Towards Robust IoT Security: The Impact of Data Quality and Imbalanced Data on AI-Based IDS

Hiba El Balbali, Anas Abou El Kalam
Cadi Ayyad University, National School of Applied Sciences, LaRTID Laboratory, Marrakech, Morocco

***Abstract*—The increased number of connected devices and the rise of Big Data have revolutionized industries and triggered a surge in cyberattacks, making security a top priority. Machine learning and Deep Learning algorithms are crucial in intrusion detection and classification, enabling systems to identify and respond to threats with precision. However, the success of these algorithms is directly related to the quality of the data they process, underscoring the critical importance of robust and well-prepared datasets. Furthermore, despite their potential in detecting and classifying attacks, some algorithms are susceptible to imbalanced datasets, struggling to accurately classify minority classes, while others demonstrate resilience to such challenges. Hence, this study presents a comprehensive analysis of the impact of data quality and imbalanced data on different classification problems, particularly binary, 8-class, and 34-class classification in an intrusion detection context. Our work extensively evaluates six ML and DL algorithms using a novel IoT dataset. Unlike existing research, we use a diverse set of metrics, including accuracy, precision, recall, F1-score, AUC-ROC, and other visual tools, to provide a robust and reliable algorithm performance assessment. This unique analysis underscores the critical importance of addressing data quality and the impact of different balancing techniques on the type of algorithms and type of classification.**

***Keywords*—Machine learning; intrusion detection; internet of things; data quality; big data**

I. Introduction

The proliferation of interconnected devices, often called the Internet of Things (IoT), has led to an unprecedented increase in the volume of data generated and exchanged daily, and has revolutionized how we live and work. These devices, from smart homes to industrial control systems and autonomous vehicles, are increasingly integrated into our daily lives, creating a vast digital ecosystem. While the IoT has brought numerous benefits, it has also introduced new vulnerabilities that cybercriminals are eager to exploit; the surge in connected devices has created a complex attack surface, making it increasingly difficult to identify and mitigate cyber threats. In fact, according to the mid-year update to the 2023 SonicWall Cyber Threat Report, IoT malware globally increased by 37% in the first six months of 2023, resulting in 77.9 million attacks, up from 57 million in the first six months of 2022 [1].

As the number of devices and the complexity of networks grow, so does the sophistication of cyberattacks. These attacks can range from relatively simple exploits, like unauthorized access to personal devices, to large-scale data breaches that compromise sensitive information. More concerning are the cyberattacks that target critical infrastructure, such as power grids, transportation networks, and healthcare systems, where IoT devices play a pivotal role. These attacks, often leveraging vulnerabilities in IoT ecosystems, can lead to severe consequences, including disruptions in critical services, financial losses, and threats to public safety.

Among these, intrusion detection has become a crucial area of research due to the increasing complexity and frequency of cyberattacks. These systems form a critical line of defense against cyber threats by identifying unauthorized access or malicious activities. However, the accuracy of intrusion detection algorithms is highly sensitive to the quality of the data they rely on [2], [3]. Low-quality or imbalanced datasets can lead to misleading results, poor model performance, and a higher risk of false positives or false negatives, which can undermine the detection system's effectiveness.

With the rise of Big Data, the challenge of handling vast and complex datasets has become even more prominent. In the context of intrusion detection, Big Data presents both an opportunity and a challenge. The enormous volume, velocity, and variety of data generated from various sources—such as network traffic logs, user behavior, and IoT device interactions—offer a wealth of information for developing sophisticated and accurate detection mechanisms. This abundance of data enables Machine Learning models to identify subtle patterns and anomalies that could indicate malicious activity, thereby improving the precision and adaptability of intrusion detection systems.

However, leveraging Big Data effectively requires overcoming several challenges. The quality and structure of data are crucial, as cybersecurity datasets are often plagued with challenges such as noise, missing values, and, most notably, class imbalance. Class imbalance is often significant in the field of intrusion detection, where normal network behavior vastly outnumbers instances of malicious activity. In a typical dataset, attacks may represent only a tiny fraction of the total data, leading to a highly skewed distribution that complicates the training of machine learning models. If not addressed properly, this imbalance can cause the models to be skewed toward the majority class, effectively ignoring minority classes that represent actual threats.

Interestingly, not all algorithms are equally affected by data imbalance. As we will discover in this study, some algorithms, like tree-based models, are inherently more robust to class imbalance and can perform well without extensive preprocessing. On the other hand, other algorithms may struggle with imbalanced data unless additional steps are taken to balance the dataset or adjust decision thresholds.

In this study, we present a comprehensive analysis of the impact of class imbalance and data quality on intrusion detection, utilizing a variety of Machine Learning and Deep Learning algorithms, namely Random Forest, XGBoost, Logistic Regression, Deep Neural Networks, K-Nearest Neighbors, and LSTM. Our research aims to explore the critical importance of addressing class imbalance, which is a common challenge in cybersecurity datasets, and to evaluate how different techniques influence the performance of various algorithms. We investigate both binary and multiclass classification scenarios to provide a broader understanding of how algorithm sensitivity to imbalance classes varies depending on the nature of the classification task.

To rigorously assess the impact of class imbalance, we employ a range of rebalancing techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE), SMOTE-Tomek Links, SMOTE with Random Undersampling, and SMOTE-Tomek Links with Random Undersampling. These techniques allow us to examine how oversampling and hybrid methods affect different classifiers. In addition, we utilize a diverse set of evaluation metrics, including precision, recall, F1-score, AUC-ROC, and others, to provide a detailed and nuanced assessment of model performance. By combining these metrics with visualization tools, such as confusion matrices, ROC curves, and PR curves, we aim to deliver a robust comparison and analysis of the effectiveness of each algorithm and technique.

Our study presents several key contributions that distinguish it from previous research in the field:

- The use of a novel dataset with several classes, enabling a more comprehensive evaluation of intrusion detection in complex scenarios where the number of attack types is varied and extensive.
- Hybrid feature selection, combining Chi-squared test and random Forest.
- The evaluation of the impact of class imbalance and data quality on intrusion detection using various Machine Learning algorithms; Random Forest, XGBoost, Logistic Regression, Deep Neural Networks, K-Nearest Neighbors, and LSTM. Our approach provides a broader understanding of how different algorithms are affected by data challenges.
- The creation of several sub-datasets adapted for different classification and balancing scenarios, providing a useful resource for researchers.
- The comparison of multiple balancing methods, including SMOTE, SMOTE-Tomek Links, and combinations with Random Undersampling, to explore their impact across different classifiers in an intrusion detection context.
- The use of a diverse set of evaluation metrics and visual tools, such as precision, recall, F1-score, AUC-ROC, and confusion matrices, ROC curve…, to deliver a detailed analysis of the model's performance. This multidimensional evaluation provides a more complete view compared to traditional existing studies.

Subsequently, this study is organized as follows: Section II will be devoted to discuss some of the related works. In Section III, we will discuss class imbalance and the importance of data quality especially in an Intrusion Detection context. Next, we will present the dataset we used to perform our analysis in Section IV. Section V will be dedicated to present our comprehensive study and workflow. Section VI presents the results, analysis, and findings. Finally, Section VII will conclude our study.

II. Related Work

The authors of [4] conducted a study on the impact of data balancing techniques for SCADA-based intrusion detection systems, utilizing the Morris Power dataset and the CICIDS2017 dataset. They evaluated several balancing methods: Random Sampling, One-Side Selection, Near-Miss, SMOTE, and ADASYN. The authors employed a combination of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks in a binary classification problem. The performance of these techniques was assessed using accuracy, precision, recall, and F1-score metrics. Their findings revealed that the unbalanced data yielded better results for the Morris Power dataset than the balanced versions. Conversely, for the CICIDS2017 dataset, the SMOTE oversampling technique demonstrated better performance. The best accuracy achieved by the model is 99.47% using the SMOTE technique with the CICIDS2017 dataset and 73.63% using the unbalanced Morris Power dataset.

The authors of [5] conducted a comparative analysis of several Deep Learning algorithms, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, for binary classification of network intrusion detection tasks. The UNSW-NB15, KDDCup'99, and NSL-KDD datasets were employed to evaluate the performance of these algorithms under both balanced and imbalanced data conditions. The models were assessed using standard metrics such as accuracy, precision, recall, and F1-score. The results indicate that, in general, balanced datasets yield superior performance compared to imbalanced datasets. Notably, the LSTM model achieved the highest accuracy of 95.4% on the balanced KDDCup'99 dataset.

In [6], the authors presented a study on the influence of resampling techniques on the performance of the ANN model. They compared different methods, namely, random undersampling (RU), random oversampling (RO), random undersampling and random oversampling (RURO), random undersampling with Synthetic Minority Oversampling Technique (RU-SMOTE), and random undersampling with Adaptive Synthetic Sampling Method (RU-ADASYN) using four datasets; KDD99, UNSW-NB15, UNSW-NB17 and UNSW-NB18, and they used macro precision, macro recall and macro F1-score to evaluate the results. From this study, we note that the training time increases with oversampling methods and decreases with undersampling, however, the precision scores were almost unchanged.

