eng

MDPI

Article

Balancing the Scale: Data Augmentation Techniques for Improved Supervised Learning in Cyberattack Detection

Kateryna Medvedieva, Tommaso Tosi, Enrico Barbierato and Alice Gatti *

Department of Mathematics and Physics, Catholic University of the Sacred Heart, 25121 Brescia, Italy; kateryna.medvedieva01@icatt.it (K.M.); tommaso.tosi@unicatt.it (T.T.); enrico.barbierato@unicatt.it (E.B.)
* Correspondence: alice.gatti@unicatt.it

Abstract: The increasing sophistication of cyberattacks necessitates the development of advanced detection systems capable of accurately identifying and mitigating potential threats. This research addresses the critical challenge of cyberattack detection by employing a comprehensive approach that includes generating a realistic yet imbalanced dataset simulating various types of cyberattacks. Recognizing the inherent limitations posed by imbalanced data, we explored multiple data augmentation techniques to enhance the model's learning effectiveness and ensure robust performance across different attack scenarios. Firstly, we constructed a detailed dataset reflecting real-world conditions of network intrusions by simulating a range of cyberattack types, ensuring it embodies the typical imbalances observed in genuine cybersecurity threats. Subsequently, we applied several data augmentation techniques, including SMOTE and ADASYN, to address the skew in class distribution, thereby providing a more balanced dataset for training supervised machine learning models. Our evaluation of these techniques across various models, such as Random Forests and Neural Networks, demonstrates significant improvements in detection capabilities. Moreover, the analysis also extends to the investigation of feature importance, providing critical insights into which attributes most significantly influence the predictive outcomes of the models. This not only enhances the interpretability of the models but also aids in refining feature engineering and selection processes to optimize performance.

Keywords: data augmentation; supervised learning; cybersecurity

Citation: Medvedieva, K.; Tosi, T; Barbierato, E.; Gatti, A. Balancing the Scale: Data Augmentation Techniques for Improved Supervised Learning in Cyberattack Detection. *Eng* **2024**, *5*, 2170–2205. https://doi.org/10.3390/eng5030114

Academic Editors: Antonio Gil Bravo and Alessandro Polo

Received: 8 July 2024
Revised: 13 August 2024
Accepted: 30 August 2024
Published: 4 September 2024

1. Introduction

The National Institute of Standards and Technology (NIST www.nist.gov/cybersecurity accessed on 1 June 2024) provides multiple definitions of cybersecurity, which may be collectively synthesized into a singular principle: cybersecurity entails the implementation of protective measures and controls designed to prevent harm and secure information stored in computer systems or transmitted through communication networks. This practice is paramount to ensure the availability, integrity, and confidentiality of information.

A report by the European Union Agency for Cybersecurity (ENISA www.enisa.europa.eu/publications/enisa-threat-landscape-2022 accessed on 15 June 2024) underscores a significant augmentation in cyberattacks toward the end of 2022 and into the initial half of 2023. The ongoing Russian invasion of Ukraine is pinpointed as a primary catalyst for this uptick. Additionally, the report identifies eight major threat groups, highlighting the dynamic and evolving nature of cyber threats.

Threats against data are categorized into data leaks and data breaches, which differ primarily in the intent behind the exposure. A data leak is usually an unintentional exposure due to human error or system vulnerabilities, whereas a data breach is a deliberate attack aimed at stealing information.

Threats against availability include Denial of Service (DoS) attacks, which disrupt normal operations by overwhelming systems or networks with excessive traffic, typically

originating from multiple sources. Additionally, internet threats comprise a range of deliberate or accidental disruptions to electronic communications, leading to outages, blackouts, shutdowns, or censorship caused by various factors including government actions, natural disasters, or cyberattacks. Information manipulation involves actions that influence values, procedures, and political processes, often with a manipulative agenda. While not always illegal, these actions pose significant threats by potentially undermining democratic principles, destabilizing societies, and eroding institutional trust. Supply chain attacks represent sophisticated cyberattacks targeting the interconnected relationships between organizations and their suppliers. These attacks can involve malicious code in software updates or hardware components, compromised supplier credentials, or exploited third-party service or product vulnerabilities.

Machine learning (ML) plays a crucial role in classifying and detecting cyberattacks by analyzing vast amounts of data and identifying patterns indicative of malicious activity. One significant advantage of ML is its ability to detect anomalies, which can indicate potential threats such as zero-day attacks and advanced persistent threats (APTs). Moreover, ML automates the threat identification process, reducing the time needed to detect and respond to threats, which helps mitigate damage and minimize downtime. ML's pattern recognition capabilities are particularly useful in identifying specific types of cyberattacks, thereby improving threat detection accuracy. Additionally, ML systems are scalable and can handle large-scale data, making them suitable for enterprises with extensive networks. Furthermore, ML models can continuously learn from new data, enhancing their detection capabilities over time. This adaptive learning is essential for staying ahead of emerging threats. Another benefit of ML in cybersecurity is its ability to reduce false positives in threat detection, allowing security teams to focus on genuine threats and avoid unnecessary alerts. By automating many aspects of threat detection and response, ML also reduces the operational costs associated with manual security monitoring and incident response.

