**, P. Shirani and S. Majumdar, "An Empirical Study on Learning Models and Data Augmentation for IoT Anomaly Detection" in IEEE Conference on Communications and Network Security Conference,

## 1.7 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 provides background concepts and reviews related work in anomaly detection, feature selection, and data augmentation.

Chapter 3 outlines our methodology and findings related to feature selection optimization. It also introduces the AMETIS framework, detailing its architecture, core components, and its role in improving anomaly detection.

Chapter 4 focuses on our methodology for data augmentation and complexity analysis, providing a detailed exploration of their impact on anomaly detection and the factors influencing their effectiveness.

Chapter 5 presents comprehensive experimental results and analysis.

Chapter 6 discusses various aspects of our approach, including limitations, practical implications, and concludes the thesis while outlining future research directions.

# Chapter 2

# Background and Related Work

This chapter provides a comprehensive examination of both the fundamental concepts and the current state of research in IoT anomaly detection, with particular emphasis on feature selection and data augmentation techniques. We begin by establishing the technical foundation through an exploration of core concepts, followed by a critical analysis of existing research approaches and their limitations. This systematic review enables us to identify research gaps and position our contributions within the broader context of IoT security research.

## 2.1 Background

The effectiveness of machine learning and deep learning models in IoT security heavily depends on the quality and preprocessing of input data. This section examines two fundamental aspects that can impact model performance: feature selection methods and data complexity metrics. Understanding these concepts is helpful for appreciating the challenges and opportunities in developing robust anomaly detection systems.

### 2.1.1 Feature Extraction

Feature extraction transforms raw data into meaningful representations through mathematical or statistical operations [19]. In IoT security contexts, this process converts raw network traffic data into higher-level features that capture important behavioral patterns. The key approaches to feature

extraction include:

### Statistical Feature Extraction

Statistical feature extraction derives numerical measurements that characterize data distributions and patterns [20]. In network security, this involves computing basic statistics like mean, variance, and standard deviation of packet sizes, flow durations, and inter-arrival times. More advanced distribution metrics, such as skewness and kurtosis, provide insights into traffic variations, while time-window statistics capture rolling averages and variances over different time intervals. Additionally, entropy-based features measure the randomness or predictability of network traffic, helping to detect anomalies.

### Signal Processing Based Extraction

Signal processing techniques transform time-domain data into alternative representations to reveal hidden traffic patterns [21]. Fourier transforms convert time-series data into the frequency domain, uncovering periodic traffic behaviors. Wavelet transforms enable multi-resolution analysis, allowing simultaneous examination of short-term and long-term patterns. Spectral analysis identifies key frequency components in network activity, while time-frequency analysis combines both temporal and frequency domain insights to detect anomalous behaviors.

### Domain-Specific Feature Extraction

Domain-specific feature extraction leverages expert knowledge of network protocols and IoT behavior patterns [22]. Protocol-based features are derived from specific protocol headers and payloads, providing insight into communication structures. Flow-level features capture characteristics of network flows, such as session duration and byte distribution, while connection-based features focus on the relationships between devices, identifying peer-to-peer or hierarchical network structures. Lastly, behavioral features model device-specific operational patterns to detect deviations that may indicate cyber threats.

**Dimensionality Reduction Based Feature Extraction**

These methods create lower-dimensional representations of data while preserving key information [23]. t-Distributed Stochastic Neighbor Embedding (t-SNE) [24] and Uniform Manifold Approximation and Projection (UMAP) [25] perform non-linear dimensionality reduction, preserving local structures and relationships in high-dimensional data. Linear Discriminant Analysis (LDA) [26] is another linear technique that not only reduces dimensionality but also enhances class separability. LDA projects the data onto a lower-dimensional space by maximizing the ratio of between-class variance to within-class variance, making it useful for supervised anomaly detection when class labels are available. Matrix factorization techniques, such as Singular Value Decomposition (SVD) [27], decompose data into latent components, enabling efficient feature extraction for anomaly detection. Principal Component Analysis (PCA) is another powerful linear technique used for dimensionality reduction. PCA transforms correlated features into a set of linearly uncorrelated principal components[28]. By focusing on the first few components, which usually contain most of the variance in the data, PCA reduces the feature space while retaining significant information. This method helps eliminate less informative features, often considered as noise or redundancy, thus simplifying the data without substantial loss of information. PCA is widely used in anomaly detection as it improves model efficiency by reducing dimensionality and enhancing model interpretability.

### 2.1.2 Feature Selection

In the field of machine learning and deep learning, the quality and relevance of input data have a major effect on model performance. This requires us to discuss the subject of feature selection and its role in improving the efficiency of deep learning models, particularly in anomaly detection applications.

Feature selection enables models to focus on the most important data, and thus improve their performance. Its procedure effectively reduces the feature set by removing unnecessary or duplicate features that could negatively affect learning or even confuse the model. From basic statistical tests to more advanced procedures, We select a variety of methodologies that have been used in the literature. In this thesis we categorized them into three groups filter methods [29], wrapper methods

[30], and embedded methods [31], each offering distinct advantages and trade-offs.

### Filter Methods

Filter methods assess the relevance of features independent of any prediction model using intrinsic data properties and statistical relationships with the target variable. This allows quick screening of irrelevant inputs before costly model training. The following are some notable filter methods:

- **Mutual Information (MI)**: MI quantifies the mutual dependence between variables based on entropy reduction. High MI score signifies that a feature substantially decreases the uncertainty regarding the value of the target variable. This is particularly effective in identifying nonlinear statistical associations that are often overlooked by linear correlation metrics [32].
- **Trank (T-test Ranking)**: Trank is a statistical test that evaluates the difference between the means of two groups. In feature selection, features are ranked by the t-statistic between class distributions. Features with means significantly differing across classes are considered informative [33].
- **Chi-Square Test ($\chi^2$)**: This method tests if the distribution of categorical feature values is significantly associated with classes. High $\chi^2$ implies dependence between that feature and a target variable [34].
- **Correlation Coefficient**: A correlation coefficient [35] measures the linear relationship between two variables, quantified by statistics like the Pearson correlation which assesses covariance normalized by variance. Features with a high correlation with the target variable are considered valuable, while those highly correlated with other features might be redundant [36].
- **SelectKBest (SKB)**: SKB reduces dimensionality by retaining the top $k$ features with the highest scores. Features are ranked based on ANOVA F-values, and those with the strongest classification power are selected while redundant or noisy features are removed [37].

### Wrapper Methods

Wrapper methods assess subsets of features based on the combined performance of the group within a specific machine learning model [30]. Unlike filter methods that score features independently, wrappers directly search for an optimal subset catered to the intricacies of the model via cross-validation. This model-dependent approach often leads to superior feature selection and accuracy compared to filter methods at the expense of a higher computational load [38]. The customized selections help reduce overfitting and improve predictions but require an efficient search of the exponential feature space, using techniques like sequential selection or evolutionary algorithms [39]. In essence, wrapper methods find feature interactions tailored to individual models by evaluating performance directly within the model, not just by static properties. The following are some notable wrapper methods:

- **Support Vector Machines - Recursive Feature Elimination (SVM-RFE)**: SVM-RFE employs a backward elimination procedure. It ranks features based on their weight magnitudes in the SVM, iteratively removing the least important feature and retraining the model [40]. The reason for specifically using SVM in this context is twofold. First, SVMs are effective in high-dimensional spaces, making them suitable for datasets with a large number of features. Second, SVMs inherently provide a ranking of features based on their contribution to the decision boundary, with the weight vector in the SVM model serving as a direct indicator of the importance of the feature. This inherent ability of SVMs to quantify feature importance makes them particularly suited for recursive feature elimination, where such rankings are crucial.
- **Random Forest (RF)**: Random Forest is an ensemble technique built on decision trees. Features are ranked based on the average reduction in impurity they cause across all trees within the forest [41].
- **Sequential Forward Search (SFS)**: SFS starts with an empty set of features and adds features one by one until an optimal set is obtained. This method is particularly useful when dealing with large feature spaces [42].

- **Sequential Backward Search (SBS)**: SBS [43] begins with the full set of features and removes them one by one. It is effective in reducing the dimensionality of the feature space while maintaining the performance of the model [44].
- **Plus L Minus R Selection (LRS)**: LRS [45] is a variant of SFS and SBS, adding the 'L' best features and removing the 'R' worst features in each iteration. For example, with L = 2 and R = 1, it adds the two best features and removes one worst feature per iteration, offering a balanced approach to feature selection.
- **Particle Swarm Optimization (PSO)**: PSO [46] is an evolutionary computation technique used as a wrapper method for feature selection. It optimizes a predefined objective function (such as classification accuracy) by iteratively updating the positions (feature subsets) of a swarm of particles (candidate solutions) [47].

**Embedded Methods**

Embedded methods combine elements of filter and wrapper approaches for efficient feature selection during model training. Rather than preprocessing, embedded methods account for feature interactions within the learning phase itself [31]. This avoids the generalized rankings of filters and the high computational expense of wrappers by integrating selection directly into optimization. For example, regularization methods such as L1 (lasso) regularization explicitly drive coefficient weights to zero, effectively eliminating weak features [48]. Other techniques such as ridge regression control the magnitude rather than the number of features, shrinking the excessive correlation while retaining predictors [31]. The key benefit of embedded feature selection is finding model-specific subsets without the extreme cost of exhaustive searches, enabling generalization and stability. The key examples of embedded methods are:

- **Lasso Regularization (L1).** L1 regularization, introduces a penalty on the absolute magnitudes of the model coefficients. This method is distinct for its capacity to reduce certain coefficients to zero, thereby inherently integrating feature selection within the model training itself. Instead of treating feature selection as an isolated process, L1 regularization seamlessly blends it into model optimization, effectively striking a balance between model complexity and performance [48].

- **Ridge Regularization (L2).** L2 regularization [49] imposes a penalty on the square of the model coefficients. Contrary to L1 regularization, L2 does not encourage coefficients to shrink to zero but rather to become smaller. It is effective in dealing with multicollinearity, where several predictors are correlated. By penalizing the square of coefficients, L2 regularization aims to distribute the importance more evenly across features, which helps in reducing overfitting.
- **Elastic Net.** Elastic Net [50] combines L1 and L2 regularization to select features like lasso while retaining groups of correlated predictors like ridge. This helps prevent exclusion of potentially valuable redundant variables.
- **Automatic Relevance Determination (ARD).** ARD [51] places independent prior distributions on each weight to determine relevance. Features with heavier priors are more likely to have weights driven to zero.

### 2.1.3 Feature Selection vs. Feature Extraction

While both feature extraction and selection aim to improve model performance, this thesis focuses primarily on feature selection for several reasons. Both feature selection and feature extraction aim to enhance model performance by improving the quality of input data. Feature selection reduces dimensionality by selecting the most relevant features, maintaining their original meaning, while feature extraction transforms existing features into new representations that capture underlying patterns.

First, IoT network traffic data already contains rich, domain-specific features derived from network flows and protocol behaviors, making additional feature extraction potentially redundant or computationally expensive. In our case, extracting new features could introduce unnecessary complexity to the anomaly detection process.

Second, feature selection offers greater interpretability, which is crucial in security applications where understanding the basis for detection decisions is essential. By selecting only the most relevant features, the model's decisions become more transparent and understandable, which is important for practical security systems.

Third, our goal of optimizing computational efficiency aligns better with feature selection, as it

reduces dimensionality without the overhead of computing new features. Feature extraction requires the creation of additional features, which could increase the computational cost of training and inference in IoT security systems.

Fourth, in the context of deep learning models, which inherently perform their own feature extraction through hidden layers, explicit feature extraction could introduce unnecessary complexity. Deep learning models are capable of learning hierarchical representations from raw data, making explicit feature extraction methods less relevant and potentially detrimental to model simplicity and performance.

Thus, feature selection is a more practical and effective approach for our research objectives. However, it is important to note that Principal Component Analysis (PCA), although a feature extraction method, was still utilized in this study. We showed that PCA can yield good results in certain scenarios, particularly in reducing dimensionality without significant loss of information. Nonetheless, the majority of the methods we employed focused on feature selection, which aligns more closely with our goals of maintaining interpretability and optimizing computational efficiency.

### 2.1.4 Data Complexity

Data complexity refers to the intrinsic characteristics of datasets, such as class ambiguity, data sparsity, high dimensionality, and intricate decision boundaries, that pose challenges for machine learning algorithms beyond mere variations in class distribution [52]. These factors collectively influence the efficacy of anomaly detection methods and their ability to learn and generalize effectively [52]. Such attributes can challenge the learning process, rendering some datasets particularly difficult for machine learning models to interpret and learn from effectively. Several metrics have been chosen as tools for evaluating the subtle aspects of data complexity that affect these outcomes [53]. The following are key examples of data complexity metrics:

- **Entropy of Class Proportions (C1).** The C1 captures the imbalance in the dataset by computing the entropy of the class proportions [52]. It achieves lower values for balanced class distributions which are considered simpler problems.
- **Maximum Fisher's Discriminant Ratio (F1).** The F1 measures the overlap between feature values across classes, with higher values indicating more complex problems where no

individual feature can discriminate the classes [54]. It computes the ratio of inter-class to intra-class scatter for each feature [55].

- **Misclassification Complexity Measure (CM).** The CM provides insights into the complexity surrounding instances prone to misclassification, indicating areas where models may require refinement or tailored approaches to improve prediction accuracy [56].
- **Error Rate of Linear Classifier (L2).** The L2 computes the fraction of instances misclassified by a linear model like SVM, with higher values suggesting the data cannot be linearly separated [57]. It quantifies the complexity in terms of linear inseparability of the classes.

### 2.1.5 Data Imbalance Techniques

Data imbalance in IoT security presents unique challenges where malicious activities typically represent a small fraction of the overall network traffic [58]. The following techniques address these challenges, each suited to different scenarios and requirements:

- **Oversampling Techniques:** These techniques balance class distributions by generating synthetic minority class samples. Synthetic Minority Over-sampling Technique (SMOTE) [59] enhances intrusion detection by generating synthetic minority class instances through interpolation between real samples, effectively balancing class distribution. This method improves model generalization, reduces bias, and enhances detection accuracy for various cyber threats, including DDoS, malware, and anomaly-based attacks. By preserving the statistical properties of the dataset, SMOTE ensures a more representative and diverse training set, making machine learning models more robust in identifying security threats. Random Over-Sampling (ROS) [59] is another commonly used technique that generates synthetic examples by replicating minority class samples. ADASYN [60] adapts sample generation based on data density, making it effective for zero-day attack detection where anomalies are subtle. Borderline-SMOTE [61] refines this by focusing on decision boundaries, improving detection of complex attacks. Generative Adversarial Networks (GANs) [62] utilize adversarial training between two neural networks, a generator and a discriminator, to produce realistic synthetic minority samples, capturing complex non-linear patterns in the data. Variational Autoencoders

(VAEs) [63] use probabilistic encoding to learn latent data distributions, generating smooth, diverse synthetic samples that effectively represent minority classes.

- **Undersampling Techniques:** These techniques reduce the majority class to prevent model bias. Random Under-Sampling (RUS) [64] removes normal traffic instances, reducing computational costs while preserving detection accuracy. Tomek Links [65] eliminates overlapping majority samples, refining decision boundaries in intrusion detection. Near Miss [66] prioritizes majority samples close to minority instances, useful for analyzing IoT device behavior in cyber threat detection. SMOTE [59] can also be considered in undersampling approaches when a balanced sample size is required for model training. The Edited Nearest Neighbor (ENN) [67] algorithm identifies and removes noisy majority class instances that are misclassified by their nearest neighbors, clarifying class boundaries for improved model performance.

- **Hybrid Approaches:** These methods combine oversampling and undersampling for optimal balance. For example, SMOTETomek [68] applies SMOTE followed by Tomek Links removal, effective in noisy network environments.

- **Algorithm-Level Methods:** These techniques adjust model learning instead of altering data distribution. Cost-Sensitive Learning [69] assigns higher penalties for misclassifying attacks, reducing false negatives in security applications. One-Class Learning [70] models only normal behavior, treating deviations as anomalies, making it ideal for detecting zero-day threats. Ensemble Methods like EasyEnsemble [71] train multiple models on various subsets of data, improving detection performance on imbalanced datasets exhibiting complex attack patterns.

