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

Alireza Toghiani Khorasgani
Concordia University, Montreal, Canada
alireza.toghianikhorasgani@mail.concordia.ca

Paria Shirani*
University of Ottawa, Ottawa, Canada
pshirani@uottawa.ca

Suryadipta Majumdar
Concordia University, Montreal, Canada
suryadipta.majumdar@concordia.ca

***Abstract*—Among many other security applications, anomaly detection is one of the biggest users of deep learning methods. This growing popularity is mainly driven by two common beliefs: (i) its ability to manage complicated patterns inside large datasets (given a large amount of data) and (ii) its no need of separate feature engineering (as it is done within the model learning). In this study, we question both of those beliefs and revisit the effectiveness of feature selection and data augmentation in the performance of popular deep-learning based anomaly detection approaches. Additionally, we study the impact of other important factors of any learning based anomaly detection approaches including learning models (both traditional Ml and deep learning), data balancing techniques, hyper parameter tuning, etc. on their performance. From this study, we first report that those common beliefs are not always true - which necessitates a framework that can evaluate the usefulness of features and data for specific use cases (varying the data and need). Then, we propose a new framework that can fill in this gap and assist the data users and anomaly detection tools to perform better by selectively choosing all the configurations (such as, features, models, hyper parameter, balanced data, augmented data). Finally, we demonstrate the effectiveness of our framework using two major IoT datasets.**

***Index Terms*—Deep Learning, Machine Learning, Feature Selection, Data Augmentation, Anomaly Detection**

I. Introduction

Deep Learning (DL) techniques are overly used in anomaly detection (e.g., [5], [38], [41], [43], [44]). This popularity is mainly from an usual fact that they are good at finding complicated patterns without requiring a separate feature selection step. However, obtaining a large-scale training dataset, which is believed to be always essential to get a better performing anomaly detection model, is reported to be one of the biggest challenges [1], [29].

Recent anomaly detection research shows varied approaches to feature selection and data augmentation, with some studies using deep learning without explicit feature selection [30], [41], [44], [47], while others incorporate it [11], [14]. Our work systematically investigates the impact of these approaches on anomaly detection performance. Unlike most existing works that leverage deep learning or traditional machine learning (ML) models in anomaly detection (e.g., [11], [38], [41], [44], [46], [47]), in this work, we intend to study how widely these beliefs are applicable in anomaly detection as follows.

- *Is avoiding feature selection always useful?*: Even though it is obvious that avoiding the feature selection step brings more automation and hence convenience in the anomaly detection process, not necessarily that always helps to obtain a better performing models.
- *Is augmenting data always useful?*: Even though in a suitable case adding more data provides a better model for anomaly detection, that might not always be the case, especially when the augmented data plays a negative role on the model.

In this paper, we consider 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, we first examine the impact of different combinations of feature selections, data balancing, and other factors on various models' performance. Next, we select 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, we build 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 [32] and IoT-23 [15]), we evaluate the effectiveness of the proposed framework in assisting the existing anomaly detection tools.

In summary, the main contributions of this paper are:

- 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 at assisting existing anomaly detection approaches in choosing on features and data. The proposed framework provides several DL/ML models along with dif-

*Part of this work has been done during the postdoctoral fellowship of the author at Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

ferent 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 (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].

II. Background

Feature Selection. Feature selection enhances performance of learning by focusing on important data and removing unnecessary or redundant features. In the following, we provide a brief background on major feature selection techniques.

Filter Methods [9] assess feature relevance independent of prediction models using intrinsic data properties and statistical relationships with the target variable, allowing quick screening of irrelevant inputs before costly model training. Notable filter methods include: (i) Mutual Information (MI) [2] measures dependence between variables using entropy reduction. High MI indicates a feature significantly reduces target variable uncertainty, capturing nonlinear associations. (ii) Trank (T-test Ranking) [35] ranks features using t-statistic between class distributions. Features with significantly different class means considered more informative. (iii) Principal Component Analysis (PCA) [20] converts correlated variables into uncorrelated principal components, reducing dimensionality while preserving key information. (iv) Chi-Square Test (χ^2) [27] assesses the association between categorical features and target classes, with higher values indicating stronger dependence. (v) SelectKBest (SKB) [13] reduces dimensionality by retaining the top k highest scoring features. It ranks features based on ANOVA F-values, selecting those with the highest classification power and removing redundant and noisy features.

Wrapper Methods [22] evaluate feature subsets based on their combined performance within a specific machine learning model [22]. This model-dependent approach often yields superior feature selection and accuracy compared to filter methods but at a higher computational cost [34]. Notable wrapper methods include: (i) Random Forest (RF) [3] ranks features by their average impurity reduction across the ensemble's decision trees. (ii) Particle Swarm Optimization (PSO) [21] optimizes feature subsets by iteratively updating positions of particle swarms representing candidate solutions.