ML models are trained using datasets to classify various types of cyberattacks. However, these datasets are frequently imbalanced, meaning that some types of attacks are significantly underrepresented compared to others. This imbalance poses a critical problem because the models trained on such datasets tend to become biased towards the more common attack types, resulting in poor detection rates for the rarer, yet potentially more dangerous, attacks. For example, if a dataset contains a vast majority of data on phishing attacks but very few instances of zero-day exploits, the model will likely excel at identifying phishing but fail to recognize the zero-day exploits. To address imbalance issues, cybersecurity researchers and practitioners use techniques such as data augmentation, synthetic data generation, and advanced ML algorithms designed to handle imbalanced data. These methods help ensure that the models remain robust and capable of accurately identifying both common and rare cyber threats, thereby providing a more secure defense against a wide range of attacks.

However, these efforts alone are not particularly compelling unless the ML models are also employed to assign weights to dataset features. This weighting process is crucial as it helps identify which features have the greatest influence on the model's responses. By understanding feature importance, researchers and practitioners can gain deeper insights into the factors driving cyber threats, leading to more effective detection and prevention strategies. Without this, the models remain black boxes, offering limited value beyond mere classification.

1.1. Aims of the Research

The primary objective of this research was to explore and evaluate various data augmentation techniques to enhance the effectiveness of supervised learning models in detecting cyberattacks. By generating a realistic dataset that simulates different types of cyberattacks, the study aimed to reflect the typical imbalances observed in genuine cybersecurity threats. Subsequently, the application of multiple data augmentation techniques,

including SMOTE and ADASYN, sought to correct the skewed class distribution, providing a more balanced dataset for training supervised machine learning models.

Furthermore, the study hypothesized that the implementation of these data augmentation techniques would lead to significant improvements in the detection capabilities of various models, such as Random Forests and Neural Networks. Another key hypothesis was that the analysis of feature importance would offer critical insights into which attributes most significantly influence the predictive outcomes of the models, thereby enhancing their interpretability and aiding in the refinement of feature engineering and selection processes.

1.2. Contribution of This Work

The contribution of this research consists of the following:

- A novel, realistic dataset that simulates various types of cyberattacks, mirroring the complex and imbalanced nature of real-world cybersecurity scenarios;
- Furthermore, a comprehensive application of several data augmentation techniques, including SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling), is adapted specifically for enhancing the dataset within the cybersecurity context;
- Additionally, the research includes a systematic evaluation of how different machine learning models perform following data augmentation, identifying those most effective in recognizing diverse cyber threats;
- Finally, this exploration enhances the understanding of predictive features and contributes significantly to the models' transparency and explainability, essential for operational trust and effective deployment in cybersecurity environments.

The work is organized as per Figure 1. Specifically, Section 2 reviews the related work. Section 3 provides the background of this work, presenting some of the data augmentation techniques and machine learning methods used. Section 4 details the experiments run on a generated dataset, while Section 5 comments on the results. Section 6 draws some conclusions and provides a few comments on future work. Finally, a list of abbreviations used and more detailed results are given in Appendix A.

2. Related Work

Supervised learning is widely used in Intrusion Detection Systems (IDSs) due to its effectiveness in employing labeled datasets to train predictive models.

Apruzzese et al. [1] provide a comprehensive review of the deployment and integration of ML in cybersecurity. Key contributions include highlighting the benefits of ML over traditional human-driven detection methods and identifying additional cybersecurity tasks that can be enhanced by Supervised Learning. The study discusses intrinsic problems such as concept drift, adversarial settings, and data confidentiality that affect real-world ML deployments in cybersecurity. Limitations of the approach include the slow pace of integrating ML into production environments and the need for continuous updates to handle evolving threats. The article also presents case studies demonstrating industrial applications of ML in cybersecurity, emphasizing the necessity of collaborative efforts among stakeholders to advance ML's role in this field.

Mijwil et al. [2] explain how supervised and unsupervised learning methods, such as logistic regression and clustering, are utilized for intrusion and anomaly detection, while DL techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) effectively identify malware and cyber threats with high accuracy. Despite their potential, ML and DL face challenges, including the need for large datasets, high false positive rates, and the continuous evolution of cyber threats, necessitating regular updates and human oversight. The paper calls for ongoing research, better datasets, and integrated AI techniques to stay ahead of cybercriminals. Finally, it emphasizes the importance of further investigations into AI applications in cybersecurity, encouraging collaboration and the development of advanced techniques to protect digital environments.