## 2.2 Related Work

This chapter presents a comprehensive review of the current state of research in anomaly detection for IoT network security, with a particular focus on the application of feature selection and data augmentation techniques. This thesis explores how these methodologies have been applied in both traditional machine learning and deep learning contexts, highlighting their potential to enhance

model performance. The chapter begins by examining anomaly detection approaches specific to IoT environments, then delves into feature selection methods in traditional and deep learning models, and concludes with an overview of data augmentation techniques. Through this exploration, this thesis aims to identify gaps in the existing literature and position our research within the broader context of the field.

### 2.2.1 Deep Learning-based Anomaly Detection

Recent advancements in deep learning have led to significant improvements in anomaly detection for IoT network security. The authors of [8] proposed Co-analyzed Malware Detection (CMD), an IoT malware detection and forensics system that integrates network and hardware data. CMD combines network and hardware data using neural networks to better detect IoT malware. This approach is notable for its holistic view of IoT devices, considering both network traffic and hardware-level data to improve detection accuracy. By integrating multiple data sources, CMD can potentially identify complex attack patterns that might be missed by traditional single-source detection methods.

Minh et al. [9] employ a Convolutional Neural Network (CNN)-based [72] interpretable ensemble system with anomaly detection to spot unknown network attacks on real datasets like CSE-CIC-IDS2018 [73], helping analysts make security decisions. Their work is particularly significant for its focus on interpretability, addressing the common criticism of deep learning models as "black boxes." This approach not only improves detection accuracy but also provides insights into the decision-making process, which is crucial for security analysts to understand and trust the system's outputs.

Wang et al. [2] focus on certifying the robustness of deep learning traffic analysis systems against attacks, indirectly emphasizing anomaly detection in network traffic. This research is crucial as it addresses the vulnerability of deep learning models to adversarial attacks, a critical concern in security applications. By developing methods to certify model robustness, this work contributes to the reliability and trustworthiness of deep learning-based anomaly detection systems in real-world deployments.

Several other studies shown in Table 2.1 utilize various anomaly detection models as the core

methodology for threat detection on real and simulated network datasets, like [1, 4, 10, 11, 74, 75]. These diverse approaches showcase the versatility of deep learning in addressing different aspects of IoT security, from real-time detection to handling complex, high-dimensional data.

Some of these studies employ feature selection [1, 10, 11] and data balancing [11] supplementary to anomaly detection models. For instance, Dong et al. [10] use feature selection to improve the efficiency of their real-time IoT malicious traffic detection framework. This approach not only enhances detection accuracy but also reduces computational overhead, which is crucial for real-time applications in resource-constrained IoT environments. Fu et al. [11] incorporate both feature selection and data balancing to enhance the robustness of their malicious traffic detection system, addressing the common challenge of imbalanced datasets in cybersecurity applications. A Additionally, Mirsky et al. [76] proposed Kitsune, which uses autoencoders [77] for finding network intrusions in real-time, showcasing the applicability of anomaly detection in live scenarios. This work demonstrates the potential of unsupervised learning techniques in identifying novel attack patterns without relying on predefined signatures, a critical capability in the rapidly evolving landscape of IoT security threats.

These diverse approaches highlight the growing importance and complexity of anomaly detection in IoT network security. They also underscore the need for more comprehensive studies that evaluate the effectiveness of various techniques across different models and datasets, considering factors such as computational efficiency, scalability, and adaptability to new types of attacks.

### 2.2.2 Data Augmentation

Data augmentation techniques have been widely adopted across various domains to enhance model performance and address data scarcity issues. In image classification, works like [78] and [7] have demonstrated significant improvements in model accuracy through augmentation techniques such as rotation, flipping, and color jittering. These methods effectively increase the diversity of training data, helping models learn more robust and generalizable features.

Text classification has significantly benefited from data augmentation techniques. For instance, studies such as [79], demonstrate how methods like synonym replacement, random insertion, random swapping, and random deletion enhance model robustness. These approaches are especially

valuable in natural language processing (NLP) tasks [80], where acquiring large and diverse datasets often poses a challenge.

In the context of network security and intrusion detection, data augmentation has been applied to address class imbalance and improve detection rates. Wang et al. [81] explored the combination of data augmentation methods with gated convolution models for building effective and robust intrusion detection systems. Their work demonstrated how augmentation could help models better learn the characteristics of rare attack types, a common challenge in cybersecurity where malicious activities are often underrepresented in datasets.

Researchers have explored various innovative data augmentation techniques specifically in the field of intrusion detection. Lim et al. [82] introduced a generative data augmentation technique called "doping" using Generative Adversarial Networks (GANs) [83] to generate synthetic samples for anomaly detection. This approach leverages the power of adversarial learning [84] to create realistic, diverse examples of anomalous behavior, potentially improving the model's ability to detect novel attacks.

Yuan et al. [85] proposed a data augmentation-based intrusion detection method for smart home security, demonstrating improvements in classification accuracy. They converted network traffic data into images and used an Auxiliary Classifier Generative Adversarial Network (AC-GAN) [86] to generate synthetic samples, effectively addressing the issue of imbalanced data. This innovative approach of transforming network data into visual representations opens up new possibilities for applying image-based augmentation techniques to network security problems.

However, these works often do not critically examine the universal applicability of data augmentation or explore scenarios where it might introduce noise or be unnecessary. This gap is particularly relevant in IoT security, where the diversity of devices and attack vectors makes the effectiveness of augmentation techniques less predictable. The effectiveness of data augmentation techniques can differ based on the specific task, model, and dataset. This variability underscores the importance of meticulously choosing which data to augment and how to implement augmentation strategies in a way that is most beneficial for the specific analytical needs of IoT anomaly detection. It also highlights the need for research that systematically evaluates the impact of different augmentation techniques across various IoT security scenarios, considering factors such as data complexity, attack

diversity, and model architecture.

| Proposal (Year) | Scope | Anomaly Detection Models | | | | | | | | | Dataset | | | Real-time | Data Augmentation | Feature Selection | Data Balancing |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AE | NN | RF | BERT | GB | RNN | CNN | Kmeans Clustering | Statistical Clustering | Simulated | Real | IoT Dataset | | | | |
| Hu et al. (2024) [74] | Network Traffic | | | | | • | | | | | | • | | • | | | |
| Dong et al. (2023) [10] | Network Traffic | | | • | | | | | | | | • | • | • | | • | |
| Wang et al. (2023) [2] | Network Traffic | • | | | | | | | | | • | • | | | | | |
| Yuan et al. (2023) [75] | Network Traffic | • | | | | | | | • | | | • | • | | | | |
| Wei et al. (2023) [4] | Network Traffic | • | • | | | | • | | | | | • | | | | • | |
| Araya et al. (2023) [87] | Smart Home | • | • | • | | | | | | | • | • | • | | | | |
| Jayaraman et al. (2023) [88] | Network Traffic | | • | • | | | | • | | | | • | • | | | • | • |
| Vitorino et al. (2022) [89] | Network Traffic | | | • | | | | | | | | • | • | | • | | • |
| Jeelani et al. (2022) [90] | Smart Home | | • | | | | | • | | | | • | • | | | | |
| Austin et al. (2021) [91] | Network Traffic | | | • | | | | | | | | • | • | | | • | • |
| Fu et al. (2021) [11] | Network Traffic | | | | | | | | | • | | • | | • | | • | • |
| Nanni et al., (2021) [78] | Detect Malware | | | | | | | • | | | | • | | | • | | |
| Catak et al., (2021) [92] | Detect Malware | | | | | | • | • | | | | • | • | | • | | • |
| Tang et al. (2020) [1] | Web Traffic | • | | | | | • | | | | | • | | • | | • | |
| Yuan et al., (2020) [85] | Intrusion Detection | | | | | | | • | | | • | | • | | • | | • |
| Al Olaimat et al., (2020) [6] | Network Traffic | | • | • | | | | | | | • | | | | • | | • |
| Perez et al., (2017) [93] | Image Classification | | • | • | | | | | | | | • | | | • | | |
| AMETIS [18]★ | Network Traffic | • | • | • | • | • | | • | | | | • | • | | • | • | • |

Table 2.1: Comparison of related works for anomaly detection.

### 2.2.3 Comparative Study

Table 2.1 summarizes the findings of the comparative study on supplementary techniques across different model configurations. While several of these approaches focus on IoT-based networks—using datasets such as CICIoT2023[14] and IoT-23[15]—not all of them are strictly IoT-oriented, as some works address general anomaly detection in other domains. This evaluation provides insights into the optimal combination of feature selection and data augmentation with anomaly detection models, and distinguishes this thesis from existing works in the field of IoT security.

Several notable works in IoT security incorporate feature selection and data augmentation techniques to enhance their anomaly detection systems. For instance, Dong et al. [10] and Fu et al. [11] employ feature selection to improve the efficiency and robustness of their detection frameworks. Similarly, studies like [74] and [4] explore various deep learning models for anomaly detection, implicitly relying on the models' ability to learn relevant features. While these approaches shows

promising results, this thesis takes a step further by critically evaluating the effectiveness of these techniques across a wide range of scenarios.

Unlike previous studies that incorporate feature selection or data augmentation as part of their methodology, this thesis systematically evaluates the impact of these techniques on model performance. It examines how different feature selection methods interact with various model architectures, offering insights into which combinations yield optimal results for specific types of IoT network data. Furthermore, this thesis adopts a comprehensive approach to data augmentation within the context of IoT security, analyzing the conditions under which it should be applied in anomaly detection scenarios. By evaluating a diverse set of models and utilizing multiple datasets, this thesis provides insights that are potentially more generalizable across various IoT network environments, contributing to the development of more effective and efficient security solutions.

### 2.2.4 Comparison with Existing IoT Anomaly Detection Approaches

Recent studies extensively explore anomaly detection in IoT networks, leveraging various machine learning and deep learning models. Hu et al. [74] proposes an FPGA-based frequency transformation combined with machine learning for detecting malicious network traffic in IoT environments. Their approach enhances real-time detection capabilities by reducing computational overhead. However, their work primarily focuses on frequency domain transformation, whereas our research evaluates the impact of feature selection and data augmentation, offering a broader framework for optimizing anomaly detection.

Wei et al. [4] introduced XNIDS, an explainable deep learning-based intrusion detection system. Their approach emphasizes interpretability by providing insights into neural network decisions. While explainability is crucial, their work does not explore the impact of feature selection and data augmentation on deep learning models. Our study fills this gap by systematically analyzing how feature selection techniques improve detection accuracy while reducing computational complexity.

Yuan et al. [75] investigates boundary augmentation to improve malicious traffic detection in IoT networks. Their study demonstrates that augmenting data can improve classifier performance. However, they do not examine cases where data augmentation might introduce noise or degrade performance. In contrast, our research systematically evaluates data complexity metrics to determine

when data augmentation is beneficial, ensuring its application does not lead to overfitting.

Unlike prior work, our research provides a holistic evaluation of anomaly detection in IoT by considering feature selection strategies and their impact on deep learning and machine learning models. Additionally, we assess data augmentation effectiveness rather than applying it indiscriminately. Our study introduces a structured framework that allows users to systematically configure and optimize their anomaly detection pipelines. These contributions differentiate our work by providing a structured and data-driven methodology for optimizing IoT anomaly detection systems.

### 2.2.5 Comparison with Existing Works on the IoT-23 Dataset

The IoT-23 dataset [15] was chosen for its comprehensive representation of IoT network traffic, featuring detailed flow-level features, a diverse range of attack types (e.g., botnets, DDoS, MITM, reconnaissance), and inherent class imbalance. These characteristics not only make IoT-23 a realistic benchmark for evaluating anomaly detection methods but also facilitate in-depth feature engineering and the assessment of data augmentation and balancing strategies. Its widespread adoption in recent studies further validates its suitability as a benchmark for IoT security research.

Several studies leverages the IoT-23 dataset to evaluate machine learning and deep learning approaches for IoT anomaly detection. For example, Jeelani et al. [90] evaluates multiple algorithms and reported that conventional methods such as Naive Bayes and SVM achieved relatively low accuracies (around 30% and 69%, respectively), while a Decision Tree classifier obtained the highest accuracy of 73% with minimal computational cost (approximately 3 seconds). Similarly, in another work [88], four classifiers are comparing on key performance metrics. Their results indicate that Random Forest achieved an F1-Score of 0.9936, closely followed by Decision Trees (F1-Score of 0.9894), whereas SVM lagged significantly (F1-Score of 0.7888). In addition, related studies by Austin et al. [91] reports near-perfect performance—F1 scores of 100% and 97.3% (with 92.35% for Linear SVM), respectively—when employing advanced feature selection and ensemble methods on subsets of IoT-23.

In comparison, the experimental results present in this thesis demonstrate highly competitive performance. For instance, our BERT model integrates with the *trank* feature selection method (scenario S8, mentioned in subsection 5.3.4) achieves an F1-Score of 99.7%, while models such as

CNN, NN, and AE consistently reports F1-Scores above 99% across different dataset subsets. These outcomes not only compare favorably with the high-performance metrics reports in [88] but also outperform traditional approaches highlights in [90] in terms of detection accuracy and efficiency.

Although variations in experimental settings and dataset configurations (e.g., the use of different IoT-23 subsets or additional pre-processing steps) may lead to differences in absolute performance numbers, the consistency of high performance across these works underscores the robustness of the IoT-23 dataset as a benchmark. Our results further confirm that a well-designed pipeline—incorporating targeted feature selection, strategic data augmentation, and hyperparameter optimization—can achieve state-of-the-art performance in IoT anomaly detection. It is important to note that the studies cited here represent only a few examples among the many works that have successfully employed IoT-23, such as those by Alharbi et al.[94], Htwe et al.[95], Iturbe-Araya et al.[96], Sun et al.[97], Araya et al. [87], and Vitorino et al.[89]. Numerous other studies have also leveraged this dataset to validate their approaches, further demonstrating its versatility and reliability as a benchmark for IoT security research.

## 2.3 Conclusion

This chapter has provided a comprehensive examination of current research in IoT anomaly detection, identifying crucial gaps in existing approaches and positioning our contributions within the broader context of the field.

The identified gaps in existing research clearly demonstrate the need for more adaptive and efficient approaches to IoT anomaly detection. Our proposed solutions, particularly in the areas of dynamic feature selection and context-aware data augmentation, address these limitations while advancing the state of the art in the field.

The following chapters will detail our methodology and demonstrate how our approach concretely addresses these challenges.

# Chapter 3

# Feature Selection Optimization

This chapter presents a comprehensive approach for optimizing feature selection within the anomaly detection pipeline for Internet of Things (IoT) network traffic logs. Our goal is to identify the most effective combinations of preprocessing techniques, feature selection strategies, hyperparameter tuning methods, and machine learning or deep learning models. By systematically comparing a variety of configurations, this thesis aims to determine which sequences of pipeline steps yield improved efficiency and accuracy in anomaly detection tasks.

## 3.1 Approach Overview

Feature selection plays a critical but often underappreciated role in building accurate and efficient anomaly detection models, particularly in complex IoT environments. Traditional practices may underestimate its value, especially when dealing with deep learning models, which are sometimes thought capable of learning optimal representations directly from raw data. In this study, this thesis challenges that assumption by explicitly incorporating different feature selection methods and comparing how they interact with data preparation, data balancing, and hyperparameter tuning.

Our method involves assembling multiple scenarios that vary the order and combination of key steps:

- *Data Preparation (D)*
- *Feature Selection (F)*

- *Data Balancing (B)*
- *Anomaly Detection (A)*

This thesis begins with a fixed starting point—data preparation—and then construct ten scenarios (denoted $S0$–$S9$) by altering the presence and order of the other components. This systematic approach allows us to isolate the individual contributions of each step and to understand the interplay among them.

Figure 3.1 illustrates these pipeline configurations. Each scenario is evaluated on multiple IoT datasets and applied to several machine learning and deep learning models. This comprehensive experimental design ensures that our conclusions are robust and generalizable across different data sources and modeling techniques. Ultimately, by comparing the results from all scenarios, this thesis aims to provide a principled guide for optimizing anomaly detection pipelines in IoT contexts, improving both performance metrics and computational efficiency.