In IoT intrusion detection contexts, there is a notable gap in the literature regarding the comparative evaluation of different methods for addressing class imbalance using different classifiers to assess the impact of these techniques on each Machine Learning or Deep Learning algorithm.

So in another context, the authors of [7] compared several sampling techniques, namely, Random Under Sampling, Random Oversampling, and the combination of the SMOTE-NC and RUS as a hybrid resampling technique. They used the Random Forest algorithm for binary classification and the accuracy, precision, recall, F1-score, and ROC-AUC metrics to evaluate each resampling technique. This study showed that sampling techniques improve the model's performance, especially the hybrid method. However, the Random Oversampling method presented an overfitting problem.

The authors [8] performed a comparative study on imbalanced learning techniques for both classification and regression tasks. In the classification part, they employed Random Forest, SVM, and K-NN algorithms to address binary classification. Their analysis compared the performance of models trained on the original dataset and models enhanced with Over-sampling, Under-sampling, SMOTE, and a combination of Over-sampling and Under-sampling techniques. Accuracy was used to evaluate the performance of the algorithms used. The results highlighted that the over-sampling approach, particularly with the Random Forest algorithm, produced the best performance, achieving an accuracy of 98.29%.

The analysis of existing studies that examine the impact of imbalanced data on the performance of machine learning and deep learning algorithms reveals a significant gap in the literature. Very few studies have conducted a comprehensive comparative analysis of various resampling techniques in the context of intrusion detection. Existing works are primarily limited to binary classification, employ a narrow range of algorithms, and rely on a limited set of evaluation metrics, which, while useful, are often insufficient to assess performance fully. Furthermore, the absence of multiclass classification in these studies leaves a critical gap in understanding how resampling techniques perform in more complex scenarios.

Our study aims to address these limitations by comparing multiple resampling techniques for intrusion detection in binary and multiclass classification tasks. This approach will enable us to explore the impact of different resampling methods on each algorithm and classification type. Additionally, we will evaluate a broad range of machine learning and deep learning algorithms using diverse metrics and graphical analysis to provide a more comprehensive performance assessment.

III. Data Quality and Imbalanced Data Challenges

A. Data Quality Concept

Data quality is a critical aspect of any data-driven process, including the degree to which data meets the needs of its intended use. High-quality data drives accurate and reliable insights, fosters trust in analytical outcomes, and supports decision-making processes across various domains. In traditional data systems, ensuring data quality primarily involved managing structured datasets and addressing issues such as missing values, redundancy, and inconsistencies.

The main part of data quality management is data quality assessment, and data quality is generally assessed using several dimensions. A dimension is a set of attributes that indicate a specific element of data quality [9]. The authors [10] described dimension as a quantitative feature of data quality that describes an aspect of data, such as accuracy, precision, consistency, and so on. Dimensions are assessed using metrics, and a metric is a quantifiable instrument that specifies how a dimension is measured. Regarding the number of existing dimensions of data quality, there is no exact agreed-upon count. The authors [9] conducted a comprehensive study that identified 179 distinct dimensions of data quality. In a subsequent analysis focused on the significance of these dimensions, they refined the list to 20 key dimensions, including Completeness, Accuracy, and Consistency, among others.

Over the years, a wide variety of dimensions have been proposed, and the most used data quality dimensions in literature are:

- Completeness: Ensures that all relevant data points are present. Missing values in critical fields, such as timestamps or identifiers, can hinder analyses and lead to misinterpretations. No data should be missing in the dataset.
- Accuracy: Ensures that the data is free from errors and reflects the real-world entities or processes it represents.
- Uniqueness: It guarantees that no duplicates are present in the data.
- Consistency: Emphasizes uniformity within and across datasets, ensuring that data from all systems within an institution is synced and reflects the same information.
- Timeliness: Ensures that data is up-to-date, relevant to the current context, and accessible when it is needed.
- Validity: Ensures that the data aligns with the specific requirements.

The advent of Big Data—characterized by its volume, velocity, variety, veracity, and value—has introduced new complexities to data quality management. Challenges such as handling heterogeneous data sources (structured, semi-structured, and unstructured) like logs, social media, and IoT sensors . . . , ensuring real-time processing, and managing data veracity are amplified [11]. Additionally, the rapidly growing nature of big data makes it difficult to ensure timeliness and relevance, as outdated data can quickly lose its value. Furthermore, the presence of noise, biases, and anomalies across massive datasets necessitates advanced techniques for data cleaning, integration, and preprocessing. Big data quality management, therefore, requires scalable approaches that combine traditional quality metrics with modern techniques such as anomaly detection and automated validation mechanisms to ensure that large-scale data remains reliable and useful.

B. Importance of Data Quality in Intrusion Detection Systems

As we have seen in our previous studies [2], [3], the effectiveness of numerous security mechanisms, tools, and modern machine learning (ML) and deep learning (DL) models is heavily reliant on the quality of the data they process. These systems leverage data at various stages to make critical decisions aimed at ensuring system integrity and protection against cyber threats. For instance, authentication systems analyze user credentials and behavior to decide whether to grant or deny access, while intrusion detection and prevention systems

(IDS/IPS) generate alerts based on patterns and anomalies within network traffic. Similarly, Security Information and Event Management (SIEM) systems aggregate and correlate logs from multiple sources to detect potential threats or unusual activities. Hence, poor data quality can severely undermine their performance, leading to inaccurate threat detection and increased false alarms [12]; the rule "Garbage in garbage out" relates here.

Additionally, ML and DL techniques have further revolutionized security mechanisms by enabling the development of predictive models that can dynamically adapt to evolving threats. High-quality data ensures that both traditional and ML/DL-based security mechanisms can differentiate between normal and malicious activities with greater precision, thereby reducing false positives and negatives. Conversely, poor data quality can lead to weak models, decreased detection accuracy, and unreliable predictions.

Thus, maintaining robust data quality through preprocessing techniques such as cleaning, normalization, and feature extraction is crucial. These measures not only enhance the performance of traditional security tools but also ensure the reliability and adaptability of ML and DL models in complex cyber environments. Ensuring data quality is therefore a cornerstone of building resilient, data-driven security systems capable of addressing the ever-evolving landscape of cyber threats.

C. Imbalanced Data

In many real-world applications, including intrusion detection, data is often imbalanced, meaning that certain classes are significantly underrepresented compared to others. The initial research on imbalanced data originated from binary classification issues; which involves the presence of both majority and minority classes, with a certain imbalance ratio [13]. This imbalance poses a significant challenge for some ML and DL models, as it can lead to biased predictions favoring the majority class while neglecting the minority class. Balancing a dataset simplifies model training by preventing the model from becoming biased toward one class. In other words, the model will no longer favor the majority class simply because it has more data [14]. Consequently, accurate detection of rare but critical events, such as intrusions, becomes difficult.

In intrusion detection systems, imbalanced data is a persistent challenge, as malicious activities, which generally represent the minority class, are significantly less frequent than normal activities. In some cases, this class imbalance can have severe implications for the performance of IDS, as it can lead to models being biased towards the majority class, thus failing to detect rare but critical intrusions. Moreover, it can result in an increased number of false alarms, where benign activities are incorrectly classified as intrusions. However, the imbalanced data issue is still very context-dependent, determined by the type of ML and DL algorithms, dataset features, and the classification task. Some models are significantly affected by this imbalance, leading to biased predictions and poor generalization performance. In contrast, other models can adapt more readily as they exhibit a certain level of robustness to class imbalance.

Several techniques have been developed to address the imbalanced data issue, including SMOTE, SMOTE combined with random undersampling (SMOTE RUS), and SMOTE-Tomek Links combined with random undersampling (SMOTE-TL RUS). Each of these techniques operates with unique mechanisms to balance datasets, enabling a better representation of the minority class during model training.

1) SMOTE (Synthetic Minority Over-sampling Technique) [15]: This method generates synthetic samples for the minority class. For each instance in the minority class, SMOTE selects one or more nearest neighbors and creates new examples by interpolating between these points. The process of generating synthetic samples involves selecting random data from the minority class, calculating the Euclidean distance to its k nearest neighbors, multiplying the difference by a random number between 0 and 1, and then adding the result to the minority class as a synthetic sample [16]. This procedure is repeated until the desired proportion of the minority class is achieved.

For a given minority sample x, a synthetic sample x_{new} is created using the formula:

$$x_{new} = x + \lambda(x_{nearest} - x) \tag{1}$$

where, $x_{nearest}$ is one of the k-nearest neighbors of x and λ is a random number in the range [0, 1].

Fifteen years after its introduction, the SMOTE method has become a foundation for the research community in addressing imbalanced classification challenges. Its innovative approach to generating synthetic samples for the minority class has inspired numerous extensions and adaptations, each building upon its foundation to tackle specific limitations or enhance its applicability [17].

2) SMOTE-TL (Synthetic Minority Over-sampling Technique – Tomek Links): It is a hybrid technique that combines the SMOTE technique with Tomek Links to address class imbalance in datasets.