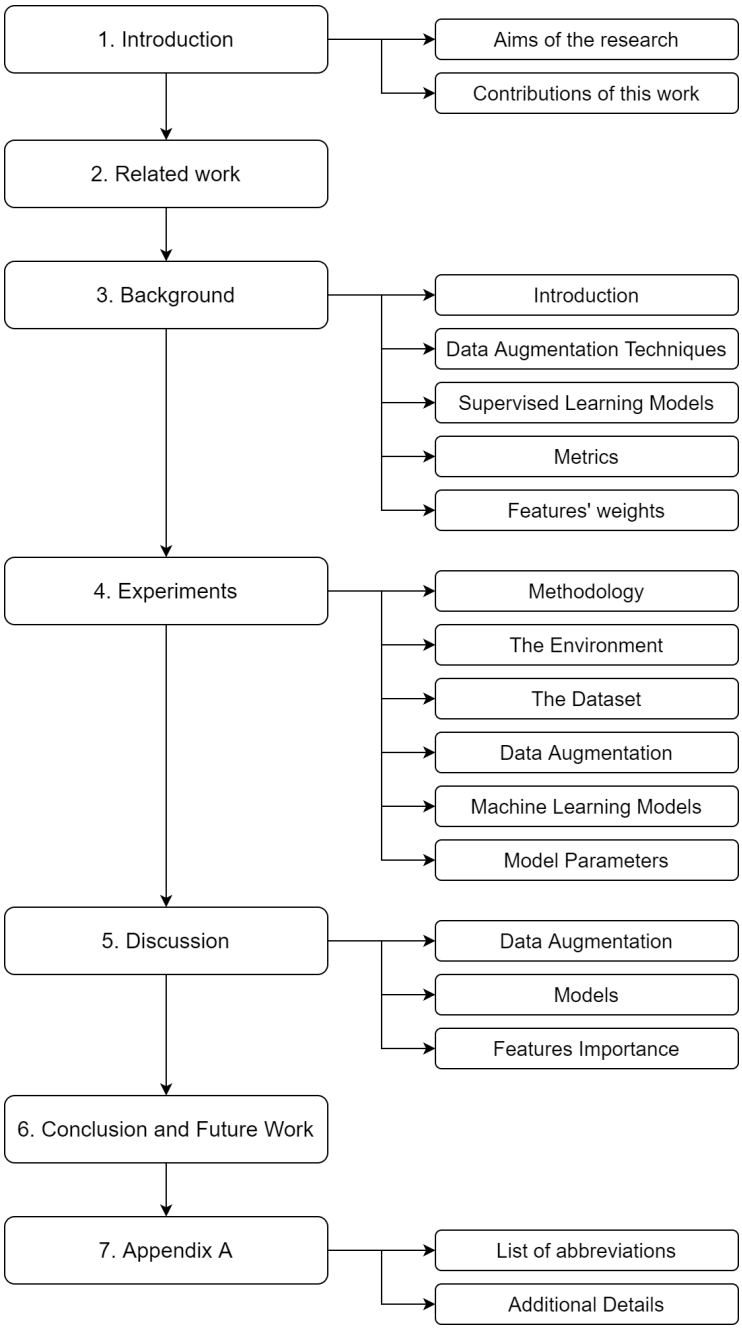

Figure 1. Visual representation of the work's structure.

A significant study by Bagui et al. [3] analyzed network connection logs using various supervised learning models, including Logistic Regression, Decision Trees (DTs), Random Forests (RFs), SVM, Naïve Bayes, and gradient-boosting trees. They introduced a new dataset, UWF-ZeekData22, which is publicly accessible through the University of West Florida's site. This dataset is labeled according to the MITRE ATT&CK framework, although the labeling process is not clearly documented. It includes 18 attributes, with the top six features identified based on information gain being the history of connection, transport layer protocol, application layer protocol, number of payload bytes the originator sent, destination IP, and number of packets the originator sent. The UWF-ZeekData22 dataset primarily covers two tactics: reconnaissance and discovery, with a significant imbalance between the number of observations for each tactic—2087 for discovery and 504,576 for reconnaissance. The focus of the research is more on classifying adversary tactics rather than specific techniques of attack. The strengths of the study by Bagui et al. include a

detailed presentation of a novel approach to preprocessing log data. The methodology involves binning numerical values (except for originator and destination ports) using a moving mean approach. Nominal features are converted into numerical labels, IP addresses are categorized based on the standard network classification of the first octet, and ports are grouped by ranges. A noted limitation in the study is its reliance on binary classification to assess model performance. The findings highlight that tree-based methods—specifically Decision Trees, gradient boosting trees, and Random Forests—excelled, achieving over 99% accuracy along with high precision, recall, f-measure, and AUROC scores.