Embedded Methods [6] integrate filter and wrapper approaches, efficiently selecting features during model training while considering interactions. They find model-specific subsets without exhaustive searches, promoting generalization and stability [6], for example: (i) Lasso Regularization (L1) [39] introduces a penalty on the absolute magnitudes of the model coefficients. This method reduces certain coefficients to zero, inherently integrating feature selection within model training. (ii) Ridge Regularization (L2) [18] penalizes the square of model coefficients, reducing their magnitude but not to zero. It addresses multicollinearity by evenly distributing feature importance, helping reduce overfitting.

Data Complexity. Data complexity encompasses various intrinsic dataset characteristics that challenge machine learning algorithms beyond simple class distribution issues [17]. These include factors such as class ambiguity, data sparsity, high dimensionality, and complex decision boundaries. Various metrics have been developed to evaluate these aspects of data complexity, including: (i) Entropy of Class Proportions (C1) measures dataset imbalance through class proportion entropy. Lower values signify balanced distributions, indicating simpler problems [28]. (ii) Maximum Fisher's Discriminant Ratio (F1) measures the overlap between feature values across classes, with higher values indicating more complexity. It computes the ratio of inter-class to intra-class scatter for each feature [12]. (iii) Misclassification Complexity Measure (CM) quantifies complexity around instances prone to misclassification, identifying areas for model refinement to enhance prediction accuracy [36]. (iv) Error Rate of Linear Classifier (L2) measures the fraction of instances misclassified by a linear model like SVM, indicating linear inseparability of the data [17].

III. Related Works

Deep Learning-based Anomaly Detection. Recent studies propose various IoT anomaly detection solutions. CMD [47] combines network and hardware data using NN, while Minh et al. [30] use a CNN-based interpretable ensemble system. Wang et al. [41] focus on certifying the robustness of deep learning traffic analysis systems. Other works [8], [11], [14], [19], [38], [44], [46] utilize different anomaly detection models for threat detection on real and simulated networks; some incorporate feature selection [11], [14], [38] and data balancing [8], [14]. However, most studies apply DL techniques without extensively focusing on feature selection or data augmentation, relying on the model's ability to learn relevant features.

Data Augmentation. Data augmentation techniques have been used across various domains, including image classification [29], [31], text classification [45], and intrusion detection [42]. These approaches are predicated on the assumption that increased data volume enhances model performance. In the context of network security, studies like [42] have applied data

[1] https://github.com/alireza12t/AMETIS-framework

augmentation to address class imbalance and improve detection rates in intrusion detection systems. 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. Our study extends these efforts by investigating how data augmentation, when strategically combined with feature selection and hyperparameter tuning, can enhance model generalization and reduce overfitting in cybersecurity anomaly detection, emphasizing a more nuanced approach to data management.

Comparative Study. Table I summarizes the findings of our comparative study on supplementary techniques across different model configurations using real IoT network traffic datasets, such as CICIoT2023 and IoT-23. This evaluation provides insights into the optimal combination of feature selection and data augmentation with anomaly detection models, and distinguishes our study from existing works that solely focus on applying anomaly detection.

IV. Methodology

Overview. Our proposed method performs two major steps towards enhancing the performance of anomaly detection: (i) *finding the best combinations of features and models* and (ii) *evaluating the impact of data augmentation*. Our primary goal is not to compare or produce specific anomaly detection models. Instead, we focus on examining how feature selection, data balancing, and hyperparameter tuning affect anomaly detection performance across various scenarios and datasets. Our framework aims to suggest best combinations of these elements based on dataset characteristics, providing insights into their interactions and impact on system effectiveness.

The *finding the best combinations of features and models* step (elaborated in Section IV-A) aims to reevaluate the common practice of underplaying the role of feature selection especially in deep learning techniques for anomaly detection. To that end, we first prepare numerous scenarios while varying the combinations of data balancing, feature selection, hyperparameter tuning and machine learning models, and then evaluate the effect of each combination on the performance (in both efficiency and accuracy) in anomaly detection to identify the best combination(s). The *evaluating the impact of data augmentation* step (elaborated in Section IV-B) aims to reevaluate the common practice of adding more data to improve the performance of deep learning models for anomaly detection. Thus, we first measure the complexity of various augmented datasets and then conduct the correlation studies between data complexity and their performance in anomaly detection to identify the usefulness of each augmented dataset.

A. Best Features and Models Selection

The overview of finding the best model configurations including features is shown in Figure 1. Specifically, our method performs the following five main feature selection and model configuration steps in different combinations to evaluate their specific impacts on the model: *Data Preparation (D)*, *Feature Selection (F)*, *Hyperparameter Tuning (H)*, *Data Balancing (B)*, and *Anomaly Detection (A)*. To analyze the effects of different modules, we devise ten scenarios ($S0$-$S9$) encompassing various sequences and combinations of modules(always starting with data preparation), systematically evaluating their impact on overall model performance. The scenario, $S0$, includes only the anomaly detection module as a baseline. The scenarios $S1$-$S4$ assess different integration approaches for feature selection and class balancing, while the scenarios $S5$-$S9$ examine the impact of hyperparameter tuning on the previous scenarios.