Tomek Links is a variant of the Condensed Nearest Neighbors undersampling approach invented by [18]. Unlike the CNN method, which randomly selects samples along with their k-nearest neighbors from the majority class to be removed, the Tomek Links method employs a rule to identify and eliminate specific pairs of observations. A pair of observations (a, b) is classified as a Tomek Link if it satisfies the following conditions [16]:

*Observation a's nearest neighbor is b.

*Observation b's nearest neighbor is a.

*Observations a and b belong to different classes, with one observation from the minority class and the other from the majority class.

Mathematically, for two instances x_a from the majority class and x_b from the minority class, a Tomek Link exists if they are the nearest neighbors of each other, and the distance $d(x_a, x_b)$ is minimal.

SMOTE-TL combines the SMOTE capability to generate synthetic data for the minority class and the Tomek Links ability to remove data recognized as Tomek links from the majority class.

The overall process of SMOTE-TL can be stated as follows:

*Apply SMOTE to create synthetic samples for the minority class.

*Identify Tomek Links by checking the nearest neighbor relationships.

*Remove the majority class instances that form Tomek Links with the minority class instances.

3) SMOTE with random under sampling: It is a hybrid resampling technique that combines the SMOTE technique with undersampling. While SMOTE effectively increases the number of minority class samples by generating synthetic examples, it can also increase the risk of overfitting, especially if the majority class is significantly larger. To counter this, undersampling is employed simultaneously, which involves randomly removing instances from the majority class to achieve a more balanced dataset and potentially improve the model's performance.

4) SMOTE-TL with random under sampling: This technique consists in applying SMOTE-TL followed by an additional Random Undersampling step. It is an advanced resampling technique that addresses class imbalance while further refining the dataset. This approach leverages the strengths of both methods to ensure balanced class distributions, remove noisy samples, and reduce the dataset's size, making it computationally efficient for subsequent model training, especially for large datasets.

In this study, we will explore several resampling techniques, namely SMOTE, SMOTE with random undersampling, and SMOTE-TL with random undersampling, and their impacts on the performance of different ML and DL models. Furthermore, we will evaluate how these techniques affect binary and multiclass classification tasks and compare these resampling methods with models trained on imbalanced data to highlight their impact.

IV. Use Case: The CICIOT 2023 Dataset

To conduct our study, we used the CICIoT2023 dataset [19], a recent and extensive IoT attacks dataset. It was created at the Canadian Institute for Cybersecurity, University of New Brunswick. This dataset includes 33 attacks that are classified into seven types of attacks. Table I represents the types of attacks and their description.

The authors extracted over 68 million records and 47 features using an IoT topology regrouping 105 devices.

To optimize computing time, we randomly selected 20% of the original dataset, constituting a subset of 13,584,750 records. Fig. 1 represents the class distribution of our selected data.

This selection was made to facilitate efficient analysis and ensure the feasibility of processing the data within the available computational resources.

V. Workflow and Execution

Our comprehensive study focuses on a multifaceted approach to intrusion detection. It begins with data selection and extraction, followed by a rigorous data cleaning process to prepare the dataset for further analysis. The study then involves creating three distinct datasets: a binary dataset, an 8-class dataset, and a 34-class dataset.

For each dataset, feature engineering is used to identify relevant features and scale the data appropriately. Subsequently, we apply multiple Machine Learning and Deep Learning algorithms to classify attacks under various scenarios of imbalanced data treatment, including using SMOTE, SMOTE-RUS, SMOTE-TL RUS, and unbalanced data. This holistic approach ensures a robust benchmarking and evaluation of the proposed classification techniques across different data representations and balancing strategies.

A. Data Preprocessing

The first phase of our work focused on data preparation, which is a critical step in ensuring the quality and relevance of our analysis.

In this initial phase of our research, we chose the CICIOT 2023 dataset, a novel Internet of Things dataset that aligns well with our study's objectives. The original dataset comprises approximately 68 million records; however, to optimize our computational resources, we randomly extracted a subset of around 13 million records. The extracted data contained approximately 2% benign traffic, compared to the original dataset's 1.6%, ensuring consistency in data representation.

Following, we performed a comprehensive data cleaning process, as illustrated in Fig. 2.

This step was crucial due to the presence of duplicate entries, which we identified and removed. We also standardized the formatting of our columns to ensure uniformity in value representation and corrected data types where necessary; for instance, some columns initially labeled as Float were incorrectly designated as String. Additionally, we conducted a thorough check for missing values to maintain data integrity. To facilitate further analysis, we encoded categorical features using the label encoder technique.

By the end of this phase, we successfully obtained a clean dataset from which we derived three distinct datasets: a binary dataset, an 8-class dataset, and a 34-class dataset, to evaluate our classification models under varying scenarios and complexities.

Following the data preparation phase, we conducted feature engineering, a crucial step in improving the performance of machine learning models by selecting the most relevant features and transforming the data into a more informative format.

As shown in Fig. 3, for each dataset created (binary, 8-class, 34-class), this phase began with feature selection, where we employed a hybrid method that integrates the Chi-Squared test and Random Forest techniques. The Chi-Squared test is a statistical method used to evaluate the independence of categorical variables, helping us identify features that have a

TABLE I. TYPES OF ATTACKS

Type of attack	Attacks	Description
DoS	TCP flood, HTTP flood, SYN flood, UDP flood	An attempt to overload a machine or network, with the aim of weakening its performance or making it completely inaccessible.
DDoS	ACK fragmentation, UDP flood, SlowLoris, ICMP flood, RSTFIN flood, PSHACK flood, HTTP flood, UDP fragmentation, TCP flood, SYN flood, SynonymousIP flood	A DoS attack where multiple machines flood the target.
Brute Force	Dictionary brute force	A hacking method that cracks passwords, encryption keys… using trial-and-error.
Reconnaissance	Ping sweep, OS scan, Vulnerability scan, Port scan, Host discovery	An attack where the actor gathers all the information of his target for exploit.
Spoofing	ARP spoofing, DNS spoofing	An attempt to gain access to a system by hiding the real identity or location.
Mirai	GRE IP flood, GRE Ethernet flood, UDP plain flood	A botnet designed to take over IoT devices and turn them into controlled bots.
Web Based	SQL injection, Command injection, Backdoor malware, Uploading attack, Cross-Site Scripting (XSS), Browser hijacking	A method used by cybercriminals to compromise computer systems, steal data, or cause damage by exploiting vulnerabilities in applications or services accessible via the Internet.