Tufan et al. [4] explored a supervised ML model trained and tested on two distinct datasets. The first dataset consists of real-world private network data collected by the authors, and the second is the UNSW-NB15 dataset, developed by the Australian Center for Cyber Security. The data from the organizational environment required significant preparation, including the conversion of private IP addresses to public ones within captured packets. A noteworthy feature extraction technique involved analyzing packets with various TCP headers, such as ICMP, SYN, SYN-ACK, NULL, FIN, XMAS (PSH-URG-FIN), and FIN-ACK, with measurements taken every two seconds for each source IP. The preprocessing steps included the removal of redundant columns, those with empty values, and those with little variation. For categorical features, one-hot encoding was employed. This approach to handling missing values and detailing the labeling process, using open-source tools like Snort and Suricata to generate alerts from packet data, marks a comprehensive method where these alerts serve as labels for training. Tufan et al. utilized both filter and wrapper methods for feature selection, aiming to effectively refine the feature set. In contrast, the study by Bagui et al. [3]. performed attribute reduction using information gain to assess the importance and relevance of features. They compared two supervised models: an ensemble model consisting of Bayesian classifiers, KNN, Logistic Regression, and SVMs, against CNNs. A significant finding from their research was that CNNs outperformed the ensemble model in both datasets, highlighting the effectiveness of deep learning approaches in handling complex data structures and patterns.

In their study, Ravi et al. [5] introduced a deep-learning ensemble approach to enhance the performance of IDSs. The effectiveness of the proposed model was evaluated using several datasets, including SDN-IoT, KDD-Cup-1999, UNSW-NB15, WSN-DS, and CICIDS-2017, although only samples from the last four were considered. These datasets were preprocessed to ensure normalization. The models implemented for classification and feature extraction were RNNs, LSTMs, and Gated Recurrent Units (GRUs). These neural networks demonstrated a high efficacy in detecting attacks across all datasets. The next phase of the proposed methodology involved dimensionality reduction using Kernel Principal Component Analysis (KPCA) to improve the manageability and effectiveness of the data. During the feature fusion process, features extracted from different models were concatenated to form a comprehensive feature set. This combined feature set was then processed by base-level classifiers, specifically SVMs and Random Forests. The outputs from these classifiers were subsequently analyzed by a meta-level classifier, Logistic Regression, to finalize the detection process. The performance metrics from this study highlighted significant success in attack classification, achieving over 98% accuracy on all datasets, except the KDD-Cup-1999 dataset, which recorded an accuracy of 89%.

Further exploration into unsupervised ML models was conducted by Verkerken et al. [6], who utilized datasets developed by the Canadian Institute for Cybersecurity. In this phase, they removed redundant features and eliminated samples with missing or infinite values, as well as duplicated rows. Notably, unlike in the study by Bagui et al. [3], previously cited, features such as IP addresses were discarded to avoid overfitting. They applied a variety of feature scaling techniques, including StandardScaler, RobustScaler, QuantileTransformer, and MinMaxScaler from the scikit-learn library. These preprocessing steps were essential for the subsequent application of models such as Principal Component Analysis (PCA), Isolation Forest, Autoencoder, and One-Class SVM, particularly focusing on anomaly detection. However, they observed a significant drop in model performance

on newer data collected in 2018, with the AUROC decreasing by an average of 30.45%, and a minimum drop of 17.85% for the One-Class SVM. The authors attributed these declines to changes in the distribution of attack labels between datasets and the inadequacy of model hyperparameters during validation.

Another innovative approach to data handling was proposed by Hwang et al. [7], aimed at addressing the challenge of high memory demand typically associated with storing traffic data. Their method focused on analyzing only the initial bytes of the first few packets in a flow, significantly reducing data storage requirements. This approach showcases a practical solution to one of the fundamental issues in network traffic analysis, highlighting the potential of unsupervised learning techniques in IDSs. Each of these studies contributes to the evolving landscape of ML applications in cybersecurity, demonstrating various strategies to enhance the effectiveness and efficiency of IDSs.

In their innovative study, Aamir and Zaidi [8] introduced a semi-supervised approach based on clustering techniques. The primary objective was to employ various clustering methods to initially label the dataset, which would then facilitate the training of supervised algorithms for classification.

A significant limitation of this study was its reliance on synthetic datasets generated through simulation. Unlike other studies, the feature set used here was relatively small, comprising variables such as traffic rate, processing delay, and server CPU utilization, which were deemed sufficient for detecting Distributed Denial of Service (DDoS) attacks. The dataset's size was also restricted, containing only 1000 observations, which may not adequately represent more complex attack scenarios. Nonetheless, the results from this dataset were compared with a subset of the CICIDS2017 dataset that specifically focused on DDoS attacks.

After the collection and normalization of the dataset, two clustering techniques —agglomerative clustering and K-means—were applied. Notably, K-means was enhanced by principal components obtained from PCA. The subsequent step involved a voting mechanism to reconcile the clustering results: if an observation was consistently labeled across both results, it was assigned a definitive class (benign or DDoS); otherwise, it was tagged as "Suspicious". This process was crucial for creating a reliably labeled dataset for subsequent supervised learning.