1) Data Preparation: We utilize two popular IoT datasets: (i) IoT Aposement 23 (IoT-23) and (ii) CICIoT2023. *(i) IoT-23 Dataset* captures network traffic from various IoT devices (e.g., smart locks, Amazon Echo, and Philips HUE lamps) comprising over 760M packets and 325M labeled flows. The dataset features 20 malware captures and three benign traffic captures, encompassing attack types such as Mirai, Torii, and Trojan. *(ii) CICIoT2023 Dataset* simulates 33 distinct attacks on a network of 105 IoT devices, categorizing attacks into seven types: DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai. Both datasets undergo preprocessing to prepare them for anomaly detection models. This preparation involves handling missing values, encoding categorical variables into numeric representations using techniques such as one-hot encoding, and standardizing numerical features to a common scale. These steps ensure the datasets are clean, properly encoded, and scaled for further analysis.

2) Feature Selection: In our study, we employ various feature selection methods discussed in Section II. For *filter methods*, we rank feature importance independently and retain features with scores higher than the average for both ML and DL models, effectively removing noisy and less relevant candidates. The *wrapper methods* return a subset of features. We integrate the *embedded methods* into our DL models. Additionally, we propose to merge the results of several feature selection methods, for which we focus on wrapper and filter techniques, as embedded feature selection methods do not provide a list of chosen features.

Our proposed combined feature selection includes: (i) *All Selected Features* merges all features identified by any selection method, ensuring no potentially significant feature is overlooked. (ii) *Common Features* selects those features chosen by all methods, identifying the core set of important features. (iii) *Majority Voting* leverages collective decisions, selecting features chosen by at least half of the employed methods. (iv) *Separate Wrapper and Filter Common Features* combines commonly selected features from wrapper and filter methods into two distinct subsets. (v) *Wrapper and Filter Majority Voting* applies majority voting separately to wrapper and filter methods, then combines selected subsets. These approaches aim to leverage the strengths of different feature selection tech-

Proposal (Year)	Scope	ML/DL Models									Dataset		Real-time	Data Augmentation	Feature Selection	Data Balancing
		AE	NN	RF	BERT	GB	RNN	CNN	Kmeans Clustering	Statistical Clustering	Simulated	Real				
Hu et al. (2024) [19]	Network Traffic					●						●	●			
Dong et al. (2023) [11]	Network Traffic			●								●	●		●	
Wang et al. (2023) [41]	Network Traffic	●									●	●				
Yuan et al. (2023) [46]	Network Traffic	●							●			●				
Wei et al. (2023) [44]	Network Traffic	●	●				●					●			●	
Fu et al. (2021) [14]	Network Traffic									●		●	●		●	●
Tang et al. (2020) [38]	Web Traffic	●					●					●	●		●	
Nanni et al., (2021) [31]	Detect Malware							●				●		●		
Catak et al., (2021) [4]	Detect Malware						●	●				●		●		●
AMETIS★	Network Traffic	●	●	●	●	●		●				●		●	●	●

TABLE I: Comparison of related works for anomaly detection. Blank space shows feature absence, (●): feature presence, (-): unavailable information, AE: Auto-Encoder, NN: Neural Network, RF: Random Forest, GB: Gradient Boosting.

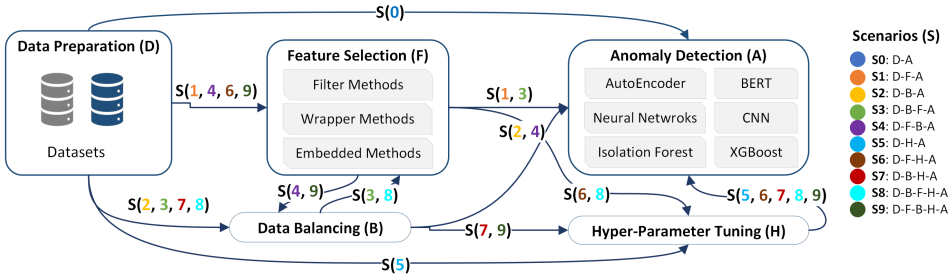

Fig. 1: Different orderings of anomaly detection pipeline modules: feature selection, hyperparameter tuning, data balancing

niques while mitigating their individual biases or limitations.

3) Anomaly Detection Models: Various deep learning and machine learning models enable robust anomaly identification. This study uses popular models like AutoEncoder (AE) [16], BERT [10], Isolation Forest (IF) [26], Neural Network (NN) [16], Convolutional Neural Networks (CNN) [24], and XGBoost [7], each suited for specific data or anomaly detection tasks. Feature selection techniques are applied to enhance performance by retaining relevant inputs, potentially improving accuracy and efficiency.