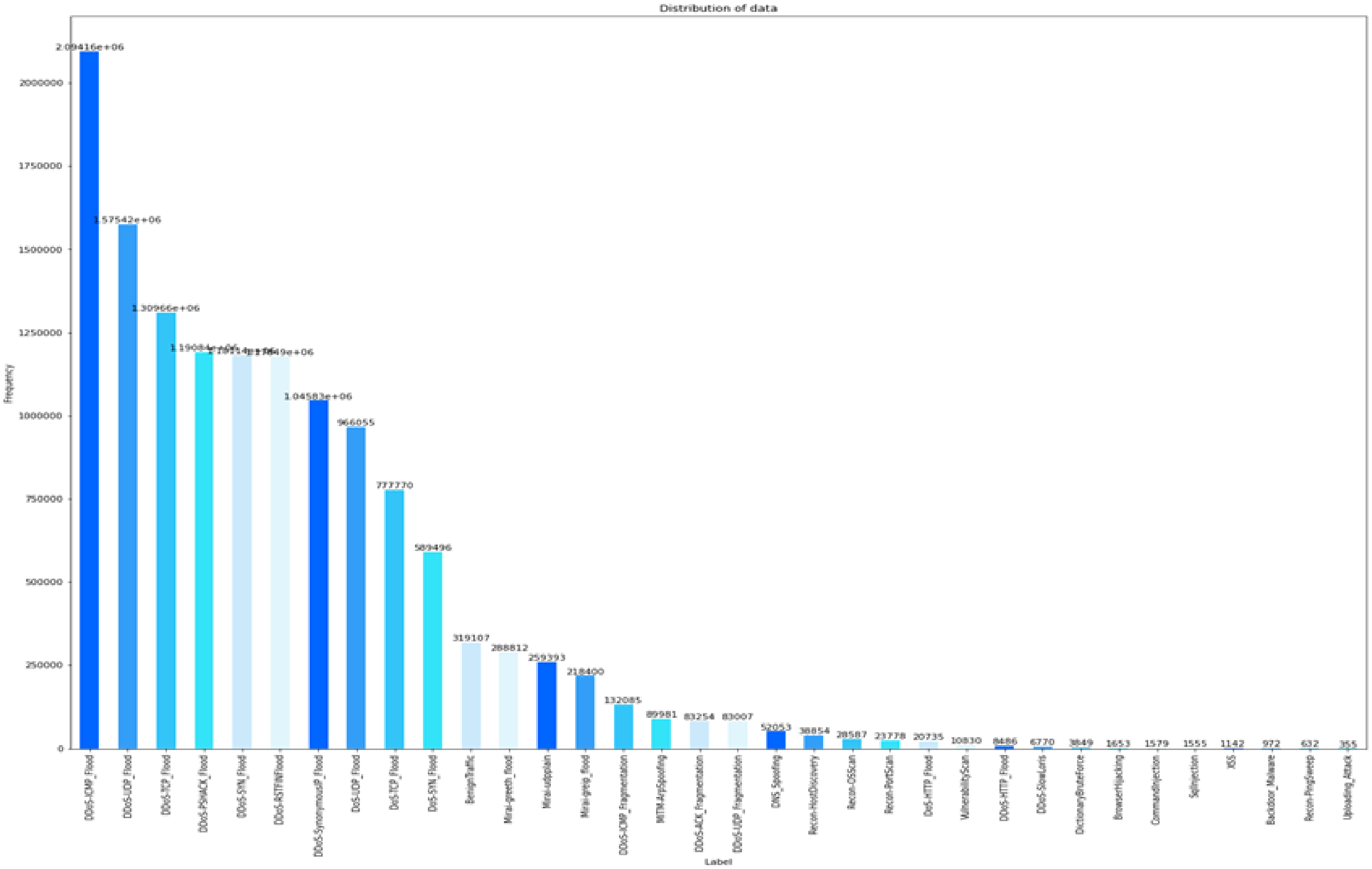

Fig. 1. Class distribution of our selected data.

significant relationship with the target variable. A high Chi-Square score indicates that the independence hypothesis is incorrect, which means the greater the Chi-Square value, the more reliant the feature is on the response and hence suitable for model training [20].

This initial filtering process efficiently narrows the feature set by highlighting the most relevant features based on their statistical significance. Subsequently, we applied the Random Forest algorithm, an ensemble learning method that assesses the importance of features based on their contribution to the model's predictive accuracy. By combining these two techniques, we benefit from the strengths of both: the Chi-Squared test provides a quick and interpretable means of initial selection, while Random Forest offers a robust evaluation of feature importance. This hybrid approach enhances the efficiency of our feature selection process and ensures that we retain the most informative features, ultimately leading to improved model performance.

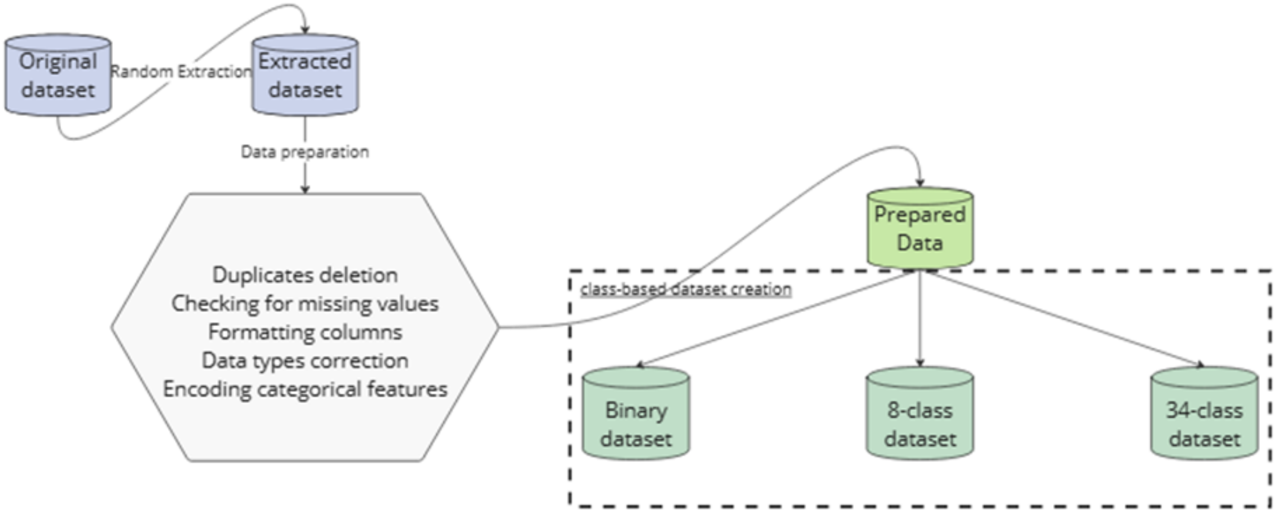

Fig. 2. Our data preparation process.

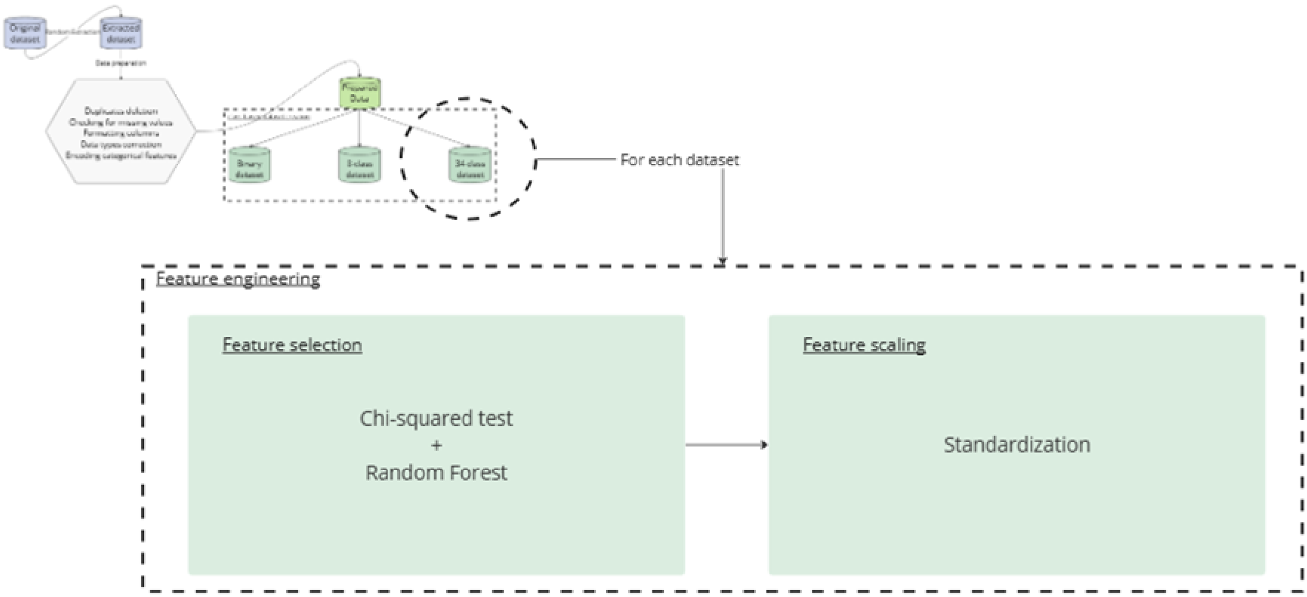

Fig. 3. Our feature engineering process.

Next, we implemented feature scaling using standardization, a crucial process that transforms our features to have a mean of zero and a standard deviation of one. This technique is particularly beneficial in machine learning, as it ensures that all features contribute equally to the distance calculations and optimization algorithms used in various models, regardless of their original scale.

B. Machine Learning and Deep Learning Modeling Under Different Balancing Scenarios

In this phase of our research, we focused on modeling various Machine Learning and Deep Learning algorithms to classify attacks under multiple scenarios: imbalanced data, balanced data using SMOTE, balanced data using SMOTE-RUS, and balanced data using SMOTE-TL RUS, for each dataset (binary, 8-class, and 34-class); which results in 12 scenarios for each algorithm.

As shown in Fig. 4, we chose different algorithms, namely XGBoost, Random Forest, Logistic Regression, Deep Neural Networks, Long Short-Term Memory (LSTM), and K-Nearest Neighbors (K-NN). These algorithms were trained on the respective training datasets, and hyperparameter optimization was conducted using the GridSearch technique.

GridSearch systematically evaluates a given set of hyperparameter combinations to identify the optimal configuration for each algorithm. This approach ensures that the models are fine-tuned to their best possible performance, reducing the risk of underfitting or overfitting. GridSearch offers the advantage of exhaustive search across the hyperparameter space, ensuring no potential configuration is missed.