Supervised algorithms used included K-nearest neighbor (KNN), Support Vector Machines (SVMs), and Random Forest, with hyperparameters finely tuned during the training phase. The classification accuracy on the synthetic dataset was impressive, particularly for Random Forest, which achieved a 96.66% accuracy rate. This methodology was also validated on the aforementioned subset of CICIDS2017, achieving over 86% accuracy, underscoring the effectiveness of this semi-supervised approach.

Concerning data augmentation, Maharana et al. [9] extensively reviewed augmentation techniques, particularly focusing on their application in ML for image data. Techniques such as flipping, cropping, rotation, and color space adjustments were explored, detailing how they help in creating varied datasets from limited data sources. These methods are critical for reducing model overfitting and improving the robustness of the predictions.

Naik et al. [10] discuss the integration of AI with traditional cybersecurity strategies, noting how AI can bring a significant improvement in handling cyber threats through technologies like Big Data, Blockchain, and Behavioral Analytics. In detail, it provides an in-depth analysis of both "distributed" AI methods (such as Multi-Agent Systems, Artificial Neural Networks, Artificial Immune Systems, and Genetic Algorithms) and "compact" AI methods (including ML Systems, Expert Systems, and Fuzzy Logic). These classifications help differentiate AI techniques based on their application scope and complexity.

Table 1 outlines a comparative review of the methodologies and outcomes from the discussed studies focused on ML and data augmentation approaches to intrusion detection.

Table 1. Comparison of related work in IDS using ML and data augmentation techniques.

Paper	Year	Strengths	Limitations
Hwang et al. [7]	2020	Approach used tried to minimize memory usage.	No standard classification.
Verkerken et al. [6]	2021	Multiclass classification. Comparison between intra- and inter-dataset evaluations.	No feature selection. No standard classification.
Aamir and Zaidi [8]	2021	Hyperparameter tuning. Validation of the approach on a synthetic dataset and benchmark dataset.	Synthetic dataset. Focus only on DDoS attacks. Using a small number of variables
Tufan et al. [4]	2021	Hyperparameter tuning. Comparison of the models trained and tested on private and publicly available datasets. Feature selection.	Focus only on probing. Designed to work offline. The institutional dataset was collected from a specific environment. No standard classification.
Bagui et al. [3]	2022	Created a new dataset labeled according to MITRE framework. Unique and detailed preprocessing strategy. Feature selection based on information gain. Comparison between models also in terms of using important features (top 6, top 9, all 18)	Dataset contains only 2 tactics. Using binary classification. Absence of hyperparameter tuning. Designed to work offline. Many duplicates in the dataset.
Ravi et al. [5]	2022	Multiclass classification. Used the most common benchmark datasets.	Proposed approach is sensitive to imbalanced datasets. No standard classification.
Maharana et al. [9]	2022	Methodological diversity, educational value	Lack of empirical data, focus on ML
Naik et al. [10]	2022	Comprehensive coverage, practical applications	Lack of case studies and evaluation metrics
Apruzzese et al. [1]	2023	Deep discussion on intrinsic problems	None
Mijwil et al. [2]	2023	Analysis of current ML and DL models applied in IDS, in terms of advantages and limitations	None
Agrawal et al. [11]	2024	Synthetic data generated by GANs	Lack of realism
Mohammad et al. [12]	2024	High accuracy in intrusion detection	Persistent challenge of class imbalance and the marginal performance improvements of complex DL models

In [11], Agrawal et al. explored the application of generative adversarial networks (GANs) in creating synthetic data for cybersecurity. Key contributions included comprehensively examining GANs' capabilities in generating realistic cyberattack data and their use in enhancing IDSs (IDS). The study identified challenges such as the efficacy of GAN-generated data in accurately representing real-world attacks and the need for further investigation into the robustness of deep learning models trained on synthetic data.

Limitations included persistent concerns about the quality and realism of the synthetic data produced by GANs. The paper emphasizes the importance of synthetic data in overcoming privacy and security concerns associated with real-world data sharing

Mohammed et al. [12] presented a method to improve IDS performance by combining deep learning architectures with data augmentation techniques. Key contributions included using four prominent datasets (UNSW-NB15, 5G-NIDD, FLNET2023, and CIC-IDS-2017) to demonstrate that simple CNN-based models can achieve a high accuracy in intrusion detection. Limitations highlighted include the persistent challenge of class imbalance and the marginal performance improvements observed with more complex deep learning architectures compared to simpler models. The study emphasized the importance of data quality and augmentation in enhancing detection capabilities.

3. Background

3.1. Introduction

In the rapidly evolving field of cybersecurity, ML plays a crucial role in enhancing threat detection and mitigation. Effective data augmentation techniques, such as SMOTE and ADASYN, are essential for addressing class imbalances in cybersecurity datasets. However, simply balancing datasets and training ML models is insufficient unless these models can also assign weights to dataset features. Understanding feature importance is key to identifying which factors most influence the model's responses, thereby improving the model's interpretability and effectiveness. This section explores various data augmentation methods, supervised learning models, and the significance of feature weighting in the context of cybersecurity, offering insights into optimizing model performance and decision-making.