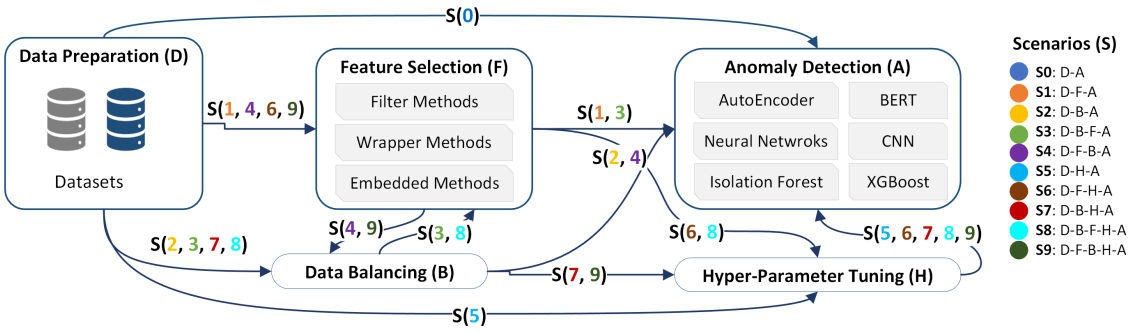


Figure 3.1: Orderings of anomaly detection pipeline modules across scenarios ($S0$–$S9$).

By systematically evaluating these diverse scenarios, this thesis aims to provide a comprehensive understanding of how each step in the anomaly detection pipeline contributes to overall performance. This approach allows for the identification of optimal configurations tailored to specific datasets and anomaly detection requirements, potentially leading to more robust and efficient anomaly detection systems in various applications.

### 3.1.1 Data Preparation

This thesis utilizes two popular IoT datasets: (i) IoT Aposement 23 (IoT-23) [15] and (ii) CICIoT2023 [14].

- **IoT-23 Dataset.** The IoT-23 dataset [15] captures network traffic from various IoT devices, including smart locks, Amazon Echo, and Philips HUE lamps. It consists of over 760 million packets and 325 million labeled flows, offering a comprehensive range of IoT-related activities. The dataset includes 20 malware captures and three benign traffic captures, providing a representative balance of malicious and normal activities. The malware captures encompass attacks from well-known malware families, such as *Mirai*, *Torii*, *Trojan*, *Gagfyt*, *Kenjiro*, and *Okiru*, making it particularly valuable for studying various IoT attack vectors. This dataset was generated in the controlled environment of the Stratosphere Laboratory, part of the Avast AIC laboratory, ensuring realistic and accurate network conditions. Detailed features included in the dataset are source and destination IPs, packet sizes, protocol distributions, and application-layer protocol predictions, allowing comprehensive analysis and robust testing of anomaly detection methods.

- **CICIoT2023 Dataset.** The CICIoT2023 dataset [14] simulates network traffic involving 33 distinct attack across a network of 105 IoT devices, such as IP cameras, smart thermostats, and home assistants. These attacks are categorized into seven major types, including *Distributed Denial of Service (DDoS)*, *Denial of Service (DoS)*, *Reconnaissance (Recon)*, *Web-based Attacks*, *Spoofing*, and *Mirai*. The dataset was generated in a controlled lab environment, ensuring the accuracy of labels while maintaining realistic conditions for IoT network operations.

  This dataset is particularly valuable for its scale and diversity, with millions of records covering both attack and benign traffic patterns. The inclusion of a wide range of attack types and device behaviors provides a comprehensive testbed for evaluating anomaly detection systems. Furthermore, its realistic attack scenarios and detailed traffic features, such as source and destination IPs, packet sizes, and protocol distributions, enable robust testing of feature selection, data augmentation, and detection methodologies.

Both datasets undergo extensive preprocessing to prepare them for use in anomaly detection models. This includes handling missing values by replacing or removing incomplete entries to maintain dataset integrity. Categorical variables, such as protocol types and connection states, are transformed into numeric formats using techniques like one-hot encoding, ensuring compatibility with machine learning models. Additionally, numerical features are standardized to a common scale using methods such as min-max normalization, ensuring consistency across features and mitigating biases caused by varying feature ranges.

These preprocessing steps are essential for maintaining data quality and consistency, enabling fair and reliable comparisons across different models and scenarios. By ensuring the datasets are clean and properly formatted, this thesis lays a strong foundation for evaluating anomaly detection techniques in diverse IoT network environments.

### 3.1.2 Feature Selection

This thesis employs various feature selection methods, including $\chi^2$, L1, L2, MI, PCA, PSO [98], RF, SKB, and Trank, as discussed in Section 2.1.2. For *filter methods*, feature importance is ranked independently, and features with scores higher than the average are retained for both ML and DL models, effectively removing noisy and less relevant candidates. This approach provides an efficient initial screening of features based on their intrinsic properties, without requiring model training.

The *wrapper methods* return a subset of features by evaluating feature subsets using the machine learning algorithm intended for classification. These methods have the potential to capture feature interactions that filter methods might overlook. However, they can be computationally intensive, particularly when dealing with large feature sets.

The *embedded methods* are integrated into the DL models, performing feature selection as part of the model training process. These methods strike a balance between the computational efficiency of filter methods and the model-specific optimization provided by wrapper methods.

In addition to employing individual methods, this thesis proposes merging the results of multiple feature selection techniques, with a focus on wrapper and filter methods, as embedded methods do not generate an explicit list of selected features. This approach aims to leverage the complementary

strengths of different selection methods, potentially resulting in more robust and comprehensive feature sets. The proposed combined feature selection strategies include:

- **All Selected Features.** It merges all features identified by any selection method, ensuring no potentially significant feature is overlooked. This approach maximizes feature retention but may include some less relevant features, requiring careful consideration of potential underfitting risks.
- **Common Features.** It selects those features chosen by all methods, identifying the core set of important features. This conservative approach ensures only the most consistently important features are retained, potentially reducing noise but risking the exclusion of relevant features identified by only some methods.
- **Majority Voting.** It leverages collective decisions, selecting features chosen by at least half of the employed methods. This balanced approach aims to find a middle ground between inclusivity and selectivity, potentially capturing a broader range of relevant features while still filtering out less important ones.
- **Separate Wrapper and Filter Common Features.** It combines commonly selected features from wrapper and filter methods into two distinct subsets. This approach allows for the comparison of features deemed important by different selection paradigms, potentially providing insights into the strengths and biases of each method type.
- **Wrapper and Filter Majority Voting.** It applies majority voting separately to wrapper and filter methods, then combines selected subsets. This method allows for method-specific consensus before combining results, potentially preserving the unique insights of each method type while still achieving a level of agreement within each category.

These approaches are designed to harness the strengths of various feature selection techniques while addressing their individual biases and limitations. Combining methods enables the creation of more robust feature sets, capable of capturing a diverse range of relevant information for anomaly detection tasks. This has the potential to enhance model performance and improve generalizability across different datasets and attack types in IoT environments.

### 3.1.3 Anomaly Detection

Various deep learning and machine learning models enable robust anomaly identification. This study uses popular models like AutoEncoder (AE) [99], BERT [100], Isolation Forest (IF) [101], Neural Network (NN) [99], Convolutional Neural Networks (CNN) [72], and XGBoost [102], each suited for specific data or anomaly detection tasks. AutoEncoders excel in unsupervised anomaly detection by learning to reconstruct normal patterns, while BERT captures complex temporal patterns in sequence data. Isolation Forest efficiently detects outliers in high-dimensional spaces, and Neural Networks, including CNNs, offer flexibility in learning non-linear relationships and spatial-temporal patterns in network traffic. XGBoost provides a powerful ensemble method, adept at handling imbalanced datasets and capturing complex feature interactions.

Feature selection techniques are applied to enhance performance by retaining relevant inputs, potentially improving accuracy and efficiency. This crucial step helps reduce noise, mitigate the curse of dimensionality, and improve model interpretability. By focusing on the most informative features, these techniques can reduce computational requirements and potentially improve the models' generalization capabilities across different IoT network environments and attack types. The combination of advanced machine learning models and effective feature selection aims to create a robust framework for detecting anomalies in the complex and dynamic landscape of IoT network traffic.

### 3.1.4 HyperParameter Tuning

This thesis uses a greedy approach for hyperparameter tuning in ML and DL models. It employs *KerasTuner* [1] with its Hyperband algorithm [103] to efficiently navigate complex hyperparameter spaces and identify optimal configurations for the models. This method efficiently explores hyperparameter configurations by evaluating many candidates briefly with small epochs and extending training for promising ones. It uses decision trees to optimize selection, focusing computational resources on configurations that improve validation metrics for anomaly detection.

The Hyperband algorithm is particularly well-suited for this task as it combines random search

[1] https://github.com/keras-team/keras-tuner

with an early-stopping mechanism. This approach allows for a more thorough exploration of the hyperparameter space compared to traditional grid search methods, while also being more computationally efficient. By adaptively allocating resources to promising configurations, Hyperband can quickly identify high-performing hyperparameter sets, even in scenarios with limited computational resources. This is especially valuable in the context of anomaly detection in IoT networks, where model performance can be highly sensitive to hyperparameter choices, and the large volume of data makes exhaustive search methods impractical. The use of this advanced tuning strategy aims to enhance the overall performance and generalizability of the anomaly detection models across various IoT network scenarios and attack types.

### 3.1.5 Data Balancing

To address class imbalance in anomaly detection datasets, this thesis employs the Synthetic Minority Oversampling Technique (SMOTE) [104] for up-sampling minority classes. This method generates synthetic data to enhance diversity and improve model generalization, as recommended in imbalanced learning for anomaly detection [3].

SMOTE works by creating synthetic examples in the feature space, rather than simply duplicating existing minority class samples. It operates by selecting a minority class instance and finding its k-nearest neighbors. New synthetic instances are then created by interpolating between the selected instance and its neighbors. This approach helps to increase the representation of minority classes without simply replicating existing data points, which can lead to underfitting. In the context of IoT network anomaly detection, where attack instances are often far less frequent than normal traffic, SMOTE can help create a more balanced dataset. This balanced representation allows machine learning models to learn more effectively from both normal and anomalous patterns, potentially improving their ability to detect rare but critical security events in IoT networks.

### 3.1.6 Methodology Scenarios

This section presents scenarios used to study the impact of feature selection and data balancing on anomaly detection performance using machine learning and deep learning models. Figure 3.1 shows evaluation of ten scenarios to find the optimal ordering of key components in the anomaly

detection pipeline.

(1) **Data Flow Baselines (Scenarios S0-S2)** These scenarios appraise the inherent capabilities of the models (S0) prior to incorporating feature selection (S1) or class balancing (S2), evaluated independently to discern their individual contributions. These baseline scenarios provide a foundation for understanding the performance of raw models and the isolated effects of feature selection and class balancing.

(2) **Feature Selection vs. Balancing Order (Scenarios S3-S4)** These scenarios inspect the efficacy of applying data balancing techniques either before (S3) or after (S4) feature selection, to find the most effective procedural order. This comparison is crucial as the order of these operations can significantly impact the final feature set and the model's ability to learn from balanced data.

(3) **Hyperparameter Tuning Integration (Scenarios S5-S7)** These scenarios investigates optimal tuning placement within the pipeline: with only anomaly detection (S5), after feature selection (S6), or following data balancing (S7). These scenarios explore how the timing of hyperparameter optimization affects model performance, considering the interplay between tuned parameters and the characteristics of the processed data.

(4) **End-to-End Integration (Scenarios S8-S9)** These scenarios constructs and evaluates comprehensive pipelines integrating all components in a sequential order, specifically, data balancing followed by feature selection and then tuning (S8) versus feature selection succeeded by data balancing and tuning (S9), to examine their holistic impact. These scenarios represent full-fledged anomaly detection pipelines, allowing for the assessment of how different orderings of all components affect overall system performance.

Examining these scenarios facilitates independent and comparative analyses of key factors affecting anomaly detection efficacy in IoT environments. This includes the isolated effects of feature selection, data balancing, and hyperparameter tuning, as well as their interactions and integration points. This thesis investigates how these techniques can optimize anomaly detection systems in cybersecurity contexts. The investigation assesses full end-to-end pipeline ordering to establish best

practices for configuring high-performance anomaly detection systems tailored to IoT frameworks, aiming to improve both efficacy and efficiency in identifying anomalies while maintaining accurate normal data characterization. To understand the trade-offs between computational costs and accuracy benefits, runtime metrics are recorded and analyzed across all scenarios. It is important to note that all model architectures and all feature selection methods are applied across all scenarios (S0-S9). This comprehensive approach allows for a thorough evaluation of each combination's effectiveness in various configurations, providing a robust framework for identifying optimal strategies in IoT anomaly detection.

## 3.2 Conclusion

This chapter systematically outlines our approach to optimizing feature selection within IoT anomaly detection pipelines. By evaluating multiple scenarios across datasets and models, we clarify the individual and combined effects of feature selection, data balancing, and hyperparameter tuning. The comprehensive analysis presented here serves as a practical guide for constructing effective anomaly detection systems, ultimately contributing to improved IoT security practices.

# Chapter 4

# Data Augmentation and Complexity Analysis

This chapter explores the role of data augmentation in enhancing anomaly detection systems, particularly in the context of IoT networks. Data augmentation is a technique used to expand and diversify datasets, with the aim of improving model performance. However, its effectiveness is closely tied to the characteristics of the dataset being augmented. This chapter focuses on analyzing the complexity of augmented datasets and investigating their impact on model performance.

## 4.1 Approach Overview

This thesis aims to identify datasets that can enhance anomaly detection performance when combined with the original data. A step-by-step process, as illustrated in Figure 4.1, is followed. Initially, datasets are combined, and the best settings from previous experiments are applied. Subsequently, the complexity of the combined data is calculated, and the selected ML or DL model is trained on this data. By analyzing the relationship between data complexity and model performance, this thesis provides recommendations for incorporating new data into the current dataset.

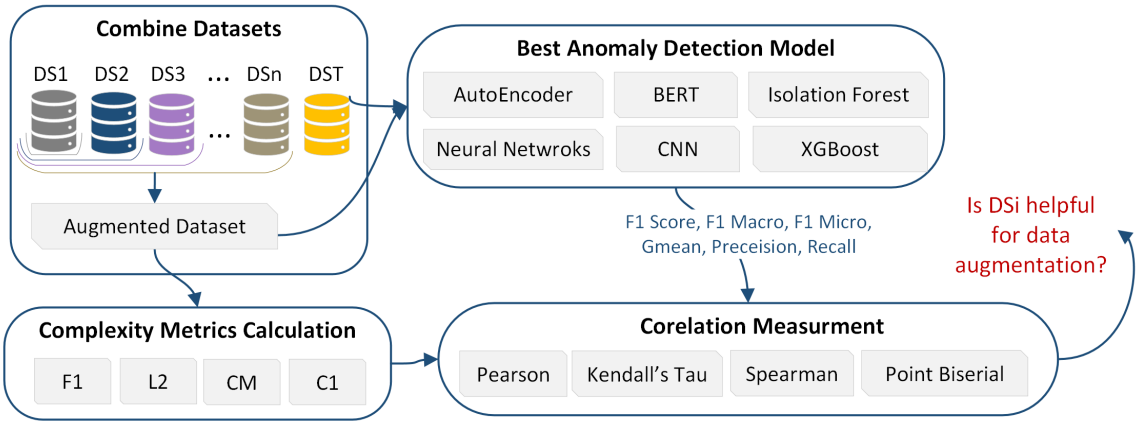


Figure 4.1: Process flow of the data augmentation.

### 4.1.1 Combine Datasets and Complexity Metrics Calculation

This thesis begins the experiment by combining various datasets into a single dataset, with the objective of creating a comprehensive pool of data that encapsulates diverse characteristics and patterns. The combined dataset undergoes an assessment using data complexity approaches as described in Section 2.1.4, such as Misclassification Complexity Measure (CM), Entropy of Class Proportions (C1), Maximum Fisher's Discriminant Ratio (F1), and Error Rate of Linear Classifier (L2). Following the evaluation of data complexity, the selected ML or DL model from the experiments is trained to identify the best-performing model and scenario.

### 4.1.2 Correlation Measurement

Determining the correlation between the performance indicators of the anomaly detection model and the complexity of the data is a critical phase in the process. To establish this link, statistical correlation approaches such as Pearson [105], Kendall's Tau [106], Spearman's Rank [107], Point Biserial [108], and (MIC) [109]methods are employed, which are explained as follows:

- **Pearson Correlation.** Measures the linear relationship between two continuous variables by evaluating how changes in one variable are associated with changes in another. It assumes a linear relationship and is sensitive to outliers [105].

- **Kendall's Tau.** A non-parametric method that assesses the strength and direction of the

relationship between two variables by comparing the concordant and discordant pairs of data points. It is particularly effective for ordinal data [106].

- **Spearman's Rank Correlation.** A non-parametric technique that evaluates the monotonic relationship between two variables by ranking the data. It is robust to outliers and useful when variables do not meet the assumptions of linearity [107].