4) HyperParameter Tuning: In our study, we use a greedy approach for hyperparameter tuning in ML and DL models. We employ *KerasTuner* [2] with its Hyperband algorithm [25] to efficiently navigate complex hyperparameter spaces and identify optimal configurations for our 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.

5) Data Balancing: To address class imbalance in anomaly detection datasets, we employ the Synthetic Minority Oversampling Technique (SMOTE) 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 [43].

6) Methodology Scenarios: This section presents various scenarios where each scenario is a combination of different steps (as shown in Figure 1) to study the impact of feature selection and data balancing on anomaly detection performance using ML and Dl models as follows.

a) Data Flow Baselines (Scenarios S0-S2) 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.

b) Feature Selection vs. Balancing Order (Scenarios S3-S4) inspect the efficacy of applying data balancing techniques either preceding (S3) or succeeding (S4) feature selection, to ascertain the most effective procedural order.

c) Hyperparameter Tuning Integration (Scenarios S5-S7) investigates optimal tuning placement within the pipeline: with only anomaly detection (S5), after feature selection (S6), or following data balancing (S7).

d) End-to-End Integration (Scenarios S8-S9) 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.

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. Our study 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

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

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.

B. Data Augmentation

This section describes the second main part of our method, building upon "Finding the Best Model and Scenario" component. In this phase, we aim to identify which dataset could improve our anomaly detection performance when combined with our original data. We follow a step-by-step process shown in Figure 2. First, we combine datasets and apply the best settings from the previous experiment. Then, we calculate the complexity of the combined data and train our chosen ML or DL model on this data. By analyzing the relationship between data complexity and model performance, we can make suggestions about adding new data to our current dataset.

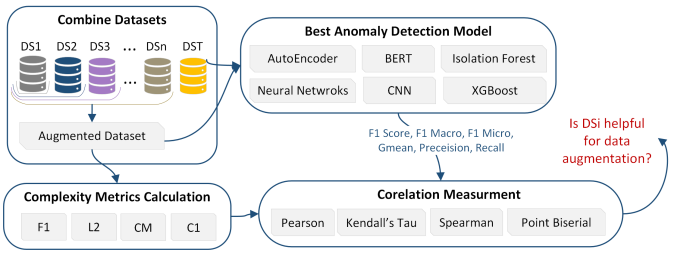

Fig. 2: Process flow of the data augmentation.

1) Data Preparation: We initiate this experiment by combining various datasets into a singular dataset. The objective here is to create a comprehensive pool of data that encapsulates diverse characteristics and patterns. This combined dataset will go through an assessment of data complexity approaches such as Misclassification Complexity Measure (CM), Entropy of Class Proportions (C1), Maximum Fisher's Discriminant Ratio (F1), and Error Rate of Linear Classifier (L2) which is explained in Section II.Upon evaluating data complexity, we engage in the training of the selected ML or DL model from experiments in finding the best model and scenario.

2) Correlation between Data Complexity and Model Metrics: Determining the correlation between the performance indicators of the previously described anomaly detection model and the data complexity of the data is a crucial phase in our process. To clarify this link, we use statistical correlation approaches such as pearson, kendall's tau, spearman's rank, and point biserial and Maximal Information Coefficient (MIC) methods. Several model performance indicators, including *g-mean*, F1-Score, *F1-macro*, *F1-micro*, *recall*, and *precision*, are used to illustrate the correlation results.

In conclusion, our developed methodology offers a framework for customizing anomaly detection systems to the particular requirements of IoT network traffic logs.

V. Evaluation

In this section, we provide a detailed evaluation of our proposed solution. We analyze the effects of data augmentation, feature selection, data imbalance, and hyperparameter tuning on different evaluation metrics.

A. Experimental Setup

Evaluation Metrics. We measure multiple metrics to provide a comprehensive view of model effectiveness specifically for imbalanced data as follows. *Precision* measures true positive rate among predicted positives, and *recall* measures correctly identified actual positives [33]. *F1-Score* balances *precision* and *recall*, [40], *F1-Macro* averages *F1-Scores* across classes, and *F1-Micro* combines overall performance [37]. *G-Mean* assesses balance between model's positive and negative class performance [23].

Dataset. Our study uses IoT-23 [15] and CICIoT2023 [32] datasets. From IoT-23, we select *8-1* (DS_1), *20-1* (DS_2), *3-1* (DS_3), *1-1*, and *42-1*. From CICIoT2023, we include the smallest (DS_4) and largest (DS_5) datasets. The *34-1* dataset from IoT-23 and a CICIoT2023 subset serve as test datasets. Across DS_1-DS_5, attack instances predominate (79% to 99.5%), with DS_2 having the highest benign percentage (21%).