3.2. Data Augmentation Techniques

Advanced techniques like synthetic data generation through methods such as SMOTE (Synthetic Minority Over-sampling Technique [13]) can also be employed to enrich the dataset without losing valuable information. Given a sample x_i from the minority class, SMOTE identifies its k nearest neighbors in the feature space. Let x_{nn} denote one of these k nearest neighbors. A synthetic sample x_{new} is generated by interpolating between x_i and x_{nn} using the equation:

$$x_{\text{new}} = x_i + \lambda(x_{nn} - x_i)$$

where λ is a random number between 0 and 1. This interpolation step creates a new sample that is a linear combination of the original sample and its neighbor, thus preserving the general data distribution while expanding the minority class.

ADASYN (Adaptive Synthetic Sampling, [14]) extends SMOTE by focusing more on generating synthetic samples next to the minority class samples that are wrongly classified by a classifier. Mathematically, ADASYN calculates the number of synthetic samples to generate for each minority class sample x_i by using a density distribution:

$$r_i = \frac{\gamma_i}{\sum_{i=1}^{N} \gamma_i}$$

where γ_i is the number of majority class samples in the k nearest neighbors of x_i. The number of synthetic samples G_i to generate for each x_i is proportional to r_i.

In cases where classes overlap significantly, SMOTE and ADASYN can introduce synthetic samples in regions where the classes are not well-separated. This can lead to increased misclassification, as synthetic samples in overlapping regions may be misclassified, reducing the overall performance of the model. Moreover, introducing synthetic samples in overlapping regions can blur the decision boundaries, making it difficult for the classifier to distinguish between classes.

When synthetic samples introduce noise, both SMOTE and ADASYN can suffer from decreased performance. In particular, synthetic samples that are not representative of

the actual data distribution can introduce noise, leading to poor generalization of the model. Furthermore, the presence of noisy samples can cause the model to overfit the synthetic data, reducing its ability to perform well on unseen data. Additionally, noise can adversely affect precision and recall, as the model may produce more false positives and false negatives.

However, several strategies can be employed to mitigate the issues of overlapping classes and noise. For example, the data can be preprocessed to remove noise before applying SMOTE or ADASYN. Secondly, more sophisticated techniques (such as Borderline-SMOTE or SVM-SMOTE) that focus on generating samples near the decision boundary can be used. Finally, it is possible to continuously evaluate the performance and tune the parameters of the over-sampling techniques to minimize the introduction of noise.

Borderline-SMOTE [15] specifically targets minority class samples that are close to the boundary with the majority class. It uses the same interpolation strategy as SMOTE but restricts it to those minority samples whose nearest neighbors include majority class samples. The synthetic sample generation formula remains as per the SMOTE technique.

Tomek-links data augmentation [16] is a technique used primarily to enhance the performance of classifiers on imbalanced datasets. It involves identifying pairs of instances that are nearest neighbors but belong to different classes and removing them to increase the separability of the classes.

A Tomek-link exists between a pair of instances x_i and x_j from different classes if there is no instance x_k such that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_j, x_i)$, where d represents the distance metric used, often the Euclidean distance. Mathematically, it can be characterized as follows:

Let S be the set of all samples, then a pair $(x_i, x_j) \in S \times S$ forms a Tomek-link if

$$(y_i \neq y_j) \wedge (\nexists x_k \in S : (d(x_i, x_k) < d(x_i, x_j) \vee d(x_j, x_k) < d(x_j, x_i)))$$

This method is especially effective for binary classification problems and is often utilized as a data cleaning technique rather than an oversampling technique.

SMOTEENN [17] combines two approaches to address the issue of class imbalance in machine learning datasets: SMOTE (Synthetic Minority Over-sampling Technique) for over-sampling the minority class and ENN (Edited Nearest Neighbor) for cleaning the data by under-sampling both classes.

ENN removes any sample that has a majority of its k nearest neighbors belonging to a different class. For a given sample x_i, it is removed if

$$\frac{1}{k} \sum_{j=1}^{k} I(y_j \neq y_i) > 0.5$$

where I is an indicator function, y_j is the class label of the j-th nearest neighbor, and y_i is the class label of x_i.

SMOTEENN applies SMOTE to generate synthetic samples and then uses ENN to remove any generated or original samples that are misclassified by their nearest neighbors. This combination helps in refining the class boundaries further than using SMOTE alone.

Finally, SMOTE-Tomek [18] is a hybrid method that combines the SMOTE approach for over-sampling the minority class with Tomek links for cleaning overlapping samples between classes. This technique is particularly effective in improving the classification of imbalanced datasets by both augmenting the minority class and enhancing class separability. SMOTE-Tomek first applies SMOTE to generate additional synthetic samples to balance the class distribution. Subsequently, it applies the Tomek links method to remove any Tomek links identified between the synthetic and original samples. This removal process helps in reducing noise and making the classes more distinct, which is beneficial for the subsequent learning process.