- **Point Biserial Correlation.** A method that assesses the relationship between a binary variable (e.g., categorical data) and a continuous variable. It is commonly used when analyzing datasets with mixed data types [108].

- **Maximal Information Coefficient (MIC).** A versatile method that identifies both linear and non-linear associations between variables. It is particularly effective for detecting complex and non-linear relationships that other methods may miss [109].

Each of the selected methods is tailored to capture specific types of relationships, providing a comprehensive understanding of how data complexity correlates with model performance in IoT anomaly detection scenarios. This multi-faceted analysis ensures a robust interpretation of the data, enabling deeper insights into the factors that influence detection outcomes.

Several performance metrics are used in this analysis, such as *g-mean*, *F1-Score*, *F1-macro*, *F1-micro*, *recall*, and *precision*. These metrics are specifically chosen to evaluate the effectiveness of anomaly detection systems, particularly for imbalanced datasets, where a balanced performance across classes is helpful. The details of how these metrics are computed and their relevance to IoT anomaly detection are presented in Section 5.1.2.

In conclusion, the developed methodology offers a structured framework for tailoring anomaly detection systems to the specific requirements of IoT network traffic logs. By incorporating comprehensive correlation analyses between data complexity and diverse performance metrics, this approach enables the development of more robust and adaptable anomaly detection systems. Understanding these relationships allows practitioners to make informed decisions regarding data preprocessing, feature selection, and model configuration, ultimately enhancing the effectiveness of security measures in IoT environments.

## 4.2 Conclusion

This chapter has outlined the role of data augmentation in anomaly detection and introduced the methodological framework for assessing its impact. By combining datasets, analyzing complexity, and exploring the relationship between complexity metrics and performance outcomes, AMETIS [18] provides the foundation for evaluating the effectiveness of augmentation strategies. The insights gained from this analysis will be presented in the evaluation chapter.

# Chapter 5

# Experimental Result

In this chapter, a detailed evaluation of the proposed solution is presented. The effects of data augmentation, feature selection, data imbalance, and hyperparameter tuning on various evaluation metrics are analyzed.

## 5.1 Experimental Setup

To ensure robustness and reliability of our results, we implement a rigorous validation process. All experiments were run three times, and the average results were used in our analysis.

### 5.1.1 Hardware and Software

Our experimental setup was orchestrated on a server running Linux version 7.9 (Nitrogen) with kernel version 3.10.0-1160.95.1.el7.x86_64, powered by an AMD Opteron (tm) Processor 6180 SE.

This thesis integrated a comprehensive suite of specialized libraries to tackle the complexity of the experiments and ensure thoroughness in the analyses. This ensemble included:

- *imbalanced-learn (v0.10.1)*[110] to address class imbalance issues,
- *keras (v2.13.1)*[111] alongside *tensorflow (v2.13.0)*[112] for the development of deep learning architectures,
- *pandas*[113] for data manipulation and preprocessing,

- *numpy*[114] for numerical computations and array processing,
- *scikit-learn (v1.2.0)*[115] for machine learning algorithms and preprocessing utilities,
- *matplotlib*[116] for creating visualizations,
- *BorutaPy*[117] for feature selection using the Boruta method,
- *mlxtend*[118] for sequential forward and backward feature selection methods,
- *ReliefF*[119] for feature scoring based on instance-based learning,
- *XGBoost*[102] for feature selection and classification using gradient boosting,
- *SMOTE (Synthetic Minority Over-sampling Technique)*[104] for handling class imbalance by generating synthetic samples.

By integrating these libraries and methods, this thesis enabled robust feature selection, data augmentation, model optimization, and evaluation, ensuring a thorough and accurate analysis of the proposed anomaly detection systems.

### 5.1.2 Evaluation Metrics

This thesis employs multiple evaluation metrics to provide a comprehensive view of model effectiveness, specifically for imbalanced data. The metrics used are as follows:

#### Precision

Precision measures the proportion of true positives among predicted positives, providing insight into the accuracy of positive predictions [120]:

$$Precision = \frac{TP}{TP + FP}$$

#### Recall

Recall measures the proportion of actual positives correctly identified by the model, highlighting its ability to detect all positive instances:

$$\textit{Recall} = \frac{TP}{TP + FN}$$

**F1-Score**

The F1-Score provides a harmonic mean of *precision* and *recall*, capturing the balance between avoiding false positives and detecting true positives [121]:

$$F1\text{-Score} = 2 \cdot \frac{\textit{Precision} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$

**F1-Macro**

The *F1-Macro* metric represents the unweighted average of the *F1-Scores* across all classes, giving equal importance to each class, regardless of its frequency:

$$F1\text{-macro} = \frac{1}{C} \sum_{i=1}^{C} F1_i$$

**F1-Micro**

The *F1-Micro* aggregates contributions from all classes to compute the global *F1-Score*, offering a holistic measure of model performance [122]:

$$F1\text{-micro} = \frac{2 \cdot \sum_{i=1}^{C} TP_i}{2 \cdot \sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FP_i + \sum_{i=1}^{C} FN_i}$$

**G-Mean**

The *G-Mean* assesses the balance between the model's performance on positive and negative classes, making it crucial for imbalanced datasets [123]:

$$\textit{g-mean} = \sqrt{\textit{TPR} \cdot \textit{TNR}}$$

Where:

$$\textit{TPR} = \frac{TP}{TP + FN}, \quad \textit{TNR} = \frac{TN}{TN + FP}.$$

#### Importance of Metrics

These metrics are particularly important in the context of anomaly detection, where false positives can trigger unnecessary alerts, and false negatives can allow critical security breaches to go undetected. By utilizing this diverse set of metrics, this thesis provides a nuanced understanding of model performance, going beyond simple accuracy measures. This approach enables a thorough evaluation of the proposed anomaly detection system's effectiveness, particularly in addressing the challenges posed by class imbalance and detecting rare but critical events.

## 5.2 Dataset

For the experimental evaluation of the proposed anomaly detection framework, two widely recognized IoT network datasets, *IoT-23* [15] and *CICIoT2023* [14], are utilized. To ensure a structured and consistent approach, selected dataset subsets are assigned standardized identifiers ($DS_1 - DS_5$), allowing for clear reference throughout the results analysis.

The IoT-23 dataset was selected due to its comprehensive coverage of various IoT malware families and benign traffic, making it highly relevant for evaluating anomaly detection models. Given its detailed network flow-level data, it enables fine-grained feature engineering and facilitates the application of feature selection and anomaly detection techniques. The dataset also presents significant class imbalance, making it suitable for testing data augmentation and balancing strategies.

#### IoT-23 Dataset Subsets

The IoT-23 dataset [15], developed by the Stratosphere Laboratory, is a comprehensive benchmark dataset for IoT security research. It contains labeled network traffic flows collected from a wide range of IoT devices, including smart cameras, intelligent lighting systems, smart speakers, home automation hubs, and more. The dataset includes 20 malware captures executed on different IoT devices, as well as 3 benign captures of real IoT device traffic, such as from a Philips HUE smart LED lamp, an Amazon Echo personal assistant, and a Somfy smart door lock. This provides a detailed representation of both benign and malicious network activities, offering valuable insights for anomaly detection model evaluation.

The dataset comprises approximately 325 million labeled network flows, covering a wide range of IoT-related activities. The total dataset size exceeds 20 million packets, captured across 23 different scenarios. Each scenario represents a specific combination of IoT devices and cyberattacks, ensuring a diverse evaluation framework. The network traffic is recorded in packet capture (PCAP) format, accompanied by extracted flow-based features in CSV format, facilitating its integration into machine learning pipelines.

IoT-23 encompasses a broad spectrum of attack types, including botnet activity, distributed denial-of-service (DDoS) attacks, and malware propagation. The dataset contains traces of Mirai, Torii, and Okiru botnet infections, which target IoT vulnerabilities to compromise devices and establish large-scale attack infrastructures. Additionally, it includes denial-of-service attacks designed to exhaust device resources and render them inoperable. The dataset also captures man-in-the-middle (MITM) attacks, where adversaries intercept communications between IoT devices to manipulate or eavesdrop on data transmission. Other attack vectors include reconnaissance activities, where adversaries probe IoT networks for exploitable weaknesses, and data exfiltration attempts, which focus on unauthorized access and information theft.

The dataset is structured with detailed features, including packet size distributions, TCP flag counts, entropy measures, and inter-arrival times. It also includes flow-level metadata, such as connection durations and byte-per-second rates. Given the dataset's diverse attack vectors and device types, specific subsets are selected to provide representative test scenarios.

The selection of subsets from the IoT-23 dataset was based on the need to ensure diversity in attack types, traffic distributions, and dataset sizes for a comprehensive evaluation of the anomaly detection framework. $DS_1$ *(Subset 8-1)* was chosen for its mix of benign and attack traffic, making it a representative sample of real-world IoT environments. $DS_2$ *(Subset 20-1)* was selected due to its higher proportion of benign traffic (21%), providing a more balanced scenario for assessing detection performance. $DS_3$ *(Subset 3-1)* was included primarily for its larger size, ensuring that the model is tested on a dataset with a greater volume of traffic, which is essential for evaluating scalability and robustness.

For consistent evaluation across all experiments, *Subset 34-1* serves as the static test dataset, allowing for a fair assessment of model generalization on unseen data. Additionally, in the data

augmentation experiments, *Subset 42-1* and *Subset 1-1* were incorporated to introduce additional network traffic variability and attack scenarios. Their inclusion allows for an analysis of how expanding the training data influences model performance and enhances anomaly detection capabilities. These subsets were chosen to provide a structured yet diverse experimental setup, ensuring a thorough assessment of the proposed framework.

**The Subsets of CICIoT2023 Dataset**

The CICIoT2023 dataset [14], developed by the Canadian Institute for Cybersecurity (CIC), is one of the most extensive datasets designed for IoT anomaly detection. It captures network traffic from 105 different IoT devices, encompassing various categories such as consumer electronics, industrial IoT systems, medical devices, and smart home assistants. The dataset is structured to simulate real-world cybersecurity incidents by including both benign and malicious traffic, allowing for a rigorous evaluation of intrusion detection frameworks.

The CICIoT2023 dataset contains over 100 million network flow records and it includes traffic logs collected over multiple weeks, ensuring that temporal variations and evolving attack strategies are captured. The recorded network flows provide a balanced mix of normal device communications and malicious activities, enabling a thorough assessment of anomaly detection models.

A distinguishing feature of the CICIoT2023 dataset is its extensive coverage of attack types, which are categorized into seven major groups. The dataset includes distributed denial-of-service (DDoS) attacks, where large volumes of traffic are directed at IoT devices to disrupt their availability. It also contains traditional denial-of-service (DoS) attacks that exploit vulnerabilities in communication protocols to degrade device performance. Another significant attack category is reconnaissance, which involves network scanning and fingerprinting techniques used by adversaries to identify vulnerable IoT devices. Web-based attacks, such as SQL injection and cross-site scripting (XSS), are also present, simulating real-world threats targeting IoT web interfaces. Brute-force attacks are included, representing scenarios where automated tools attempt unauthorized logins using credential stuffing or dictionary attacks. The dataset further incorporates spoofing and evasion attacks, where adversaries manipulate source addresses and communication patterns to bypass security defenses. Additionally, botnet infections such as Mirai and Gafgyt are recorded, demonstrating

how IoT devices can be co-opted for large-scale coordinated cyberattacks.

The dataset is structured using flow-based traffic analysis, leveraging the CICFlowMeter tool to extract statistical attributes from network packets. It provides a wide range of features, including flow duration, packet inter-arrival times, byte-per-second transmission rates, and entropy-based measures for assessing randomness in traffic patterns. Furthermore, it includes TCP/IP header attributes such as flag counts, protocol distributions, and source-destination correlations. The dataset is available in pre-processed CSV files, making it suitable for direct integration into machine learning and deep learning models.

By incorporating the CICIoT2023 dataset into the evaluation framework, this study ensures that the proposed anomaly detection models are tested against a diverse set of real-world attack scenarios. The dataset's inclusion allows for a comparative assessment of detection performance under different network conditions, providing valuable insights into the adaptability and scalability of the developed framework.

To evaluate model performance under different network conditions, two subsets of CICIoT2023 dataset is included in the experiments to complement IoT-23 by providing additional diversity in attack scenarios and network conditions. To ensure a balanced evaluation, two subsets were selected based on their size and representational value. $DS_4$ (Smallest Subset) was chosen to assess model performance in low-data environments, which is essential for understanding how well the framework operates with limited training samples—a scenario often encountered in real-world IoT deployments. In contrast, $DS_5$ (Largest Subset) was selected due to its extensive traffic volume, allowing for an evaluation of scalability and the model's ability to handle high-volume network data. These subsets were strategically included to examine how dataset size impacts detection accuracy, training efficiency, and the framework's adaptability across different data availability scenarios.

**Justification for Dataset Naming**

The standardized naming convention ($DS_1$ – $DS_5$) is introduced to:

- Ensure clarity and consistency in experimental result discussions.
- Provide a structured reference for subset comparisons across different models.

- Facilitate reproducibility by allowing precise identification of dataset configurations.

By using this naming scheme, experimental evaluations remain organized, ensuring that dataset selection and its impact on anomaly detection results are effectively analyzed.

## 5.3 Finding the Best Model and Scenario

Analyzing the impact of feature selection, data balancing and hyperparameter tuning across different scenarios reveals some consistent patterns in the effects on model performance. By comparing scenarios, we observe both positive and negative impacts of modules on models.

Feature selection methods significantly enhanced the performance of deep learning models across various datasets. This improvement demonstrates the critical role of feature selection in optimizing model performance, particularly in the complex domain of IoT network traffic analysis. The dramatic improvements observed suggest that many features in the original datasets may be redundant or irrelevant for anomaly detection tasks, and their removal allows models to focus on the most informative aspects of the data.

CNNs exhibited remarkable improvements, with F1-scores increasing from 16% to 99% using $\chi^2$ and PCA in $DS_1$, and from 61.33% to 98% using Random Forest (RF) in $DS_5$. These substantial gains highlight the effectiveness of feature selection in enhancing CNN performance, particularly when combined with methods that can capture both linear (PCA) and non-linear (RF) relationships in the data.

NN models also demonstrated substantial enhancements, particularly with the Trank method boosting F1-scores from 43% to 99.39% in $DS_1$, and RF improving from 1% to 99.47% in $DS_3$. The dramatic improvement in $DS_3$ suggests that the original feature set may have been particularly noisy or irrelevant for this dataset, and feature selection was crucial for enabling the NN to learn meaningful patterns.

BERT models performed consistently well across all datasets, with F1-scores in $DS_1$ rising from 94.81% to 98-99% using various methods. The high initial performance of BERT models suggests their robustness in handling complex, high-dimensional data, with feature selection providing incremental but consistent improvements.

Autoencoder (AE) models, while already performing well in $DS_1$, $DS_2$, and $DS_3$, showed improvements with PCA and Mutual Information (MI) in $DS_4$ and $DS_5$. This indicates that even for models designed to learn efficient representations of the data, explicit feature selection can still provide benefits, especially on more complex datasets.

In summary, this thesis' findings underscore the importance of tailoring feature selection techniques to specific learning algorithms and datasets for optimal performance. The most effective feature selection approach can vary across different model architectures and datasets, highlighting the need for a flexible, adaptive approach to feature selection in IoT anomaly detection systems. This variability also suggests that ensemble methods combining multiple feature selection techniques might be a promising direction for future research, potentially offering more robust and consistent performance across diverse IoT network scenarios.

### 5.3.1 Impacts of Individual Feature Selection Methods

This set of experiments is to evaluate the effects of different feature selection methods on our models. Examining scenarios $S0$ and $S1$ across different datasets Figures(5.1, 5.2, 5.3, 5.4, 5.5) reveals significant positive effects of feature selection on DL models. For instance, CNN models show significant improvements, with F1-scores increasing from 16% to 99% using $\chi^2$ and PCA in $DS_1$, and from 61.33% to 98% using RF in $DS_5$. NN models also face big changes, particularly with the trank method boosting F1-scores from 43% to 99.39% in $DS_1$, and RF improving from 1% to 99.47% in $DS_3$. BERT models perform consistently across all datasets, with F1-scores rising from 94.81% to 98-99% in $DS_1$ using various methods, often coupled with reduced training times of more than 50% (e.g., from over 200 minutes to 50 minutes). AE models, while already performing well in $DS_1$, $DS_2$, and $DS_3$, show improvements with PCA and MI in $DS_4$ and $DS_5$. These results underscore the potential of feature selection to enhance DL model performance in anomaly detection tasks.