B. Finding the Best Model and Scenario

Analyzing the impact of feature selection, data balancing and hyperparamter 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. CNNs exhibited remarkable improvements, with F1-scores increasing from 16% to 99% using χ^2 and PCA in DS_1, and from 61.33% to 98% using Random Forest (RF) in DS_5. 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. BERT models performed consistently well across all datasets, with F1-scores in DS_1 rising from 94.81% to 98-99% using various methods. 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.

In summary, our 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.

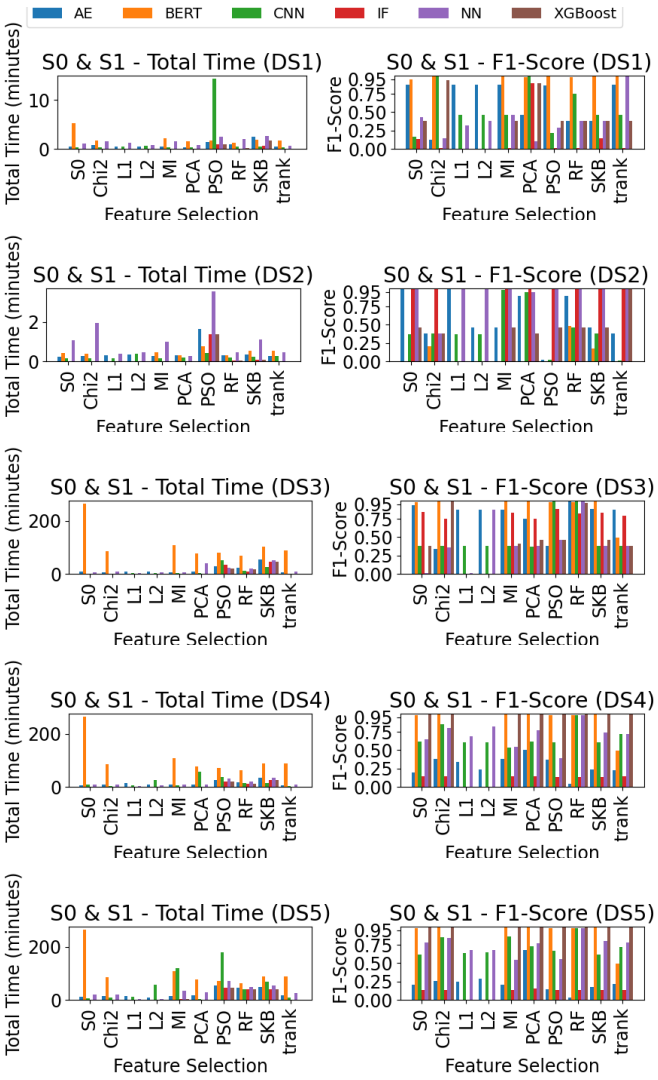

Fig. 3: F1-score vs. total training time across different datasets of DS_1 to DS_5 across scenarios S0 and S1.

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 (Figure 3) 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 χ^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 depicts 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 χ^2 feature selection method.

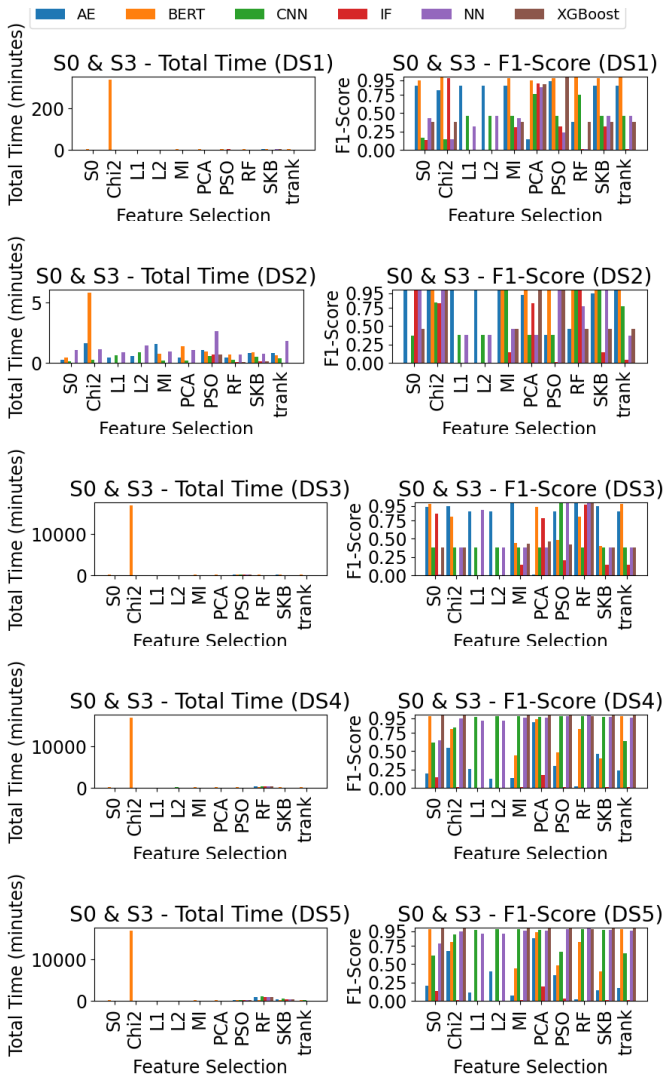

Fig. 4: F1-score vs. total training time across different datasets of DS_1 to DS_5 across scenarios S0 (baseline) and S3 (balancing before feature selection).