It is interesting to discuss the computational costs of the data augmentation techniques and their impact on the model. The time complexity of SMOTE is $O(T \times k \times d)$, where T is the number of synthetic samples, k is the number of nearest neighbors, and d is the dimensionality of the data. This results in a moderate increase in training time due to the need to find nearest neighbors and generate synthetic samples. ADASYN has a similar time complexity to SMOTE but includes additional computations to determine the difficulty of instances, resulting in slightly higher computational costs and additional processing time compared to SMOTE. Borderline SMOTE shares the same time complexity as SMOTE but with additional steps to identify boundary samples. This leads to higher computational costs as identifying boundary samples requires extra computations. The time complexity for Tomek Links is $O(n^2 \times d)$, where n is the number of samples and d is the dimensionality. This results in significant preprocessing time due to the need to compute pairwise distances between samples, thus impacting overall model training time. SMOTEENN combines the costs of SMOTE and the Edited Nearest Neighbors (ENNs) technique, typically $O(T \times k \times d) + O(n \times k \times d)$. The high computational cost results from combining synthetic sample generation and nearest neighbor cleaning, leading to a notable increase in resource usage. SMOTE Tomek combines the costs of SMOTE and Tomek Links, typically $O(T \times k \times d) + O(n^2 \times d)$. This combination results in very high computational costs due to extensive pairwise distance computations and synthetic sample generation, significantly impacting training time and resource consumption.

The computational costs of these techniques directly impact model training time and resource usage. Techniques with higher time complexity, such as SMOTE Tomek and SMOTEENN, significantly increase preprocessing time and extend overall training time. High computational costs translate to increased CPU and memory usage, which can be limiting factors for large datasets or complex models. Techniques with quadratic time complexity (e.g., Tomek Links) may not scale well with large datasets.

Data augmentation is crucial in cybersecurity for generating more comprehensive datasets that can help in better training machine learning and deep learning models. The scientific literature proposes different surveys (see, for example, [1,19–21]) centered around machine learning models applied in cybersecurity.

3.3. Supervised Learning Models

This work considers some of the supervised learning models such as Naive Bayes, KNN, XGBoost (XGB), Gradient Boosting Machine (GBM), Logistic Regression, and Random Forest, as well as deep learning models like RNNs and LSTMs on cyberattack datasets.

Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence between features. Its strengths include being fast and efficient, especially with large datasets, and performing well with small amounts of training data. However, it assumes independence between features, which is rarely true in real-world data, and is not suitable for datasets with highly correlated features. KNN is a simple, non-parametric algorithm that classifies a sample based on the majority class among its k-nearest neighbors. KNN is simple to implement and understand, and it is effective for small datasets with well-defined classes. Its limitations are that it is computationally intensive with large datasets and its performance can degrade with high-dimensional data. XGB is an optimized distributed gradient boosting library designed to be highly efficient and flexible. XGB offers high performance and accuracy, and it handles missing values and large datasets well. However, it requires careful tuning of hyperparameters and can be prone to overfitting if not properly regularized. GBM builds an additive model in a forward stage-wise manner, optimizing differentiable loss functions. GBM is known for its high accuracy and robustness, and it is effective for both regression and classification tasks. The main limitations are that it is computationally intensive and slow to train, and it can be prone to overfitting without proper tuning. Logistic Regression is a linear model used for binary classification that predicts the probability of a categorical dependent variable. It is simple and interpretable, and it is efficient for binary and multinomial classification. However, it assumes a linear

relationship between the features and the log-odds of the outcome, and it is not suitable for complex datasets with non-linear relationships. The Random Forest Classifier is an ensemble learning method that constructs multiple decision trees and merges them to obtain a more accurate and stable prediction. Random Forest handles large datasets with higher dimensionality well and is robust to overfitting due to its ensemble nature. However, it is computationally intensive, especially with a large number of trees, and less interpretable than single decision trees.

RNNs are a class of neural networks where connections between nodes form a directed graph along a temporal sequence, allowing them to exhibit temporal dynamic behavior. RNNs are effective for sequence prediction problems and can handle time-series data and sequential data well. However, they are prone to the vanishing gradient problem, making training difficult, and require significant computational resources. Finally, LSTMs are a special kind of RNN capable of learning long-term dependencies and mitigating the vanishing gradient problem. LSTMs are capable of learning long-term dependencies and are effective for time-series and sequential data. However, they are computationally expensive and slower to train, and they require extensive hyperparameter tuning.

3.4. Metrics

Accuracy is a widely used metric that reflects the overall correctness of a model's predictions. It is calculated as the proportion of correctly predicted instances relative to the total number of instances within the dataset as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

While offering a general sense of model performance, accuracy can be misleading in scenarios with imbalanced datasets.

Precision, on the other hand, focuses on the proportion of true positive predictions among all predicted positives. It essentially measures the model's ability to accurately identify positive instances:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

A high precision value indicates a low false positive rate, signifying the model's proficiency in distinguishing between positive and negative cases.