Among machine learning models, XGBoost shows a significant improvement on the $DS_1$, $DS_3$ and $DS_4$ datasets, where the F1-Score increased from 38.15% to 93.41% using the $\chi^2$ feature selection method.

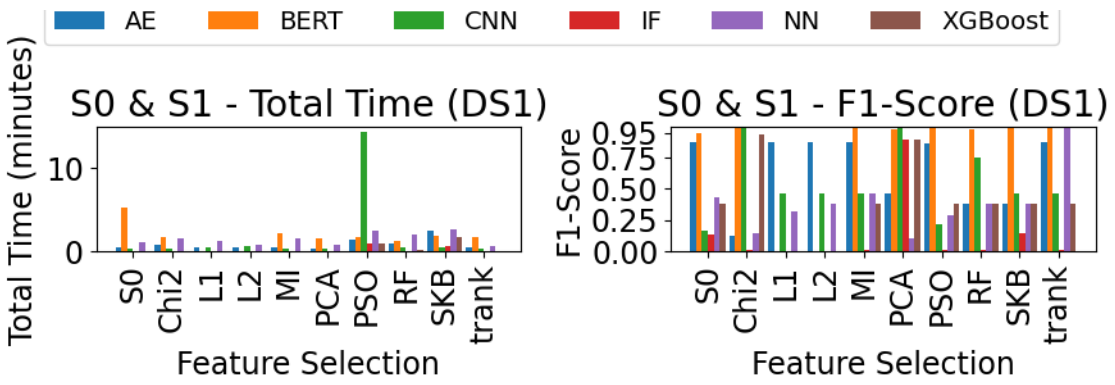


Figure 5.1: F1-score vs. total training time for dataset $DS_1$ across scenarios S0 and S1.

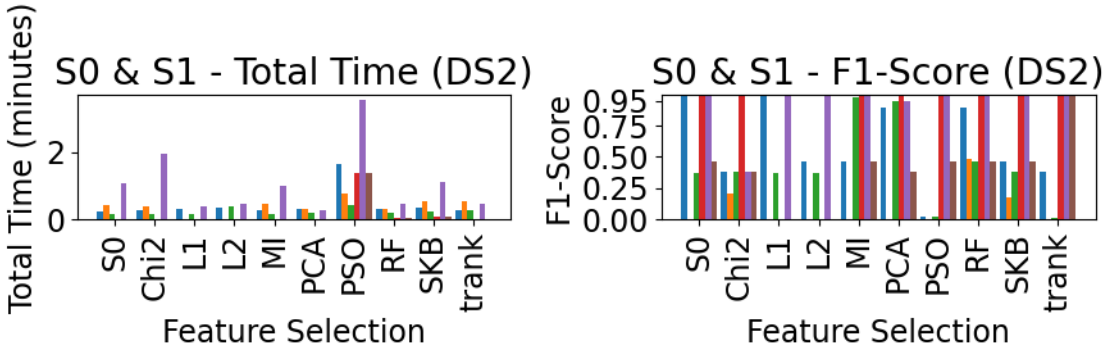


Figure 5.2: F1-score vs. total training time for dataset $DS_2$ across scenarios S0 and S1.

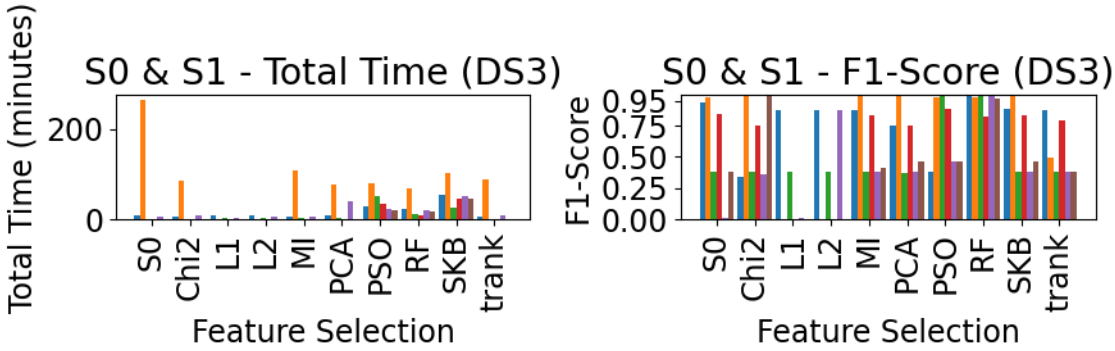


Figure 5.3: F1-score vs. total training time for dataset $DS_3$ across scenarios S0 and S1.

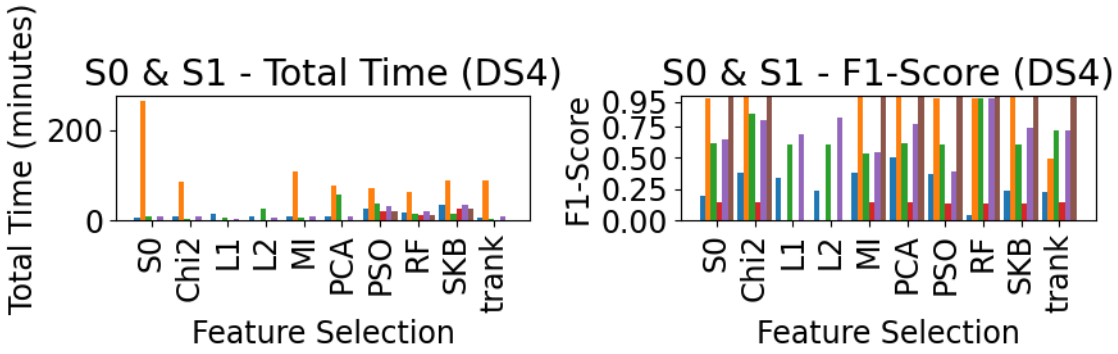


Figure 5.4: F1-score vs. total training time for dataset $DS_4$ across scenarios S0 and S1.

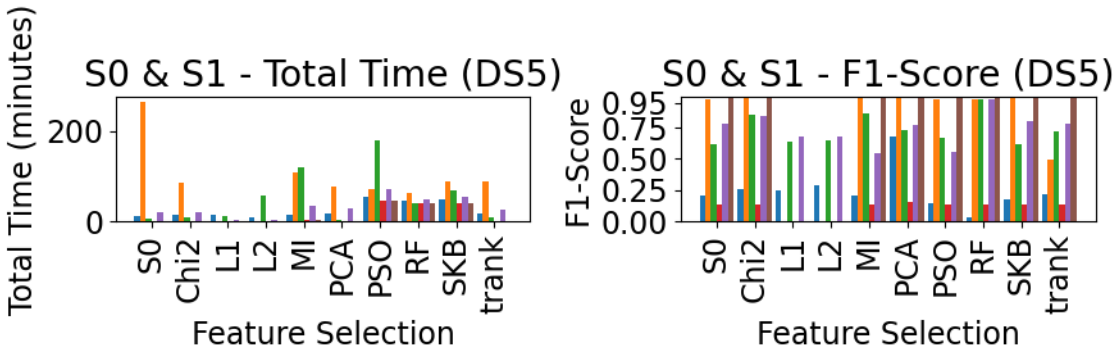


Figure 5.5: F1-score vs. total training time for dataset $DS_5$ across scenarios S0 and S1.

### 5.3.2 Impacts of Feature Selection and Data Balancing

The interplay between feature selection and data balancing significantly impacts the performance of anomaly detection models. Comparing scenarios S0 (baseline), S3 (balancing before feature selection), and S4 (balancing after feature selection) reveals notable insights. The results for S0 vs. S3 are shown in Figures 5.6, 5.7, 5.8, 5.9, and 5.10, while the results for S0 vs. S4 are presented in Figures 5.11, 5.12, 5.13, 5.14, and 5.15.

Comparing scenarios where feature selection is applied before (S3) and after (S4) data balancing revealed further insights. In $DS_4$, the CNN model improved from 60% to 98% when feature selection (using MI, PSO, SKB, and RF) was applied after balancing. Similarly, in $DS_1$, the AE model

increased from 50% to 95% with PCA applied after balancing. These results highlight that the optimal approach depends on dataset characteristics and computational constraints. Notably, PCA emerged as a consistently effective method across various scenarios and datasets, underscoring its robustness in enhancing model performance for anomaly detection tasks.

Across various datasets, CNN models showed substantial improvements, with F1-scores increasing from 16% to 70% using PCA in $DS_1$, and from 37.21% to 98.24% using MI, RF, and SKB in $DS_2$. NN models demonstrated even more dramatic enhancements, particularly in $DS_3$, where the F1-score rose from 1% to 99% using RF. AE models also benefited, with F1-scores in $DS_4$ improving from 50% to 95% using PCA.

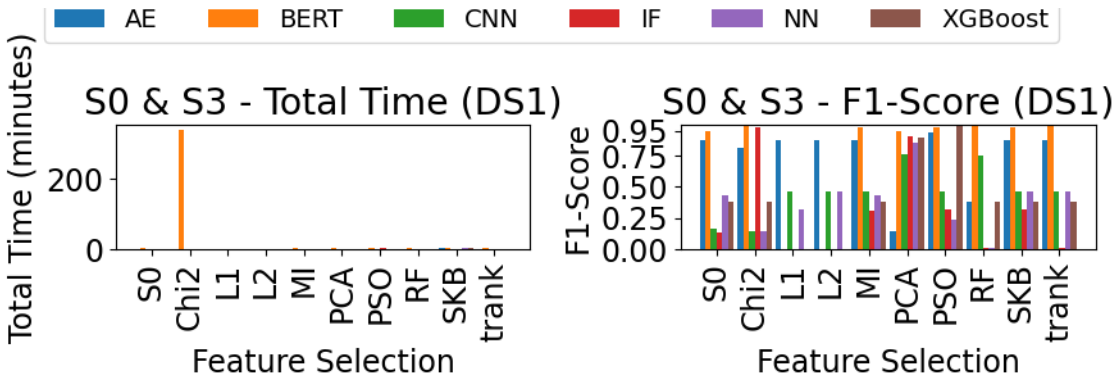


Figure 5.6: F1-score vs. total training time for dataset $DS_1$ across scenarios S0 and S3.

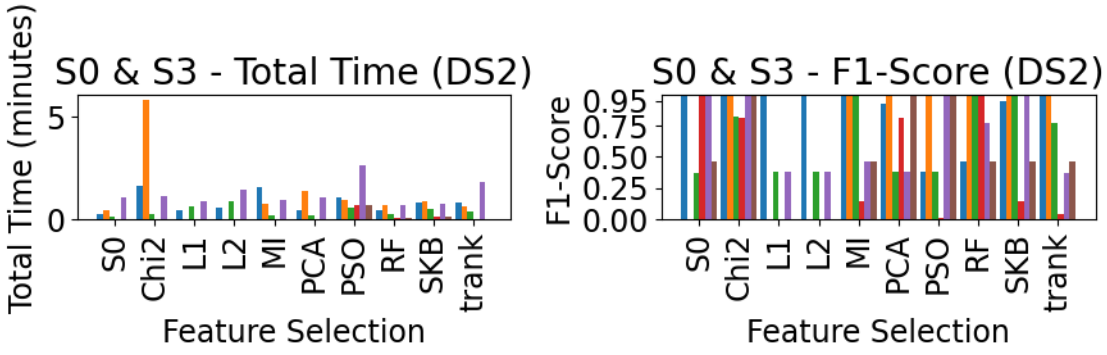


Figure 5.7: F1-score vs. total training time for dataset $DS_2$ across scenarios S0 and S3.

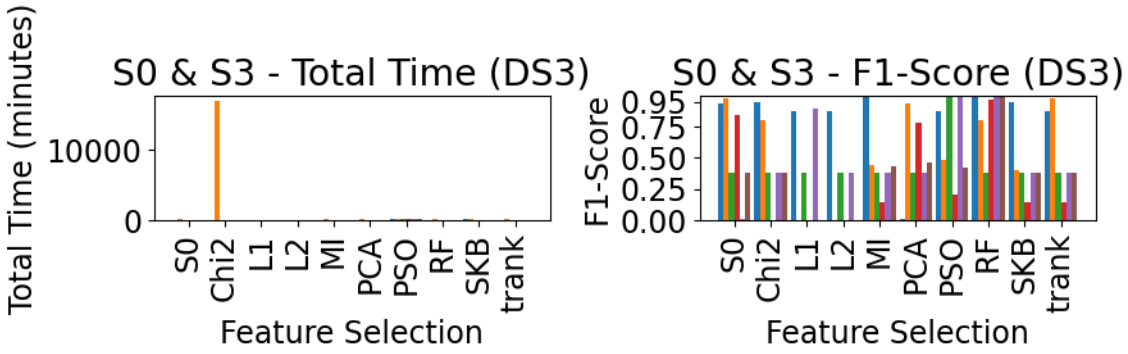


Figure 5.8: F1-score vs. total training time for dataset $DS_3$ across scenarios S0 and S3.

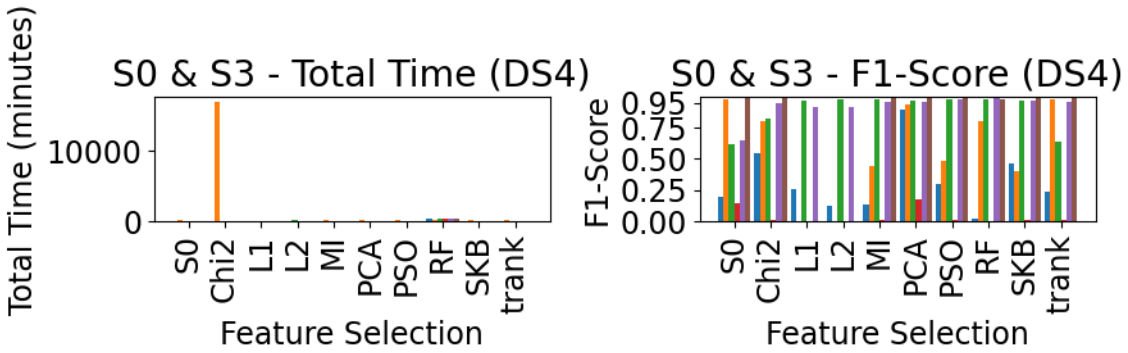


Figure 5.9: F1-score vs. total training time for dataset $DS_4$ across scenarios S0 and S3.

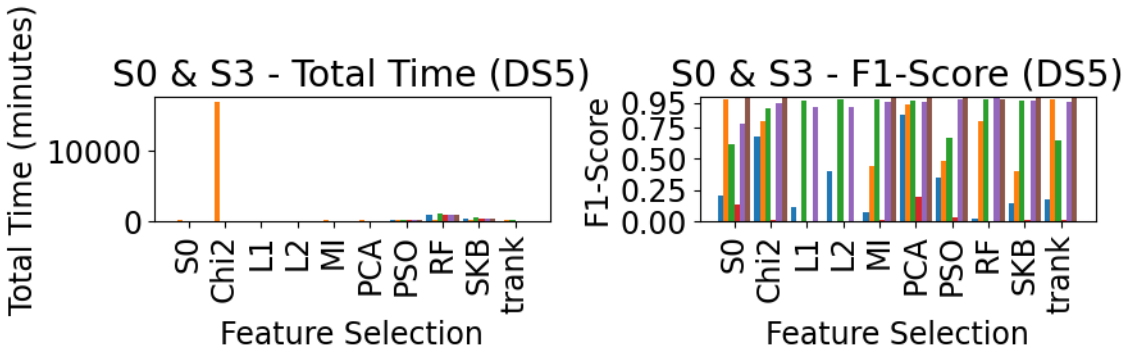


Figure 5.10: F1-score vs. total training time for dataset $DS_5$ across scenarios S0 and S3.