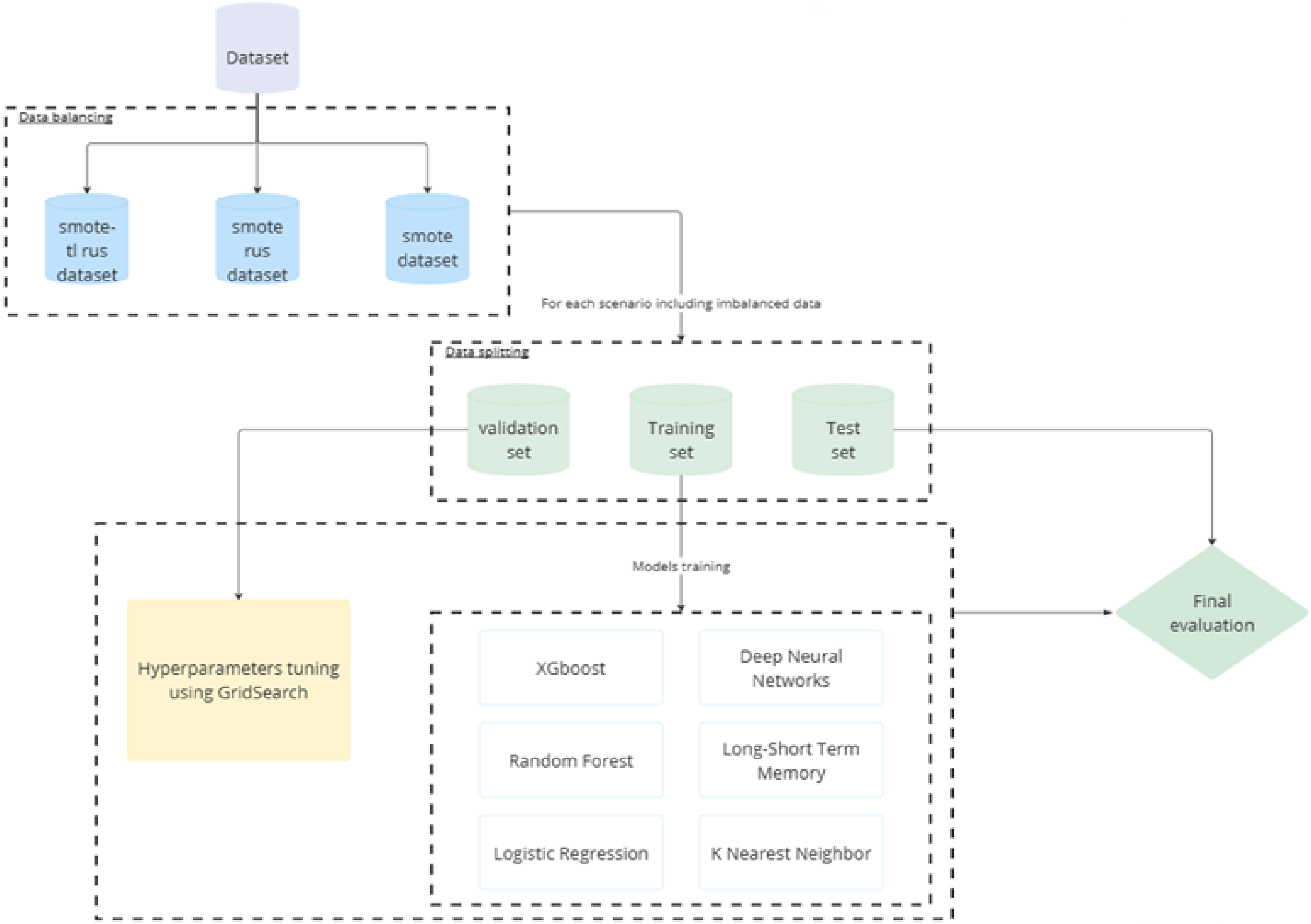

Fig. 4. Our ML and DL modeling workflow.

Once the optimal hyperparameters were identified through validation, the final models were evaluated on the test set. This evaluation step aimed to provide a robust and unbiased assessment of the model's performance.

To ensure a comprehensive and reliable evaluation of our models, we used various performance metrics, namely accuracy, precision, recall, F1-score, AUC-ROC, confusion matrix, PR curve, and ROC curve. These metrics were chosen to capture different aspects of model performance, such as overall accuracy, ability to minimize false positives, sensitivity to detecting true positives, and the balance between precision and recall. Furthermore, the AUC-ROC, PR curve, and confusion matrix provide thorough insights into the model's behavior, offering a deeper understanding of each classifier's strengths and weaknesses.

By leveraging diverse balancing techniques, a wide range of classification algorithms, and multiple datasets, our study provides a detailed and thorough evaluation of the models, ensuring that the results are reliable and generalizable across different attack classification scenarios. This comprehensive approach represents a significant improvement over existing studies that rely on fewer metrics or scenarios.

VI. Results and Discussion

As we mentioned before, we present and analyze the performance of several Machine Learning and Deep Learning algorithms for intrusion detection using a binary dataset, an 8-class dataset, and a 34-class dataset under different balancing scenarios, namely imbalanced data, balanced with SMOTE, and balanced with SMOTE-RUS and SMOTE-TL RUS.

Our experiment was conducted on a Windows 10 Operating System with 16 GB RAM and an Intel(R) Core(TM) i7-7600U CPU @ 2.80 GHz, 2904 MHz processor.

As shown in Table II, for the binary dataset, the results reveal notable differences across balancing strategies.

XGBoost achieved consistent results across all metrics, particularly with the original imbalanced dataset, where it performs slightly better in accuracy, precision, recall, and F1-score. This suggests that XGBoost handles imbalanced data effectively. On the other hand, K-NN demonstrates an excellent performance, especially with the SMOTE-TL RUS

TABLE II. Performance Metrics of Our Models for Binary Classification Across Different Balancing Scenarios

		Binary classification			
		SMOTE-TL RUS	**SMOTE RUS**	**SMOTE**	**Imbalanced data**
XGBoost	**accuracy**	98.78	98.20	98.68	**99.28**
	precision	98.85	99.07	98.96	99.57
	recall	98.71	97.33	98.38	99.69
	f1-score	98.78	98.19	98.67	99.63
	AUC ROC	98.78	98.20	98.67	90.99
KNN	**accuracy**	**99.35**	99.07	99.27	99.32
	precision	99.78	99.77	99.86	99.75
	recall	98.92	98.36	98.68	99.55
	f1-score	99.35	99.06	99.27	99.65
	AUC ROC	99.64	99.3	99.61	99.44
Logistic Regression	**accuracy**	95.75	95.44	95.54	**98.54**
	precision	93.61	93.48	93.60	99.26
	recall	98.21	97.71	97.78	99.25
	f1-score	95.86	95.54	95.64	99.25
	AUC ROC	98.61	98.48	98.59	97.26
Random Forest	**accuracy**	95.16	95.60	98.45	**99**
	precision	91.77	92.94	99.66	99.26
	recall	99.23	98.72	97.24	99.70
	f1-score	95.35	95.74	98.43	99.49
	AUC ROC	99.71	99.45	99.68	99.77
LSTM	**accuracy**	**98.78**	97.75	98.66	97.66
	precision	99.01	95.55	99.26	97.66
	recall	98.55	97.96	98.04	100
	f1-score	98.78	97.76	98.65	98.82
	AUC ROC	93.83	97.91	98.76	79.74
DNN	**accuracy**	98.75	98.76	**99.24**	99.22
	precision	98.66	98.75	99.70	99.65
	recall	98.84	98.76	98.78	99.55
	f1-score	98.75	98.75	99.24	99.60
	AUC ROC	99.81	99.82	99.90	99.79

and SMOTE scenarios, highlighting high precision and recall values. It also maintains strong performance on the imbalanced dataset, indicating that K-NN can benefit from both balanced and imbalanced cases.

In addition, Logistic Regression shows moderate performance compared to other models, with significant improvements in the imbalanced dataset. Random Forest shows considerable strength in precision and recall, particularly on the imbalanced dataset and SMOTE scenarios. Its ability to achieve high AUC-ROC scores across all scenarios demonstrates its robustness and reliability for binary classification tasks.

Regarding Deep Learning models, LSTM demonstrates robust performance across most metrics, with high accuracy, precision, recall, and F1-score in all balancing scenarios. However, the AUC-ROC value varies significantly, being lower with imbalanced data, indicating that the model's ability to distinguish between classes is less effective without balancing techniques. SMOTE-TL RUS and SMOTE show relatively consistent and strong results, suggesting these techniques enhance the model's discrimination ability while maintaining other performance metrics. Furthermore, for DNN, the performance is consistently strong across all scenarios, with minimal differences in metrics. The DNN model achieves its highest AUC-ROC with SMOTE, reflecting its remarkable ability to differentiate between classes with imbalanced data. Additionally, its precision and F1-score are notably high, indicating effective handling of class imbalance.

Table III shows the performance of our models across different balancing techniques on an 8-class dataset.

As we can see, XGBoost performs best with imbalanced data, achieving the highest accuracy and AUC ROC. However, its recall and F1-score are low, indicating poor performance with minority classes. Furthermore, K-NN exhibits strong performance across all techniques, with particularly high accuracy and AUC ROC on imbalanced data. Recall and F1-score are low however, SMOTE improves its recall and F1-score, highlighting its suitability for balanced datasets.