2) Impacts of Feature Selection and Data Balancing: The interplay between feature selection and data balancing significantly affects the performance of anomaly detection models. Comparing scenarios S0 (baseline), S3 (balancing before feature selection), and S4 (balancing after feature selection) further insights in our experiment (Figures 4 and 5). 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 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. The order of applying feature selection and data balancing proved crucial; in DS_4, the CNN model improved from 60% to 98% when feature selection (using MI, PSO, SKB, and RF) was applied after balancing. PCA emerged as a consistently effective method across various scenarios and datasets, underscoring its robustness in enhancing DL model performance for anomaly detection tasks in this scenario.

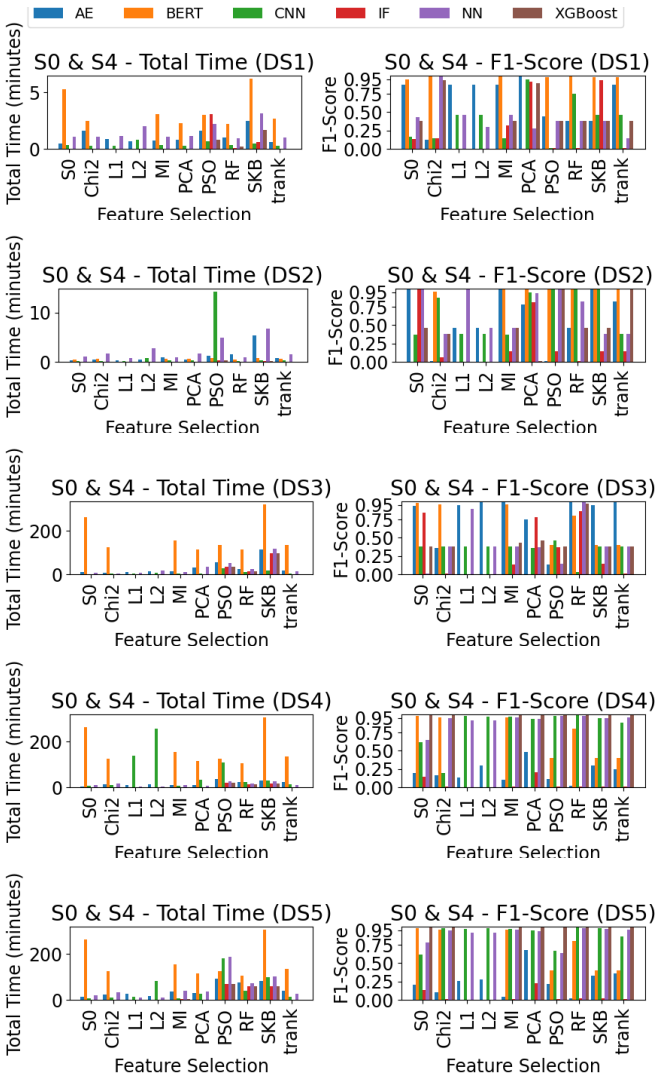

Fig. 5: F1-score vs. total training time across different datasets of DS_1 to DS_5 across scenarios S0 (baseline) and S4 (balancing after feature selection).

3) Impact of Hyperparameters: As shown in Figure 6, the impact of hyperparameters on feature selection is explored by comparing scenarios S5 and S6. This analysis reveals varied impacts on model performance. In DS_1, the CNN model experiences an increase in F1-Score with various feature selection methods in S6 (e.g., from 13.33% to 98.33% with χ^2). 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. In DS_3, CNN model with PSO and RF faces a substantial increase in F1-Score from 42.40% to 99.56% in S6. For ML models, IF with χ^2 feature selection shows improvement, with the F1-score increasing from 38.15% to 93.41% in S6 for DS_1. 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.