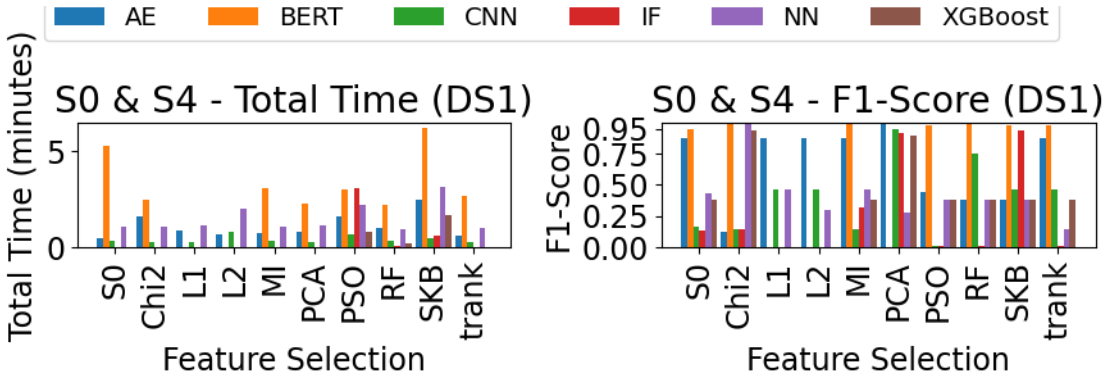


Figure 5.11: F1-score vs. total training time for dataset $DS_1$ across scenarios S0 and S4.

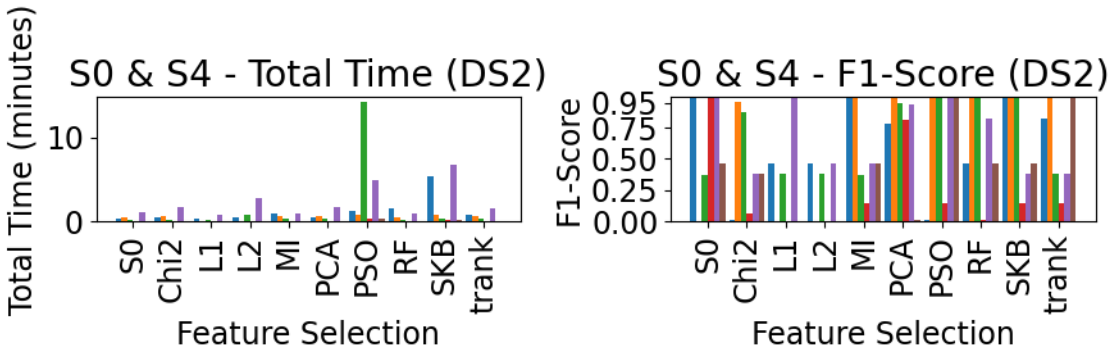


Figure 5.12: F1-score vs. total training time for dataset $DS_2$ across scenarios S0 and S4.

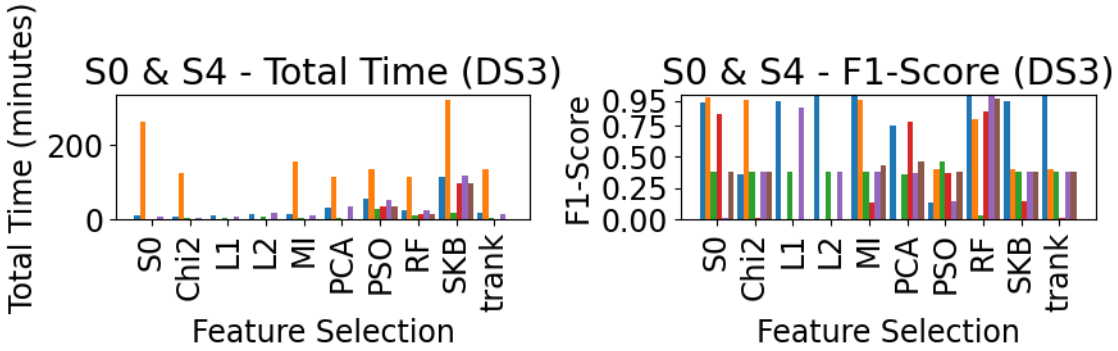


Figure 5.13: F1-score vs. total training time for dataset $DS_3$ across scenarios S0 and S4.

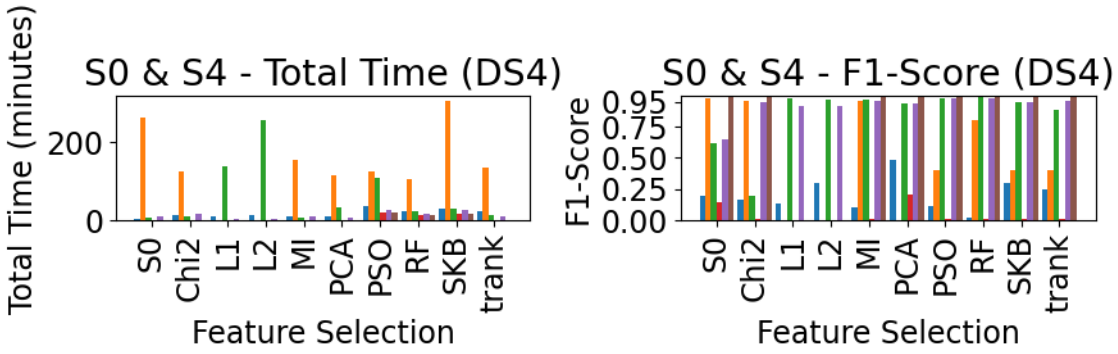


Figure 5.14: F1-score vs. total training time for dataset $DS_4$ across scenarios S0 and S4.

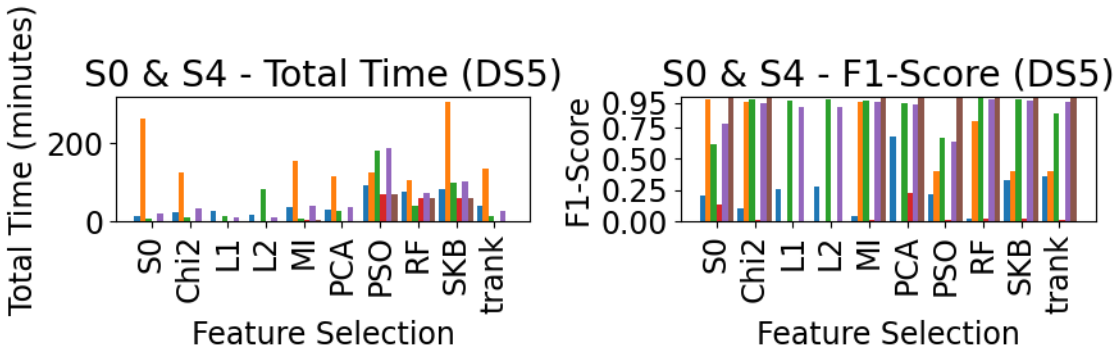


Figure 5.15: F1-score vs. total training time for dataset $DS_5$ across scenarios S0 and S4.

### 5.3.3 Impact of Hyperparameters

As shown in Figures 5.16–5.20, the effects of hyperparameter tuning on feature selection are analyzed by comparing scenarios S5 and S6. The results demonstrate varying impacts on model performance, emphasizing the intricate relationship between hyperparameter tuning and feature selection in IoT anomaly detection.

In $DS_1$, the CNN model experiences a dramatic increase in F1-Score with various feature selection methods in S6 (e.g., from 13.33% to 98.33% with $\chi^2$). This substantial improvement suggests that the combination of appropriate feature selection and optimized hyperparameters can unlock the full potential of CNNs for anomaly detection in this dataset. The effectiveness of $\chi^2$ in this

case indicates that linear relationships between features and target variables play a crucial role in distinguishing anomalies in $DS_1$.

In $DS_2$, the AE model demonstrates a positive impact of combined hyperparameter tuning and feature selection, with the F1-score increasing to 99.51% using PSO, SKB, and RF feature selection in S6. This result underscores the synergistic effect of these techniques, where the autoencoder's ability to learn efficient data representations is enhanced by both careful feature selection and optimized model parameters. The combination of PSO, SKB, and RF suggests that a multi-faceted approach to feature selection, incorporating both wrapper and filter methods, can be particularly effective for complex datasets.

In $DS_3$, the CNN model with PSO and RF faces a substantial increase in F1-Score from 42.40% to 99.56% in S6. This improvement highlights the potential of evolutionary algorithms (PSO) and ensemble methods (RF) in identifying relevant features for CNNs, especially when combined with hyperparameter optimization. The significant performance gain suggests that $DS_3$ may contain complex, non-linear relationships that are best captured by this combination of techniques.

For ML models, IF with $\chi^2$ feature selection shows improvement, with the F1-score increasing from 38.15% to 93.41% in S6 for $DS_1$. This result demonstrates that even for inherently robust models like Isolation Forest, the combination of appropriate feature selection and hyperparameter tuning can lead to substantial performance gains. The effectiveness of $\chi^2$ in this case suggests that linear feature relevance is important for anomaly detection in $DS_1$, even for tree-based models.

These results suggest that the combination of hyperparameter tuning and feature selection can significantly enhance model performance, but the effectiveness varies across different models and datasets. This variability underscores the importance of a flexible, adaptive approach to anomaly detection in IoT networks.

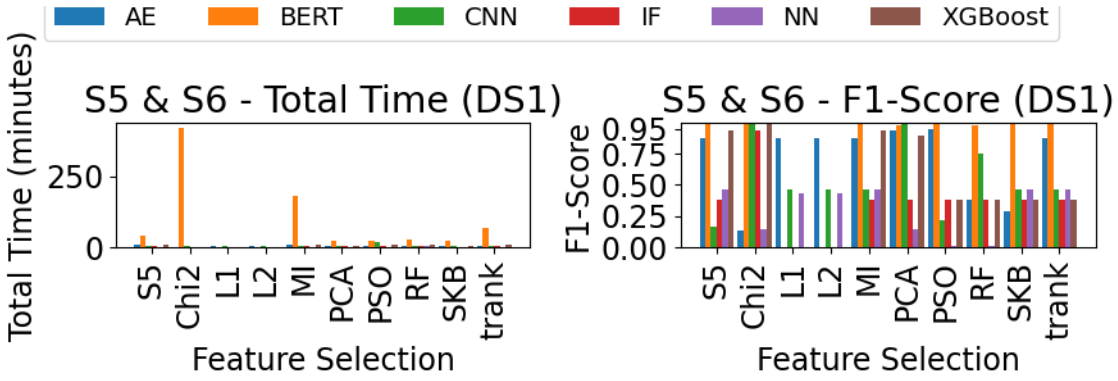


Figure 5.16: F1-score vs. total training time for dataset $DS_1$ across scenarios S5 and S6.

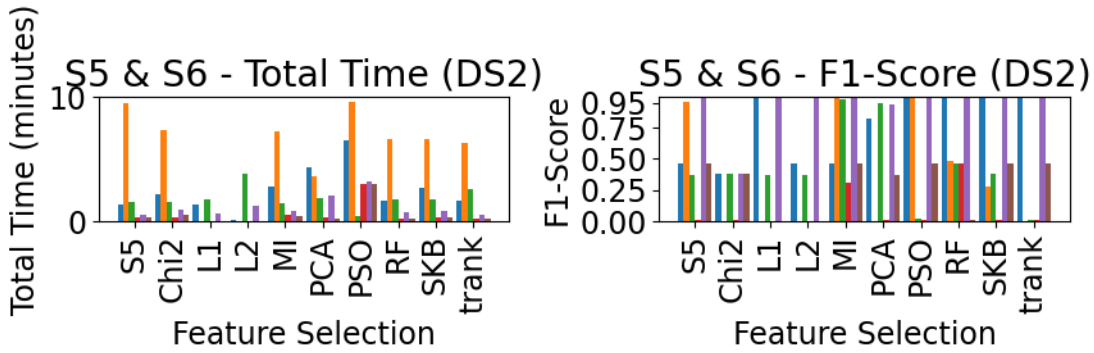


Figure 5.17: F1-score vs. total training time for dataset $DS_2$ across scenarios S5 and S6.

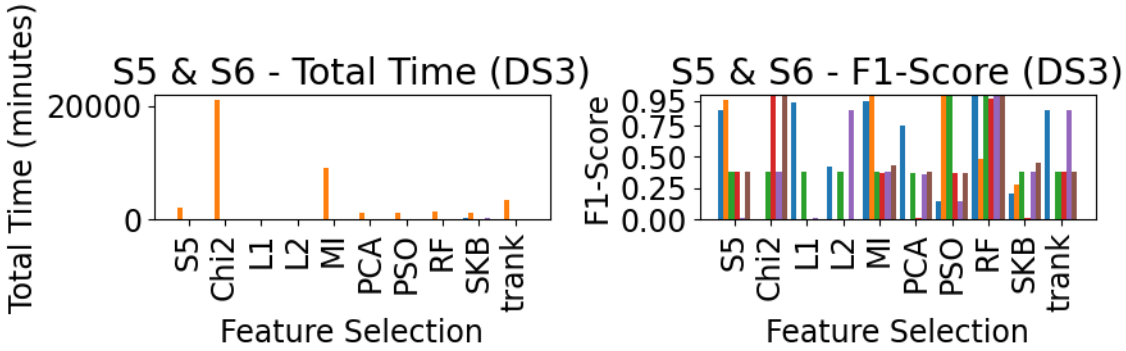


Figure 5.18: F1-score vs. total training time for dataset $DS_3$ across scenarios S5 and S6.

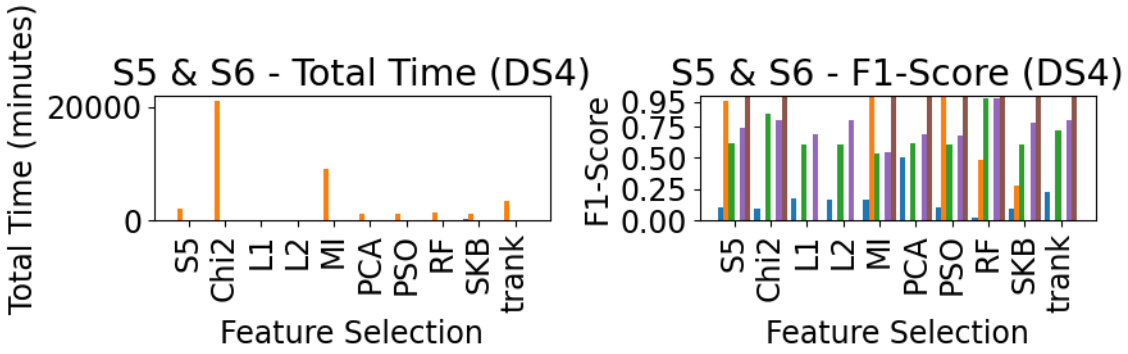


Figure 5.19: F1-score vs. total training time for dataset $DS_4$ across scenarios S5 and S6.

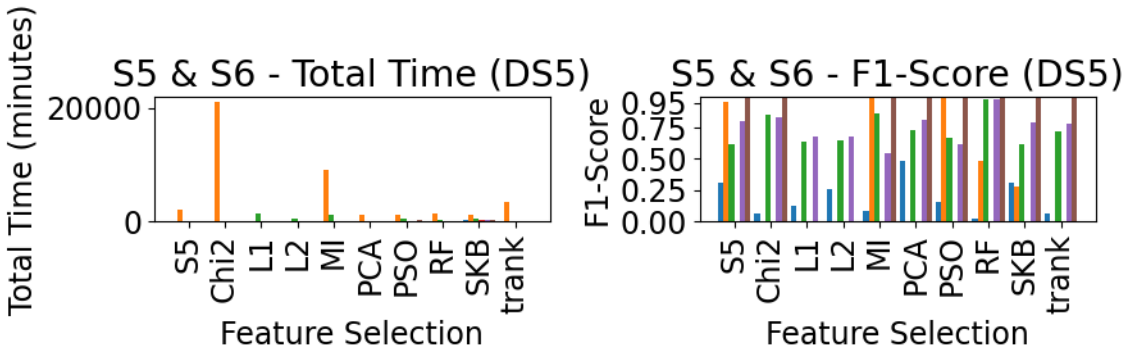


Figure 5.20: F1-score vs. total training time for dataset $DS_5$ across scenarios S5 and S6.

### 5.3.4 Best Model and Scenario Selection

The evaluation reveals that optimal strategies for anomaly detection systems vary significantly based on data and model characteristics. The BERT model with trank feature selection (scenario S8) achieves 99.70% F1-Score in 60 seconds for $DS_1$, while the AE model using Mutual Information (scenario S4) attains 99.50% F1-Score in 1,727 seconds for $DS_2$. The CNN model with MI feature selection (scenario S1) performs well on the largest dataset ($DS_3$), achieving 99.50% F1-Score.