In addition, Logistic Regression's performance is relatively stable, with moderate accuracy and AUC ROC. Balancing techniques improve recall and F1-score slightly, but it is clear that the model struggles to capture complex relationships in the data. Random Forest performs poorly on imbalanced data,

TABLE III. Performance Metrics of Our Models for 8-Class Classification Across Different Balancing Scenarios

		8-class classification			
		SMOTE RUS	**SMOTE-TL RUS**	**SMOTE**	**Imbalanced data**
XGBoost	**accuracy**	51.46	59.59	44.18	**81.22**
	precision	56.25	61.42	69.52	76.59
	recall	50.46	58.75	44.15	54.87
	f1-score	49.16	55.05	38.37	52.52
	AUC ROC	88.43	92.3	91.93	96.64
KNN	**accuracy**	64.51	70.59	78.92	**88.07**
	precision	65.62	70.48	79.66	64.82
	recall	64.53	69.10	78.93	57.13
	f1-score	64.55	69.06	78.32	58.56
	AUC ROC	92.20	92.47	96.93	92.54
Logistic Regression	**accuracy**	55.20	60.50	59.12	**80.55**
	precision	57.27	61.22	61.41	63.03
	recall	55.58	58.23	59.11	41.15
	f1-score	53.15	57.32	58.06	42.81
	AUC ROC	88.73	89.68	90.56	94
Random Forest	**accuracy**	49.1	63.11	58.04	**73.30**
	precision	55.08	68.53	76.70	77.16
	recall	48.45	61.82	58.06	56.78
	f1-score	46.55	61.62	57.25	57.57
	AUC ROC	89.39	92.42	95.32	92.69
LSTM	**accuracy**	51.46	58.18	59.61	**78.47**
	precision	69.72	67.29	79.83	27.30
	recall	52.42	57.12	59.59	32.35
	f1-score	47.35	56	59.21	27.72
	AUC ROC	88.04	89.76	89.29	86.02
DNN	**accuracy**	55.83	61.18	73.22	**88.13**
	precision	60.55	65.55	75.96	72.96
	recall	56.30	58.88	73.22	57.3
	f1-score	54.22	58.08	73.03	59.73
	AUC ROC	89.54	90.06	96.15	97.52

with accuracy and F1-score trailing other models. Smote-TL RUS significantly boosts RF's recall and F1-score, indicating its reliance on balanced datasets to handle minority classes effectively.

Concerning Deep Learning models, LSTM shows inconsistent performance, with high precision on SMOTE but very low recall on imbalanced data. Balancing techniques like SMOTE-TL RUS and SMOTE enhance its recall and F1-score. Moreover, DNN performs well overall, with the best accuracy and F1-score on imbalanced data. Its AUC ROC is also the highest among all models. SMOTE significantly enhances its recall and F1-score, highlighting its ability to leverage balanced data effectively.

Table IV shows the performances of our models across different balancing techniques on a 34-class dataset.

As observed, XGBoost performs quite well with imbalanced data, achieving moderate accuracy and a high AUC ROC, indicating its ability to separate classes probabilistically. However, its low recall and F1-score reveal poor performance on minority classes. Balancing techniques such as SMOTE and SMOTE RUS provide slight improvements in recall and F1-score but result in lower overall accuracy. Furthermore, K-NN demonstrates robust performance, particularly with Sample SMOTE, achieving high accuracy and AUC ROC. While its recall and F1-score are relatively low on imbalanced data, balancing techniques like Sample SMOTE significantly boost these metrics, showing its adaptability to rebalanced datasets.

Moreover, Logistic Regression maintains moderate performance across all scenarios, with high AUC ROC values. However, its accuracy and F1-score are low, reflecting its inability to model complex relationships. SMOTE slightly improves recall and F1-score, but the model struggles to achieve competitive performance compared to others. Random Forest shows high accuracy and precision on imbalanced data but low recall and F1-score, indicating challenges in handling minority classes. SMOTE noticeably improves RF's recall and F1-score, demonstrating its reliance on balanced datasets to perform effectively.

Among Deep Learning models, LSTM shows inconsistent performance. It achieves a high AUC ROC on imbalanced data but has low recall and F1-score. SMOTE enhances its recall and F1-score, indicating its potential with balanced

TABLE IV. Performance Metrics of Our Models for 34-Class Classification Across Different Balancing Acenarios

		34-class classification			
		Imbalanced data	**SMOTE-TL RUS**	**SMOTE RUS**	**SMOTE**
XGBoost	**accuracy**	**76.28**	39.08	40.18	40.44
	precision	56.55	35.65	41.06	44.55
	recall	53.63	35.86	40.82	40.46
	f1-score	49.64	30.83	35.80	38.46
	AUC ROC	95.01	90.73	92.38	69.83
KNN	**accuracy**	**79,03**	50.76	46.72	74,6
	precision	58,78	46.43	47.51	74,72
	recall	48,89	44.50	47.15	74,62
	f1-score	48,78	39.65	45.12	73,48
	AUC ROC	88,14	85.06	84.31	97,77
Logistic Regression	**accuracy**	**76.03**	46.27	42.81	51.86
	precision	44.94	43.04	43.10	51.40
	recall	36.50	44.17	45.86	51.82
	f1-score	35.37	38.19	39.76	48.97
	AUC ROC	97.83	92.91	92.64	94.67
Random Forest	**accuracy**	**79.93**	30.35	30.46	66.54
	precision	64.34	23.72	26.35	72.97
	recall	51.74	29.90	31.95	66.54
	f1-score	52.18	22.06	24.12	64.52
	AUC ROC	95.55	86.83	90.99	98.39
LSTM	**accuracy**	**80.63**	42.04	35.94	55.75
	precision	38.07	31.12	31.88	59.62
	recall	35.33	38.24	40.57	55.69
	f1-score	33.77	30.16	30.34	51.67
	AUC ROC	98.07	92.51	92.40	96.11
DNN	**accuracy**	**83.14**	36.81	34.68	63.55
	precision	50.35	24.52	28.29	67.12
	recall	45.99	31.55	37.93	63.49
	f1-score	45.46	22.82	29.02	61.49
	AUC ROC	98.98	89.47	90.31	97.48

datasets. Similarly, DNN achieves the best overall performance on imbalanced data, with the highest accuracy and AUC ROC. While its recall and F1-score are initially low, SMOTE and Smote-TL RUS significantly enhance these metrics.

Due to space limitations, we only show ROC curves, PR curves, and confusion matrices of the best models across all balancing scenarios on each dataset.

Fig. 5, Fig. 6, and Fig. 7 show the ROC curve of the best model in each dataset.

Fig. 8, Fig. 9, and Fig. 10 show the PR curve of the best model in each dataset.

Fig. 11, Fig. 12, and Fig. 13 show the confusion matrices of the best model in each dataset.

As we can deduce, our curves and matrices further confirm and support the reported scores and the accompanying discussion. They visually illustrate the variations in performance across models and balancing techniques.

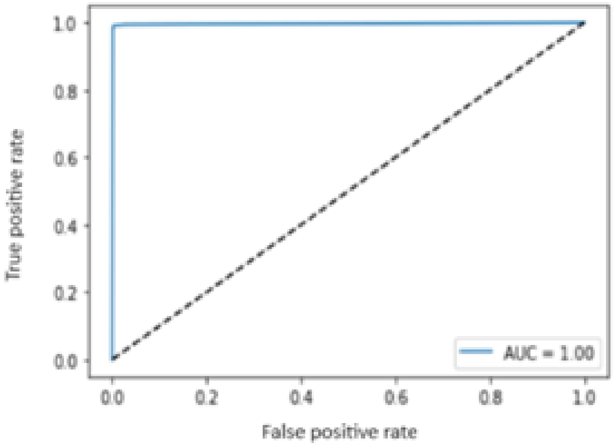

Fig. 5. ROC Curve of the DNN model using imbalanced data and the binary dataset.

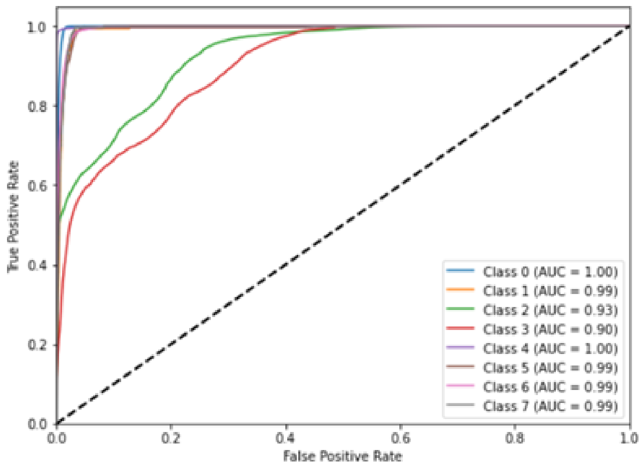

Fig. 6. ROC Curve of the DNN model using imbalanced data and the 8-class dataset.