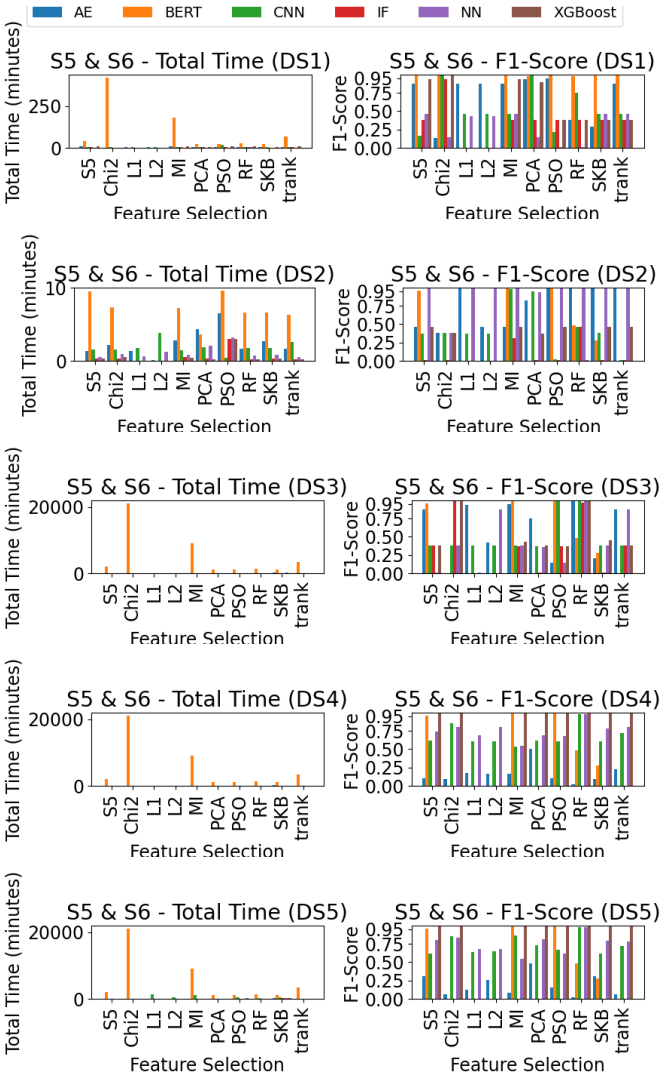

Fig. 6: F1-score vs. total training time across different datasets of DS_1 to DS_5 across scenarios S5 (hyperparameter tuning without feature selection) and S6 (hyperparameter tuning with feature selection).

4) Best Model and Scenario Selection: Our evaluation reveals that optimal strategies for anomaly detection systems vary significantly based on data and model characteristics.

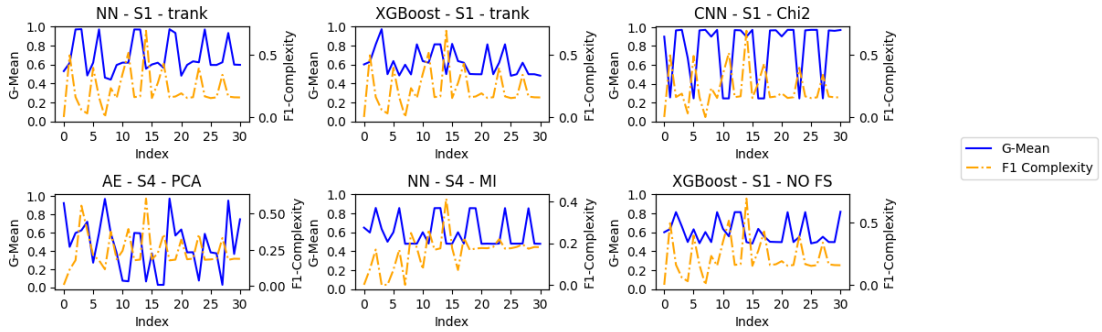

Fig. 7: Correlation between data complexity and G-Mean across various feature selection techniques. The x-axis represents different combinations of datasets ranging from single datasets to all five combined.

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%. 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 II 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 II: Top-5 model and scenarios.

C. 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 7, the G-Mean metric fluctuates as more datasets are incrementally added, indicating that the effects of augmentation can vary significantly. While augmentation helps balance sensitivity and specificity, its benefits are not universal across all scenarios.

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 III: Metrics and complexity measures correlations.

Table III 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. 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 indicates that while data augmentation can enhance model performance, it may also introduce complexity that hinders results under certain conditions. Future research should focus on understanding these dynamics more clearly and identifying the optimal conditions for using data augmentation effectively.

D. Impacts of Combined Feature Selection Methods

The results of our study 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. Computational cost is 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.

VI. Conclusion

While many IoT security works employ anomaly detection, they often overlook the impacts of feature selection and data balancing. Our study shows these elements significantly influence model performance, depending on dataset characteristics and processing methods. Our framework optimizes configurations for specific datasets, potentially enhancing existing tools. Future work will address current limitations by incorporating generative models, conducting real-world testing, and validating across a broader range of IoT-related datasets, aiming to generalize our findings and provide more conclusive metrics for data augmentation decisions.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their valuable comments. The authors also would like to thank Dr. Giulia Fanti for her insightful feedback to this work. Thanks to Cristian Barros Ferreira for CNN deployment. This material is based upon work supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Department of National Defence Canada (DND) under the Discovery Grants RGPIN-2021-04106 and DGDND-2021-04106.