MI and RF feature selection methods consistently enhance performance across multiple models and datasets. The order of applying feature selection and data balancing significantly impacts performance, with post-balancing feature selection often yielding better results. Table 5.1 presents the

top five combinations, balancing performance and computational efficiency across diverse datasets. These findings quantify the differences in model-feature-data interactions, providing practitioners with valuable insights for optimizing anomaly detection systems in specific use cases.

| **Dataset** | **Model** | **Feature Selection** | **Scenario** | **F1-Score** | **F1-Macro** | **Total Time (s)** |
|---|---|---|---|---|---|---|
| $DS_1$ | BERT | trank | S8 | 0.997 | 0.989 | 60.138 |
| $DS_2$ | AE | MI | S4 | 0.995 | 0.984 | 1727.421 |
| $DS_4$ | XGBoost | MI | S3 | 0.996 | 0.996 | 184.534 |
| $DS_2$ | IF | RF | S1 | 0.996 | 0.985 | 1.923 |
| $DS_3$ | CNN | MI | S1 | 0.995 | 0.984 | 1727.421 |

Table 5.1: Top-5 model and scenarios.

## 5.4 Impact of Data Augmentation

We analyze the impact of data augmentation on model performance using various dataset combinations from the IoT-23 collection. As shown in Figure 5.21, the G-Mean metric fluctuates as more datasets are incrementally added, indicating that the effects of augmentation can vary significantly. This variability underscores the complex nature of data augmentation in IoT anomaly detection, where the introduction of additional data does not always lead to uniform improvements in model performance.

While augmentation helps balance sensitivity and specificity, its benefits are not universal across all scenarios. This observation highlights the need for careful consideration when applying data augmentation techniques in IoT security contexts. The fluctuations in G-Mean suggest that augmentation can sometimes improve the model's ability to detect anomalies without compromising its performance on normal traffic, but this balance is not consistently achieved across all dataset

combinations.

| Correlation Function | Metric | Complexity Measure | Correlation | Model | Scenario (Si) | Feature Selection |
|---|---|---|---|---|---|---|
| Pearson | G-Mean | F1 | -0.66 | CNN | S1 | Chi2 |
| Spearman | G-Mean | F1 | -0.67 | CNN | S1 | Chi2 |
| Spearman | F1-Score | F1 | -0.69 | CNN | S1 | Chi2 |
| Pearson | F1-Score | C1 | -0.70 | CNN | S1 | Chi2 |
| Spearman | F1-Macro | F1 | -0.69 | CNN | S1 | Chi2 |
| Pearson | F1-Macro | C1 | -0.75 | CNN | S1 | Chi2 |
| Spearman | G-Mean | F1 | -0.53 | AE | S4 | PCA |
| MIC | F1-Score | F1 | 0.79 | NN | S1 | trank |
| MIC | G-Mean | C1 | 0.80 | CNN | S1 | Chi2 |
| MIC | G-Mean | F1 | 0.93 | NN | S4 | MI |
| MIC | F1-Score | L2 | 0.63 | XGBoost | S1 | - |

Table 5.2: Metrics and complexity measures correlations.

Table 5.2 highlights strong positive correlations, particularly for NN and CNN models, between MIC and performance metrics, suggesting that MIC could be a useful indicator of model performance under data augmentation. The Maximal Information Coefficient (MIC) appears to capture non-linear relationships between data complexity and model performance that other correlation measures might miss. This finding could be particularly valuable for developing automated systems to assess the potential effectiveness of data augmentation strategies in IoT anomaly detection.

However, other correlation measures, such as Pearson and Spearman, show that increased data complexity might negatively affect metrics like F1-Macro and G-Mean, especially in CNN models. This discrepancy between correlation measures reveals the multifaceted nature of the relationship between data complexity and model performance. The negative correlations observed with Pearson and Spearman suggest that simple linear relationships do not fully capture the impact of data augmentation on model performance.

This indicates that while data augmentation can enhance model performance, it may also introduce complexity that hinders results under certain conditions. The potential for augmentation to introduce complexity that negatively impacts performance is a crucial consideration in IoT anomaly detection, where the ability to quickly and accurately identify threats is paramount. This finding underscores the need for careful evaluation of augmentation strategies to ensure they enhance, rather than hinder, model performance.

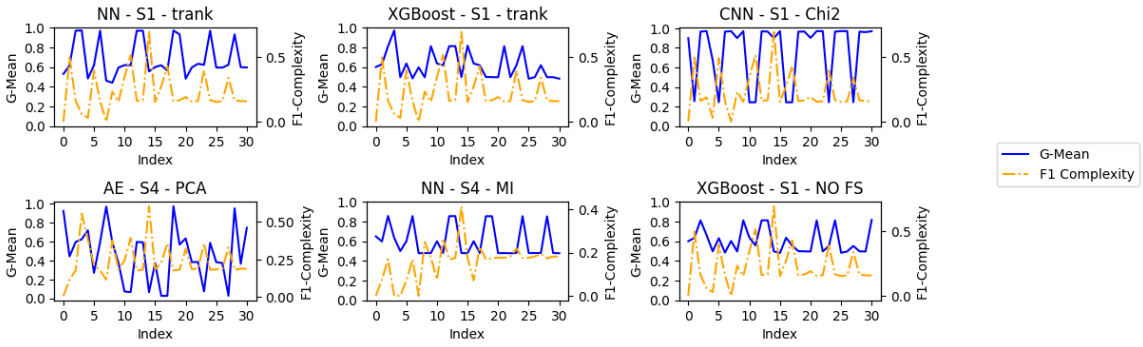


Figure 5.21: Correlation between data complexity and G-Mean.

Future research should focus on understanding these dynamics more clearly and identifying the optimal conditions for using data augmentation effectively. This could involve developing more sophisticated metrics for assessing data complexity in the context of IoT network traffic, as well as exploring adaptive augmentation techniques that can adjust based on the characteristics of the existing dataset and the specific requirements of the anomaly detection task. Additionally, investigating the relationship between data complexity measures and specific types of IoT attacks could provide valuable insights for tailoring augmentation strategies to improve detection of particular threat categories.

## 5.5 Impacts of Combined Feature Selection Methods

The results of this thesis reveal that combined feature selection methods do not exhibit a consistent impact when applied across models, scenarios, and datasets. In some cases, they significantly enhance performance (e.g., majority voting in $DS_2$ improved F1-score from 30% to over 95%). However, effectiveness varies widely, with some combinations yielding substantial improvements

while others show negligible or negative impacts. This variability highlights the dataset-specific nature of feature selection effectiveness and suggests that the optimal combination of feature selection methods may depend on the underlying characteristics of the data, the specific anomalies being detected, and the chosen model architecture.

Computational cost emerges as a critical factor, sometimes outweighing performance gains. For example, BERT on $DS_5$ achieved 99% F1-Score with RF feature selection in 39 minutes, but performance decreased when applying all feature selections, taking 134 minutes. This variability emphasizes the need for case-by-case evaluation, considering both performance and computational efficiency for specific datasets and models. It suggests that an adaptive approach, potentially incorporating automated feature selection method choice based on dataset characteristics and model performance, could be more effective in IoT anomaly detection systems. Future research could focus on developing frameworks for automatically selecting the most appropriate combination of feature selection methods and exploring techniques to reduce the computational cost of combined approaches.

## 5.6 AMETIS Framework

Based on our methodological approach, we develop AMETIS [18] (named after **A**thena and **Metis**, the symbols of deep and strategic decision-making), a framework that implements our methodology for evaluating and optimizing IoT anomaly detection systems. AMETIS provides a systematic way to assess different combinations of features, models, and configurations.

The term *framework* is used to describe AMETIS because it provides a structured, modular, and adaptable approach for evaluating and optimizing anomaly detection models for IoT security. A framework, in the context of machine learning and cybersecurity, is defined as a set of systematic procedures, tools, and methodologies that guide the development and evaluation of solutions in a specific domain [124, 125].

AMETIS qualifies as a framework due to the following characteristics:

- **Modular Design.** AMETIS is built with distinct components for data preprocessing, feature selection, data balancing, model evaluation, and hyperparameter tuning. Each component can

be customized or replaced, allowing flexibility in anomaly detection experiments.

- **Adaptability and Extensibility.** The framework is designed to be easily applied to different datasets without requiring fundamental changes to its architecture. Users can input their own datasets by adapting the provided interfaces and scripts. The modular nature of AMETIS ensures that adapting it to various data distributions or new datasets do not require major code alterations or architectural changes. It evaluates multiple machine learning and deep learning models, making it suitable for diverse security environments. New feature selection methods or anomaly detection algorithms can be integrated without major modifications to the core structure.

- **Automated Empirical Recommendations.** AMETIS evaluates different models and scenarios through various combinations of feature selection, data augmentation, hyperparameter tuning, and data balancing. Based on these empirical experiments, AMETIS can suggest which modules should be applied and in what order to optimize performance. This capability helps users optimize anomaly detection configurations through systematic, scenario-based evaluations rather than purely dataset-driven predefined assumptions.

- **Systematic Evaluation Process.** AMETIS implements a structured pipeline for comparing different anomaly detection models under multiple experimental scenarios. This ensures that feature selection methods and data augmentation strategies can be objectively assessed across datasets.

- **Reproducibility and Open-source Availability.** The framework is implemented with well-documented configurations, allowing other researchers to replicate experiments or extend its functionality [126].

Given these characteristics, AMETIS functions as more than a small-scale test or individual model evaluation tool—it provides a structured methodology for optimizing IoT anomaly detection, making the term *framework* appropriate. However, if further expansion is needed to include additional automation or real-time integration, future iterations could refine its scope to better align with industrial frameworks used in production environments.

### 5.6.1 Extensibility

AMETIS is designed to be extensible, allowing:

- Integration of new models
- Addition of custom evaluation metrics
- Support for different dataset formats
- Custom experimental scenarios

The source code and documentation for AMETIS are available in our public repository[1], enabling reproducibility and extension of our work.

## 5.7 Automated Machine Learning (AutoML)

AutoML has gained significant attention as a methodology for automating various aspects of machine learning model development, including hyperparameter optimization, feature engineering, model selection, and pipeline configuration [127–129]. By systematically exploring a wide range of configurations, AutoML frameworks such as *Google AutoML* [130], *AutoKeras* [131], *H2O AutoML* [132], and *TPOT* [133] aim to minimize the need for manual intervention, thus making machine learning more accessible to non-experts while optimizing model performance.

Despite its advantages, AutoML was not incorporated into this study for several reasons. First, AutoML solutions tend to operate as black-box systems, offering limited interpretability of the decision-making process [134]. Given the importance of explainability in IoT anomaly detection, particularly for security applications where transparency is crucial, a more structured and interpretable approach was preferred. Unlike AutoML, our proposed framework allows researchers to explicitly analyze the impact of different feature selection methods, hyperparameter settings, and data augmentation strategies on model performance.

Second, AutoML frameworks typically prioritize model accuracy over computational efficiency,

[1] https://github.com/OCyberLab/Ametis

often requiring substantial computational resources and prolonged execution times [135]. In contrast, our study systematically optimizes feature selection and hyperparameter tuning while maintaining computational feasibility, ensuring that models remain practical for IoT anomaly detection.

Moreover, AutoML does not inherently evaluate the impact of *data augmentation* and *data complexity metrics*, which are central to our research. While some frameworks offer feature selection capabilities, they do not explicitly assess whether data augmentation enhances or degrades model performance. Our study provides a systematic investigation of *when and how data augmentation should be applied*, ensuring that it does not introduce unnecessary noise or computational overhead.

By explicitly designing a framework that evaluates feature selection, data balancing, and augmentation strategies within a structured pipeline, our approach provides deeper insights into the effectiveness of various machine learning techniques. This level of customization and interpretability is not readily available in existing AutoML solutions, making our framework a more suitable choice for IoT anomaly detection.

# Chapter 6

# Discussion and Conclusion

In this chapter, we analyze the proposed approach's applicability, strengths, limitations, and broader implications in the context of anomaly detection. The widespread adoption of deep learning techniques for anomaly detection frequently relies on assumptions, such as deep learning models' inherent capability to bypass explicit feature selection and the belief that more augmented data consistently improves model performance. However, this research systematically revisits and challenges these assumptions, providing empirical evidence that blind reliance on these principles may not always yield optimal outcomes.

This research fundamentally advances our understanding of IoT security optimization through a systematic investigation of the interplay between feature selection, data augmentation, and model architecture. To address the complexity of optimizing anomaly detection systems, we developed the AMETIS framework. AMETIS offers practitioners a flexible, adaptive solution for customizing anomaly detection systems according to specific IoT network traffic log requirements. By selectively configuring features, models, hyperparameters, balanced data, and augmented data, AMETIS effectively addresses the observed variability across different IoT scenarios.

Evaluations conducts on two large real-world IoT datasets—CICIoT2023 and IoT-23, demonstrate the framework's efficacy in enhancing anomaly detection performance across various deep learning and machine learning models. For example, applying appropriate feature selection techniques significantly improves CNN performance, with F1-scores increasing up to 99% using $\chi^2$

and PCA methods on certain datasets. These outcomes underscore the importance of tailoring feature selection strategies to specific model architectures and datasets, rather than applying a generic solution.

Moreover, the sequence of feature selection and data balancing emerges as a critical determinant of performance. Specifically, performing feature selection after data balancing consistently produces superior results, highlighting the necessity for an integrated, holistic approach to preprocessing rather than treating these processes independently.

Computational efficiency was another significant factor. Notably, BERT models demonstrates substantial efficiency improvements, reducing training time by over 50% while simultaneously enhancing F1-scores. This underscores the importance of balancing performance gains with computational resource considerations.

Despite these encouraging results, certain limitations remain. The current research primarily evaluates performance using two specific datasets, potentially limiting the generalizability of the findings. Furthermore, generative models, which have shown promise in related domains, have not yet been integrated into AMETIS.

Future research efforts will aim to overcome these limitations by incorporating generative models into the framework, conducting real-world validations, and extending evaluations across diverse datasets. These enhancements are expected to strengthen AMETIS's applicability and provide more precise metrics for data augmentation decisions, further bolstering its utility for anomaly detections.

In conclusion, this study highlights the inherent complexities involves in optimizing anomaly detection systems and challenges prevalent assumptions within the field. By providing a robust, data-driven, and flexible approach to model configuration, the AMETIS framework represents a significant step forward, offering practical and theoretical contributions to the effective and efficient implementation of anomaly detection systems.

## 6.1 Broader Impact

The implications of our findings extend far beyond immediate technical improvements. Our research challenges fundamental assumptions about deep learning applications, suggesting a need to

reevaluate current approaches to system design. The demonstrated importance of feature selection timing and data complexity measurement provides new frameworks for security system optimization.

For practitioners, our findings offer immediate practical benefits through improved resource utilization and detection accuracy. The computational efficiency gains achieved through optimal feature selection and model configuration provide a pathway for enhancing security operations within existing resource constraints.

In the broader context of cybersecurity, our research suggests new approaches to system design that consider the complex interplay between data characteristics, model architecture, and security objectives. This holistic approach to security system optimization offers valuable insights for other domains facing similar challenges in threat detection and response.

## 6.2 Final Reflections

This research has shown that building effective anomaly detection systems is more complex than it might seem. Success depends on carefully choosing features, balancing data, selecting the right model, and tuning the settings. We found that simply applying deep learning methods without a clear strategy is not enough to handle the challenges of real-world data.

The AMETIS framework created in this thesis is a step forward in anomaly detection. It helps evaluate different combinations of features, data, and models to find the best setup for specific needs. This work challenges some common ideas, like the belief that deep learning always works without feature selection or that adding more data always improves results. Instead, we learned that these steps need to be carefully planned and adjusted based on the situation.

As anomaly detection becomes more important in many areas, the methods and tools from this research can help improve systems. AMETIS is both a practical tool and a guide for designing better anomaly detection approaches, making it easier to tackle the challenges of working with complex data.

# Bibliography


[1] Ruming Tang, Zheng Yang, Zeyan Li, Weibin Meng, Haixin Wang, Qi Li, Yongqian Sun, Dan Pei, Tao Wei, Yanfei Xu, and Yan Liu. Zerowall: Detecting zero-day web attacks through encoder-decoder recurrent neural networks. *IEEE Symposium on Security and Privacy*, 2020.

[2] Kai Wang, Zhiliang Wang, Dongqi Han, Wenqi Chen, Jiahai Yang, Xingang Shi, and Xia Yin. BARS: Local robustness certification for deep learning based traffic analysis systems. 01 2023.

[3] Yang Wang, Yue Yang, Hu Wang, and Philip S Yu. Imbalanced learning for anomaly detection: a comprehensive review. *ACM CSUR*, 55(2):1–37, 2021.