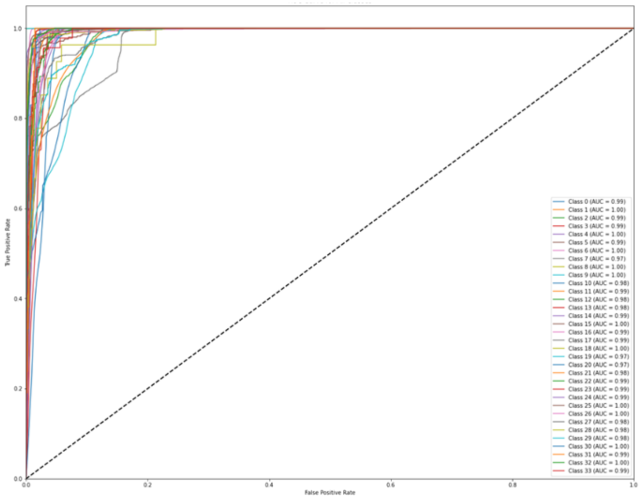

Fig. 7. ROC Curve of the DNN model using imbalanced data and the 34-class dataset.

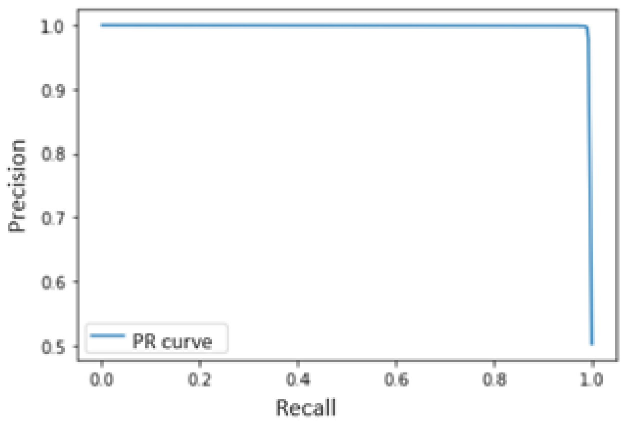

Fig. 8. PR Curve of the DNN model using imbalanced data and the binary dataset.

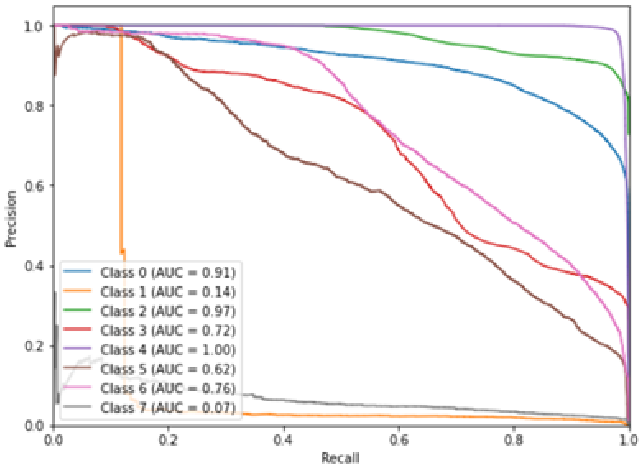

Fig. 9. PR Curve of the DNN model using imbalanced data and the 8-class dataset.

VII. Conclusion

In this study, we conducted a comprehensive analysis of the impact of data quality and class imbalance on classification problems, focusing on binary, 8-class, and 34-class classification in the context of intrusion detection. Using a novel IoT dataset, we thoroughly evaluated the performance of various Machine Learning and Deep Learning algorithms in different balancing scenarios and with different datasets. Our work goes beyond traditional evaluations by incorporating a diverse set of performance metrics; accuracy, precision, recall, F1-score, AUC-ROC, and visual tools such as confusion matrices, PR curves, and ROC curves. This approach ensures a robust and reliable assessment of algorithmic performance.

Additionally, our study demonstrates how results vary depending on each balancing technique and classification type, with certain models excelling in binary tasks but showing limitations in multiclass settings, while others are susceptible to imbalanced data and handle it differently. Another key outcome of this research is the creation of multiple sub-datasets tailored for different classification and balancing scenarios, offering a valuable resource for researchers seeking to extend this analysis or apply these datasets for their research.

In our future work, we intend to investigate the strategic role of data quality in reinforcing the resilience of Intrusion Detection Systems against sophisticated adversarial threats. Moreover, our findings also emphasize the trade-off between the performance of our models and computational cost, as these robust models often require significantly greater computational resources. Finally, extending this approach to other critical domains such as automotive or healthcare systems could provide further insights into the generalization of our findings.

References

[1] A. Marton, "IoT malware attacks up by 37% in the first half of 2023," iotac.eu, 2023. https://iotac.eu/iot-malware-attacks-up-by-37-in-the-first-half-of-2023/#: :text=According to the mid-year,first six months of 2022. (accessed Dec. 14, 2024).

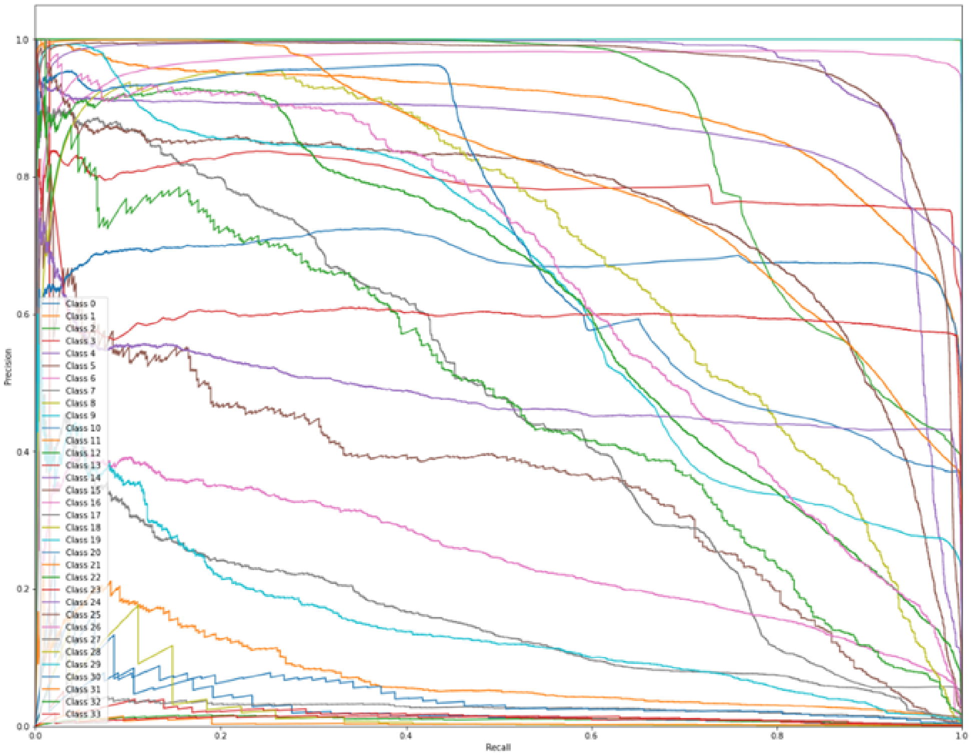

Fig. 10. PR Curve of the DNN model using imbalanced data and the 34-class dataset.

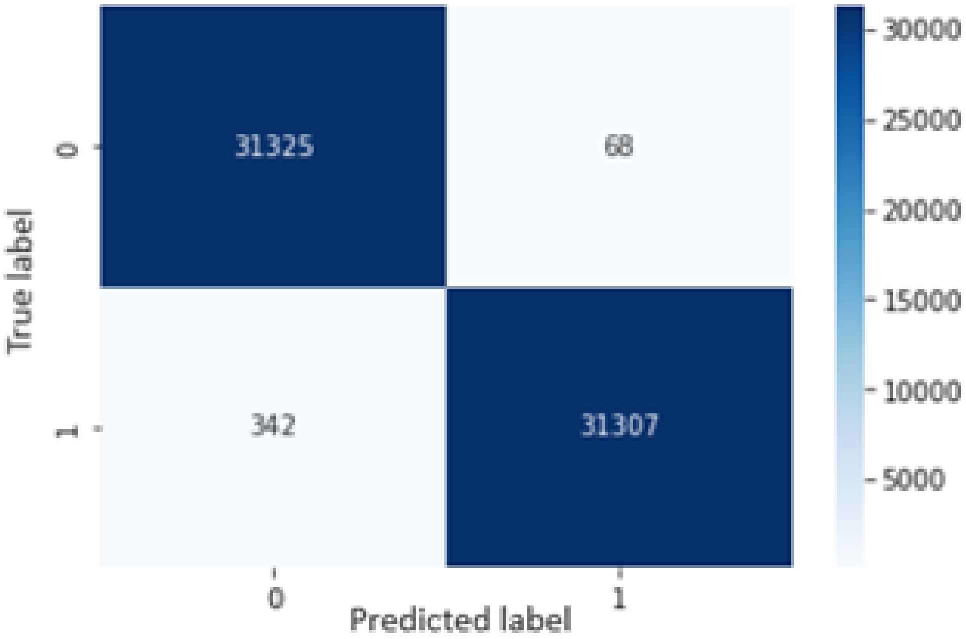

Fig. 11. Confusion matrix of the DNN model using imbalanced data and the binary dataset.