REFERENCES

[1] M. Al Olaimat, D. Lee, Y. Kim, J. Kim, and J. Kim. A learning-based data augmentation for network anomaly detection. pages 1–10, 2020.

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

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

[4] F. O. Catak, J. Ahmed, K. Sahinbas, and Z. H. Khand. Data augmentation based malware detection using convolutional neural networks. *PeerJ Computer Science*, 7:e346, 2021.

[5] R. Chalapathy and S. Chawla. Anomaly detection using deep learning: A survey. *arXiv: Machine Learning*, 2019.

[6] 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.

[7] T. Chen and C. 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.

[8] A. Chohra, P. Shirani, E. B. Karbab, and M. Debbabi. Chameleon: Optimized feature selection using particle swarm optimization and ensemble methods for network anomaly detection. *Computers & Security*, 117:102684, 2022.

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

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Y. Dong, Q. Li, K. Wu, R. Li, D. Zhao, G. Tyson, J. Peng, Y. Jiang, S. Xia, and M. Xu. HorusEye: A realtime IoT malicious traffic detection framework using programmable switches. In *32nd USENIX Security Symposium*. USENIX Association, 2023.

[12] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

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

[14] C. Fu, Q. Li, M. Shen, and K. 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.

[15] S. Garcia, A. Parmisano, and M. J. Erquiaga. IoT-23: A labeled dataset with malicious and benign IoT network traffic, May 2021.

[16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[17] T. B. Ho, B. M. Jain, and R. S. Srivastava. Complexity of classification problems and artificial neural networks. *Neurocomputing*, 43(1-4):219–230, 2002.

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

[19] Z. Hu, H. Hasegawa, Y. Yamaguchi, and H. Shimada. Enhancing detection of malicious traffic through fpga-based frequency transformation and machine learning. *IEEE Access*, 2024.

[20] I. T. Jolliffe and J. Cadima. *Principal Component Analysis*. Springer, 2016.

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

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

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

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018.

[26] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 eighth IEEE international conference on data mining*, pages 413–422. IEEE, 2008.

[27] H. Liu and R. 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.

[28] A. C. Lorena, L. P. Garcia, J. Lehmann, M. C. Souto, and T. K. Ho. How complex is your classification problem? a survey on measuring classification complexity. 52(5):1–34, 2021.

[29] A. Mikołajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. pages 117–122, 2018.

[30] C. Minh, K. Vermeulen, C. Lefebvre, P. Owezarski, and W. Ritchie. An explainable-by-design ensemble learning system to detect unknown network attacks. CNSM, 2023.

[31] L. Nanni, M. Paci, S. Brahnam, and A. Lumini. Comparison of different image data augmentation approaches. *Journal of Imaging*, 7(12), 2021.

[32] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment, 2023.

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

[34] Y. Saeys, I. Inza, and P. Larra naga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[35] M. L. Samuels, J. A. Witmer, and A. A. Schaffner. *Statistics for the Life Sciences*. Pearson, 2012. Chapter on T-test.

[36] M. R. Smith, T. Martinez, and C. Giraud-Carrier. An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256, 2014.

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

[38] R. Tang, Z. Yang, Z. Li, W. Meng, H. Wang, Q. Li, Y. Sun, D. Pei, T. Wei, Y. Xu, and Y. Liu. Zerowall: Detecting zero-day web attacks through encoder-decoder recurrent neural networks. *IEEE Symposium on Security and Privacy*, 2020.

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

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

[41] K. Wang, Z. Wang, D. Han, W. Chen, J. Yang, X. Shi, and X. Yin. Bars: Local robustness certification for deep learning based traffic analysis systems. In *NDSS Symposium*. Internet Society, 2023.

[42] Y. Wang, J. Liu, X. Chang, J. 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.

[43] Y. Wang, Y. Yang, H. Wang, and P. S. Yu. Imbalanced learning for anomaly detection: a comprehensive review. *ACM CSUR*, 55(2):1–37, 2021.

[44] F. Wei, H. Li, Z. Zhao, and H. Hu. XNIDS: Explaining deep learning-based network intrusion detection systems for active intrusion responses. *IEEE S&P*, 2023.

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

[46] Q. Yuan, C. Liu, W. Yu, Y. Zhu, G. Xiong, Y. Wang, and G. Gou. BoAu: Malicious traffic detection with noise labels based on boundary augmentation. *Computers & Security*, 131:103300, 2023.

[47] Z. Zhao, Z. Li, J. Yu, F. Zhang, X. Xie, H. Xu, and B. Chen. CMD: Co-analyzed IoT malware detection and forensics via network and hardware domains. *IEEE Transactions on Mobile Computing*, 22(1):1–14, 2023.