[4] Feng Wei, Hongda Li, Ziming Zhao, and Hongxin Hu. xNIDS: Explaining deep learning-based network intrusion detection systems for active intrusion responses. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 4337–4354, Anaheim, CA, August 2023. USENIX Association.

[5] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey, 2019.

[6] Mohammad Al Olaimat, Dongeun Lee, Youngsoo Kim, Jonghyun Kim, and Jinoh Kim. A learning-based data augmentation for network anomaly detection. pages 1–10, 2020.

[7] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pages 117–122, 2018.

[8] Ziming Zhao, Zhaoxuan Li, Jiongchi Yu, Fan Zhang, Xiaofei Xie, Haitao Xu, and Binbin Chen. CMD: Co-analyzed IoT malware detection and forensics via network and hardware domains. *IEEE Transactions on Mobile Computing*, 22(1):1–14, 2023.

[9] Céline Minh, Kevin Vermeulen, Cédric Lefebvre, Philippe Owezarski, and William Ritchie. An explainable-by-design ensemble learning system to detect unknown network attacks. In *2023 19th International Conference on Network and Service Management (CNSM)*, pages 1–9, 2023.

[10] Yutao Dong, Qing Li, Kaidong Wu, Ruoyu Li, Dan Zhao, Gareth Tyson, Junkun Peng, Yong Jiang, Shutao Xia, and Mingwei Xu. HorusEye: A realtime IoT malicious traffic detection framework using programmable switches. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 571–588, Anaheim, CA, 2023.

[11] Chuanpu Fu, Qi Li, Meng Shen, and Ke Xu. Realtime robust malicious traffic detection via frequency domain analysis. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2021.

[12] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. A survey on iot security: Application areas, security threats, and solution architectures. *IEEE Access*, 7:82721–82743, 2019.

[13] Gulshan Kumar, Gaurav Kumar, Mayank Bhatia, and Huaming Chen. A comprehensive survey of ai-enabled cyber security solutions for iot networks. *Security and Communication Networks*, 2022, 2022.

[14] Euclides Carlos Pinto Neto, Sajjad Dadkhah, Raphael Ferreira, Alireza Zohourian, Rongxing Lu, and Ali A. Ghorbani. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment, 2023. ISSN 1424-8220. URL `https://www.mdpi.com/1424-8220/23/13/5941`.

[15] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga. IoT-23: A labeled dataset with malicious and benign IoT network traffic, May 2021. URL `https://doi.org/10.5281/zenodo.4743746`.

[16] Sherali Zeadally, Farhan Shaikh, and Ayesha Talpur. Energy-efficient security and privacy mechanisms in iot: A comprehensive review. *IEEE Internet of Things Journal*, 10(8):7158–7183, 2023.

[17] Jean-Paul A Yaacoub, Ola Salman, Hassan N Noura, Nora Kaaniche, Ali Chehab, and Mohammad Malli. Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and Microsystems*, 77:103201, 2020.

[18] Alireza Toghiani Khorasgani, Paria Shirani, and Suryadipta Majumdar. An empirical study on learning models and data augmentation for iot anomaly detection. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, 2024.

[19] Alice Zheng and Amanda Casari. Feature engineering for machine learning: principles and techniques for data scientists. 2018.

[20] Chen Li, Wei Zhou, and Xin Zhang. Statistical feature extraction techniques for iot anomaly detection: A review. *IEEE Internet of Things Journal*, 10:1501–1515, 2023.

[21] Guang Sun, Kegen Chen, Qiang Guo, Yong Jiang, and Zhi Zhou. Signal processing techniques in network-aided positioning: A survey. *IEEE Signal Processing Magazine*, 22(4): 12–23, 2005.

[22] Thuy TT Nguyen and Grenville Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, 10(4):56–76, 2011.

[23] Dominik Olszewski, Marcin Iwanowski, and Waldemar Graniszewski. Dimensionality reduction for detection of anomalies in the iot traffic data. *Future Generation Computer Systems*, 151:137–151, 2024. ISSN 0167-739X.

[24] T. Tony Cai and Rong Ma. Theoretical foundations of t-sne for visualizing high-dimensional clustered data, 2022.

[25] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[26] Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *Ai Communications*, 30:169–190,, 05 2017. doi: 10.3233/AIC-170729.

[27] Avis Priyati, Arif Dwi Laksito, and Heri Sismoro. The comparison study of matrix factorization on collaborative filtering recommender system. In *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, pages 177–182, 2022.

[28] I.T. Jolliffe. *Principal component analysis (2nd edition)*. Springer Verlag, Berlin, 2002.

[29] M. Dash and H. Liu. Feature selection using principal feature analysis. *International Conference on Knowledge Discovery and Data Mining*, pages 64–73, 1997.

[30] R. Kohavi and G. H. John. Wrappers for feature subset selection. In *Artificial Intelligence*, volume 97, pages 273–324, 1997.

[31] M. C. Chandrashekar, A. K. Qin, and P. N. Suganthan. Embedded feature selection: An overview. *Journal of Information and Data Management*, 4(1):23, 2013.

[32] Kiran S Balagani and Vir V Phoha. Feature selection for intrusion detection using neuroevolution and correlation-based feature selection. *Journal of Information Assurance and Security*, 5(1):369–378, 2010.

[33] Myra L. Samuels, Jeffrey A. Witmer, and Andrew A. Schaffner. *Statistics for the Life Sciences*. Pearson, 2012. Chapter on T-test.

[34] Huan Liu and Rudy Setiono. Chi2: Feature selection and discretization of numeric attributes. *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence*, pages 388–391, 1995.

[35] N. Mukhopadhyay. correlation coefficient. pages 315–318, 2011.

[36] John Lee Rodgers and W. Alan Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[37] George Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305, Aug 2003.

[38] Yvan Saeys, Inaki Inza, and Pedro Larra naga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[39] Jaime A Vergara and Pablo A Est'evez. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186, 2014.

[40] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[41] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[42] Allan W Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 100(9):1100–1103, 1971.

[43] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[44] Thomas Marill and Donald Green. On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9(1):11–17, 1963.

[45] P. M. Narendra and Keinosuke Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26(9):917–922, 1977.

[46] R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43.

[47] James Kennedy and Russell Eberhart. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4:1942–1948, 1995.

[48] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, January 1996.

[49] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[50] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[51] David JC MacKay. Bayesian methods for adaptive models. *California Institute of Technology, Pasadena, CA*, 1994.

[52] Ana C. Lorena, Luis P.F. Garcia, Jens Lehmann, Marcilio C.P. Souto, and Tin Kam Ho. How complex is your classification problem? a survey on measuring classification complexity. 52 (5):1–34, 2021.

[53] Salvador Garcia, Joaquin Derrac, Jose Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2015.

[54] Albert Orriols-Puig, Jorge Bermúdez, and David E Goldberg. A fsdepthdresstereuse feature selection for nearest neighbor classifiers. *Pattern Recognition Letters*, 31(12):1758–1764, 2010.

[55] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[56] Michael R. Smith, Tony Martinez, and Christophe Giraud-Carrier. An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256, 2014.

[57] Tu Bao Ho, Bipin M Jain, and Ram Sewak Srivastava. Complexity of classification problems and artificial neural networks. *Neurocomputing*, 43(1-4):219–230, 2002.

[58] Bartosz Krawczyk. Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5:221–232, 2016.

[59] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[60] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *IEEE International Joint Conference on Neural Networks*, pages 1322–1328, 2008.

[61] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. *Advances in intelligent computing*, pages 878–887, 2005.

[62] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*, 2014.

[63] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[64] Mulyana Saripuddin, Azizah Suliman, Sera Syarmila Sameon, and Bo Norregaard Jorgensen. Random undersampling on imbalance time series data for anomaly detection. In *Proceedings of the 2021 4th International Conference on Machine Learning and Machine Intelligence*, pages 151–156, 2021. doi: 10.1145/3490725.3490748.

[65] Ivan Tomek. Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:769–772, 1976.

[66] Inderjeet Mani and I. Zhang. knn approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of Workshop on Learning from Imbalanced Datasets*, 2003.

[67] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421, 1972.

[68] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. In *ACM SIGKDD Explorations Newsletter*, volume 6, pages 20–29, 2003.

[69] Samee U. Khan, Syed Hassan Ahmed Sherazi, Abdullah Gani, and Mohsen Guizani. Cost-sensitive learning and its applications in internet of things. *Future Generation Computer Systems*, 89:640–651, 2018.

[70] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[71] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

[72] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[73] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy*, 2018.

[74] Zhenguo Hu, Hirokazu Hasegawa, Yukiko Yamaguchi, and Hajime Shimada. Enhancing detection of malicious traffic through fpga-based frequency transformation and machine learning. *IEEE Access*, 12:1–12, 2024.

[75] Qingjun Yuan, Chang Liu, Wentao Yu, Yuefei Zhu, Gang Xiong, Yongjuan Wang, and Gaopeng Gou. BoAu: Malicious traffic detection with noise labels based on boundary augmentation. *Computers & Security*, 131:103300, 2023.

[76] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[77] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[78] Loris Nanni, Michelangelo Paci, Sheryl Brahnam, and Alessandra Lumini. Comparison of different image data augmentation approaches. *Journal of Imaging*, 7(12), 2021.

[79] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[80] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2nd edition edition, 2008.

[81] Yixiang Wang, Jiqiang Liu, Xiaolin Chang, Jinqiang Wang, et al. On the combination of data augmentation method and gated convolution model for building effective and robust intrusion detection. *Cybersecurity*, 3(1):1–12, 2020.

[82] Swee Kiat Lim, Yi Loo, Ngoc-Trung Tran, Ngai-Man Cheung, Gemma Roig, and Yuval Elovici. Doping: Generative data augmentation for unsupervised anomaly detection with gan. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1122–1127. IEEE, 2018.

[83] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[84] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[85] Danni Yuan, Kaoru Ota, Mianxiong Dong, Xiaoyan Zhu, Tao Wu, Linjie Zhang, and Jianfeng Ma. Intrusion detection for smart home security based on data augmentation with edge computing. In *ICC*, pages 1–6. IEEE, 2020.

[86] Augustus Odena, Chris Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651. PMLR, 2017.

[87] Juan Ignacio Iturbe Araya and Helena Rif
-Pous. Anomaly-based cyberattacks detection for smart homes: A systematic literature review. *Internet of Things*, 22:100792, 2023.

[88] Bhuvana Jayaraman, Mirnalinee Thanga Nadar, Anirudh Anand, and Karthik Raja Anandan. Detecting malicious iot traffic using machine learning techniques. *Romanian Journal of Information Technology and Automatic Control*, 33(4):47–58, 2023.

[89] João Vitorino, Nuno Oliveira, and Isabel Praça. Adaptative perturbation patterns: Realistic adversarial learning for robust intrusion detection. *Future Internet*, 14(4):108, 2022.

[90] Falaq Jeelani, Dhajvir Singh Rai, Ankit Maithani, and Shubhi Gupta. The detection of iot botnet using machine learning on iot-23 dataset. In *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, volume 2, pages 634–639, 2022.

[91] Michael Austin. *IOT malicious traffic classification using machine learning*. West Virginia University, 2021.

[92] Ferhat Ozgur Catak, Javed Ahmed, Kevser Sahinbas, and Zahid Hussain Khand. Data augmentation based malware detection using convolutional neural networks. *PeerJ Computer Science*, 7:e346, 2021.

[93] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[94] Abdulmohsen Alharbi, Md. Abdul Hamid, and Husam Lahza. Predicting malicious software in iot environment based on machine learning and data mining techniques. *International Journal of Advanced Computer Science and Applications*, 2022.

[95] Chaw Su Htwe, Zin Thu Thu Myint, and Yee Mon Thant. Iot security using machine learning methods with features correlation. *Journal of Computing Theories and Applications*, 2(2): 151–163, 2024.

[96] Juan Ignacio Iturbe-Araya and Helena Rif
-Pous. Impact of dataset composition on machine learning performance for anomaly detection in smart home cybersecurity. In *2024 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–10. IEEE, 2024.

[97] Bin Sun, Renkang Geng, Lu Zhang, Shuai Li, Tao Shen, and Liyao Ma. Securing 6g-enabled iot/iov networks by machine learning and data fusion. *EURASIP Journal on Wireless Communications and Networking*, 2022(1):113, 2022.

[98] Aniss Chohra, Paria Shirani, ElMouatez Billah Karbab, and Mourad Debbabi. Chameleon: Optimized feature selection using particle swarm optimization and ensemble methods for network anomaly detection. *Computers & Security*, 117:102684, 2022.

[99] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[100] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[101] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth IEEE international conference on data mining*, pages 413–422. IEEE, 2008.

[102] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[103] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

[104] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[105] Karl Pearson. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

[106] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[107] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[108] J.P. Guilford. *Fundamental Statistics in Psychology and Education*. McGraw-Hill, 5th edition edition, 1965.

[109] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.

[110] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python package to tackle the curse of imbalanced datasets in machine learning. `https://imbalanced-learn.org`, 2017. Accessed: 2025-01-28.

[111] François Chollet et al. Keras: The python deep learning library. `https://keras.io`, 2015. Accessed: 2025-01-28.

[112] Martín Abadi et al. Tensorflow: An open source machine learning framework for everyone. `https://www.tensorflow.org`, 2015. Accessed: 2025-01-28.

[113] Wes McKinney et al. Pandas: Python data analysis library. `https://pandas.pydata.org`, 2010. Accessed: 2025-01-28.

[114] Charles R. Harris et al. Numpy: The fundamental package for scientific computing in python. https://numpy.org, 2020. Accessed: 2025-01-28.

[115] Fabian Pedregosa et al. Scikit-learn: Machine learning in python. https://scikit-learn.org, 2011. Accessed: 2025-01-28.

[116] J. D. Hunter. Matplotlib: A 2d graphics environment. https://matplotlib.org, 2007. Accessed: 2025-01-28.

[117] Miron B. Kursa and Witold R. Rudnicki. Borutapy: Wrapper for feature selection. https://github.com/scikit-learn-contrib/boruta_py, 2010. Accessed: 2025-01-28.

[118] Sebastian Raschka. Mlxtend: A library of machine learning extensions and utilities. https://rasbt.github.io/mlxtend/, 2018. Accessed: 2025-01-28.

[119] Marko Robnik-Šikonja and Igor Kononenko. Relieff: Feature weighting algorithm. https://github.com/ynsfzln/ReliefF, 1997. Accessed: 2025-01-28.

[120] David Martin Powers. Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[121] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.

[122] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[123] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *ICML*, volume 97, pages 179–186, 1997.

[124] Y Zhang, Y Zhao, and J Liu. A comprehensive review of cybersecurity frameworks. *ACM Computing Surveys*, 50(6):1–36, 2018.

[125] F Van Lingen, S Jansen, and J Pepple. Software frameworks: Definitions, benefits, and evaluation methods. *IEEE Transactions on Software Engineering*, 45:1215–1230, 2019.

[126] J Bozic and F Skopik. Reproducibility in machine learning-based security: A framework perspective. *Journal of Cybersecurity*, 7(1):1–17, 2021.

[127] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021. ISSN 0950-7051.

[128] Matthias Feurer and Frank Hutter. *Hyperparameter Optimization*, pages 3–33. Springer International Publishing, Cham, 2019.

[129] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning: Methods, Systems, Challenges*. Springer Publishing Company, Incorporated, 1st edition, 2019. ISBN 3030053172.

[130] Google automl. `https://cloud.google.com/automl`, 2023.

[131] Haifeng Jin, Qingquan Song, and Xia Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 1946–1956, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016.

[132] Eric Ledell, Hatice Arslan, Kai Yang, et al. H2o automl: Scalable automatic machine learning. *Journal of Open Source Software*, 5(48):1–3, 2020. doi: 10.21105/joss.02288.

[133] Randal S. Olson and Jason H. Moore. *TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning*, pages 151–160. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05318-5.

[134] Samuel de Oliveira, Oguzhan Topsakal, and Onur Toker. Benchmarking automated machine learning (automl) frameworks for object detection. *Information*, 15(1), 2024. ISSN 2078-2489.

[135] Pieter Gijsbers, Erin LeDell, Sébastien Poirier, Janek Thomas, Bernd Bischl, and Joaquin Vanschoren. An open source automl benchmark. June 2019. 6th ICML Workshop on Automated Machine Learning, AutoML@ICML2019 ; Conference date: 14-06-2019 Through 14-06-2019.