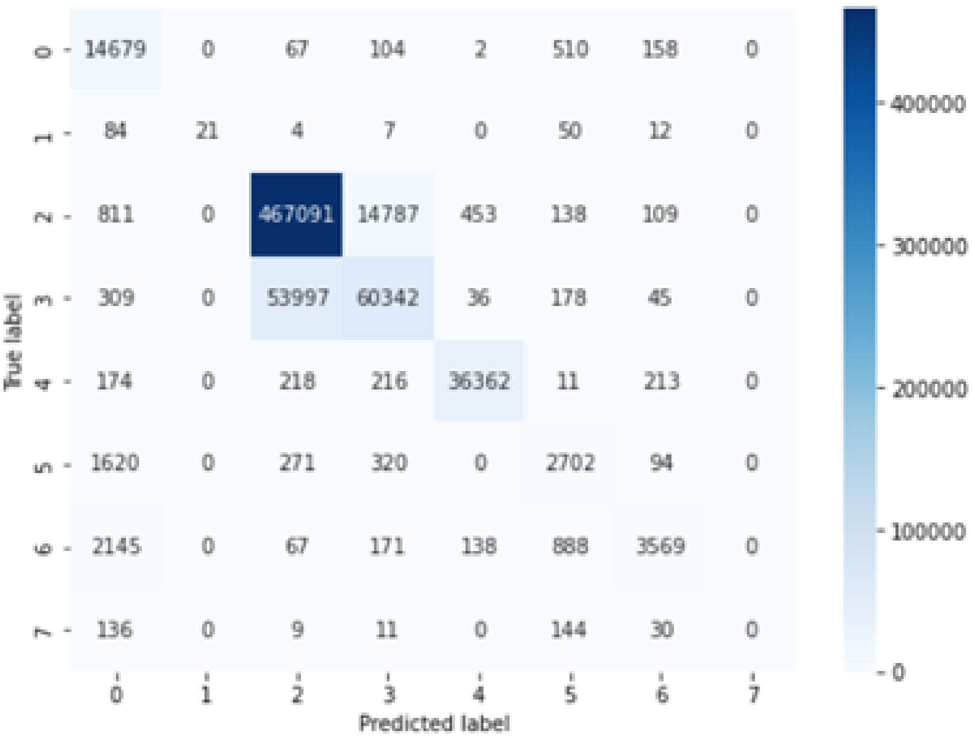

Fig. 12. Confusion matrix of the DNN model using imbalanced data and the 8-class dataset.

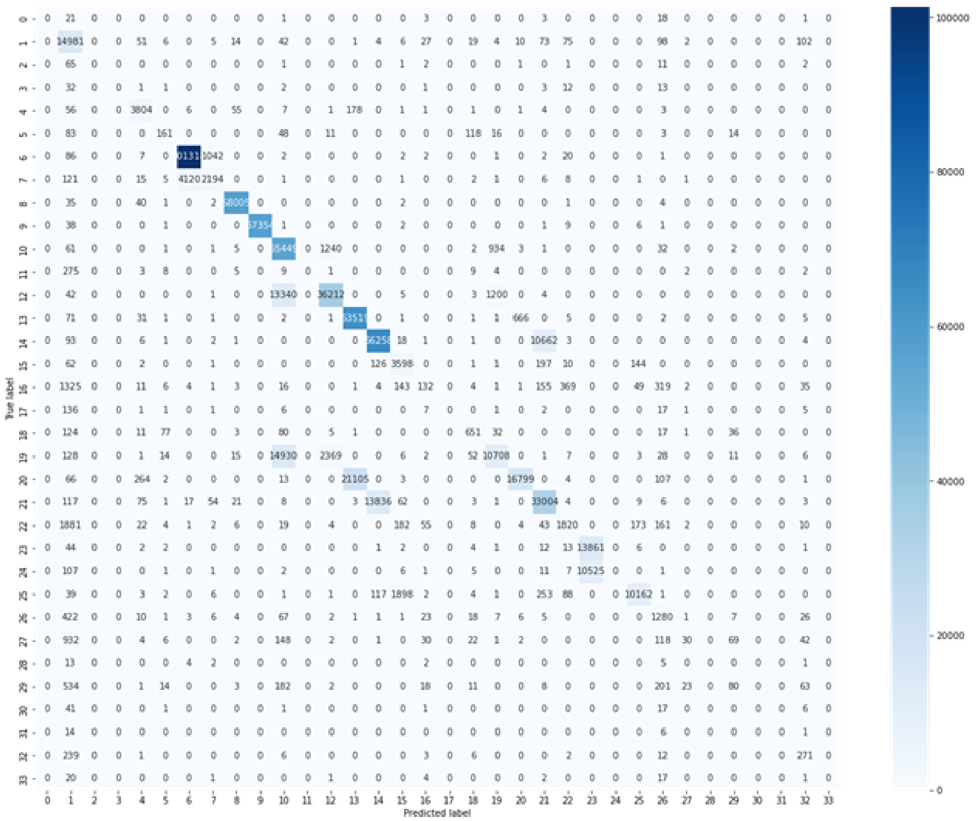

Fig. 13. Confusion matrix of the DNN model using imbalanced data and the 34-class dataset.

[2] H. El Balbali and A. Abou El Kalam, "AI-Driven Big Data Quality Improvement for Efficient Threat Detection in Agricultural IoT Systems," 2023. doi: https://doi.org/10.1007/978-3-031-54318-0_5.

[3] H. El Balbali, A. Abou El Kalam, and M. Talha, "Big Data Between Quality and Security," pp. 1315–1326, 2023, doi: 10.1007/978-3-031-27409-1_120.

[4] A. Balla, M. H. Habaebi, E. A. A. Elsheikh, M. R. Islam, and F. M. Suliman, "The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems," Sensors, vol. 23, no. 2, 2023, doi: 10.3390/s23020758.

[5] A. Meliboev, J. Alikhanov, and W. Kim, "Performance Evaluation of Deep Learning Based Network Intrusion Detection System across Multiple Balanced and Imbalanced Datasets," Electron., vol. 11, no. 4, 2022, doi: 10.3390/electronics11040515.

[6] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," J. Big Data, vol. 8, no. 1, 2021, doi: 10.1186/s40537-020-00390-x.

[7] T. Wongvorachan, S. He, and O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," Inf., vol. 14, no. 1, 2023, doi: 10.3390/info14010054.

[8] S. Rezvani and X. Wang, "A broad review on class imbalance learning techniques," Appl. Soft Comput., vol. 143, 2023, doi: 10.1016/j.asoc.2023.110415.

[9] R. Y. Wang and D. M. Stong, "Beyond accuracy: What data quality means to data consumers," J. Manag. Inf. Syst., vol. 12, no. 4, pp. 5–34, 1996, doi: 10.1080/07421222.1996.11518099.

[10] M. Talha, A. Abou El Kalam, and N. Elmarzouqi, "Big data: Trade-off between data quality and data security," Procedia Comput. Sci., vol. 151, pp. 916–922, 2019, doi: 10.1016/j.procs.2019.04.127.

[11] M. Talha, N. El Marzouqi, and A. Abou El Kalam, "Quality and Security in Big Data: Challenges as opportunities to build a powerful wrap-up solution," J. Ubiquitous Syst. Pervasive Networks, vol. 12, no. 1, pp. 09–15, 2019, doi: 10.5383/juspn.12.01.002.

[12] N. Tran, H. Chen, J. Bhuyan, and J. Ding, "Data Curation and Quality Evaluation for Machine Learning-Based Cyber Intrusion Detection," IEEE Access, vol. 10, pp. 121900–121923, 2022, doi: 10.1109/ACCESS.2022.3211313.

[13] D. Dablain, B. Krawczyk, and N. V. Chawla, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data," IEEE Trans. Neural Networks Learn. Syst., vol. 34, no. 9, pp. 6390–6404, 2023, doi: 10.1109/TNNLS.2021.3136503.

[14] N. Buhl, "Introduction to Balanced and Imbalanced Datasets in Machine Learning," encord, 2022. https://encord.com/blog/an-introduction-to-balanced-and-imbalanced-datasets-in-machine-learning/ (accessed Dec. 11, 2024).

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[16] R. A. A. Viadinugroho, "Imbalanced Classification in Python: SMOTE-Tomek Links Method," Towards Data Science, 2021. https://towardsdatascience.com/imbalanced-classification-in-python-smote-tomek-links-method-6e48dfe69bbc (accessed Dec. 24, 2024).

[17] A. Fernández, S. García, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," J. Artif. Intell. Res., vol. 61, pp. 863–905, 2018, doi: 10.1613/jair.1.11192.

[18] I. Tomek, "A generalization of the k-NN rule," IEEE Trans. Syst. Man. Cybern., vol. 6, no. 2, pp. 121–126, 1976, doi: 10.1109/TSMC.1976.5409182.

[19] 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," Sensors, vol. 23, no. 13, p. 5941, 2023, doi: 10.3390/s23135941.

[20] sampath kumar Gajawada, "Chi-Square Test for Feature Selection in Machine learning," Towards Data Science, 2019. https://towardsdatascience.com/chi-square-test-for-feature-selection-in-machine-learning-206b1f0b8223