Recall, alternatively referred to as sensitivity or true positive rate, measures the model's capacity to identify all relevant positive instances. It is calculated as the ratio of correctly predicted positive observations to the total actual positive observations within the dataset:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

A high recall value signifies a low false negative rate, implying the model's effectiveness in capturing all pertinent positive cases.

The F1-score addresses the potential shortcomings of relying solely on precision or recall by providing a harmonic mean of both metrics. This consolidated metric offers a balanced assessment, particularly valuable in situations with imbalanced class distributions. The F1-score is calculated as follows:

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

The F1-score ranges from 0 to 1, with a value of 1 signifying the ideal scenario where both precision and recall are perfect.

Finally, support refers to the total number of actual occurrences for each class within the dataset. While not a direct measure of model performance, support provides crucial context for interpreting the other metrics. By understanding the class distribution and the

number of data points per class (support), we can more effectively evaluate the significance of the calculated precision, recall, and F1-score values.

3.5. Features' Weights

CatBoost [22] is a machine learning algorithm developed by Yandex, which is part of the family of gradient boosting algorithms. The term "CatBoost" reflects its capability to handle categorical features effectively and its nature as a boosting algorithm. It is specifically designed to offer high performance with a focus on speed and accuracy, which is particularly advantageous when dealing with categorical data. This model optimizes the gradient boosting process through the use of symmetric trees and Oblivious Trees, enhancing both speed and accuracy while mitigating overfitting. This makes the model particularly robust and suitable for large datasets, with an implementation that supports GPU acceleration and multi-core processing.

A key feature of CatBoost is its capability to provide insights into the importance of features in the model. Within the domain of cybersecurity, the process of assigning weights to variables during data analysis holds paramount importance for several compelling reasons. Firstly, cybersecurity necessitates the examination of vast datasets to identify anomalies, potential threats, and existing vulnerabilities. Assigning weights to variables allows for the discernment of the most impactful factors contributing to potential security breaches. This acquired knowledge empowers cybersecurity professionals to prioritize their efforts on the most critical aspects, ultimately enhancing the efficacy of threat detection and mitigation strategies.

Secondly, the ever-evolving nature of cybersecurity threats presents a constant challenge. Attackers continuously develop novel techniques to exploit system vulnerabilities. Through the assessment of variable weights, security models can be dynamically adapted and updated to reflect the evolving threat landscape. This dynamic approach ensures the continued relevance and robustness of security measures in the face of emerging threats.

Finally, the assessment of variable weights also facilitates the explainability and interpretability of machine learning models. In the context of cybersecurity, comprehending the rationale behind a specific alert generation is crucial. This transparency not only fosters trust-building with stakeholders but also aids in forensic investigations to trace the origins and methodologies employed in cyberattacks.

4. Experiments

4.1. Methodology

The methodology employed in this study was aimed to address the challenges posed by imbalanced datasets in cyberattack detection. Initially, a realistic dataset was constructed to simulate various types of cyberattacks, reflecting the imbalances typically observed in real-world cybersecurity threats. This dataset serves as the foundation for evaluating the effectiveness of different data augmentation techniques.

To correct the skewed class distribution, the study applied several data augmentation techniques, notably SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling). These techniques generate synthetic samples to balance the dataset, thereby providing a more equitable distribution of classes for training supervised machine learning models.

Following the augmentation, various supervised machine learning models were trained and evaluated on the balanced dataset. The performance of these models was assessed using standard metrics such as accuracy, precision, recall, and F1-score to determine the improvements brought by the data augmentation techniques (see Appendix A for more details).

Additionally, the study analyzed feature importance to identify which attributes most significantly influence the predictive outcomes of the models. This analysis not only enhances the interpretability of the models but also provides valuable insights for refining feature engineering and selection processes, thereby optimizing model performance.

4.2. The Environment

The laboratory environment used VMware vSphere version 8.0.2.00300 for creating and managing virtual networks. This setup is crucial for developing and testing network security solutions effectively. The Security Onion server, version 2.4.60, was configured with two network interfaces. One interface was connected to the internal network, and the other was used for Docker container communications. The server was equipped with 12 CPUs, 24 GB of RAM, and a 250 GB hard disk configured as Thick Provision Lazy Zeroed. A Windows Server 2022 Standard Evaluation version (21H2) was set up with 2 CPUs, 12 GB of RAM, and a 90 GB hard disk.

The Windows 10 Pro client (Version 22H2) operated with 2 CPUs, 4 GB of RAM, and a 48 GB hard disk. Its network interface was connected to the same internal network, facilitating various network security experiments. An Ubuntu 22.04.4 LTS client was part of the network. Each system was configured with 2 CPUs, 3 GB of RAM, and a 25 GB hard disk. These clients interacted with other network components to simulate real-world traffic and attack scenarios. The main web server (APACHE01) and the reverse proxy server both ran on Ubuntu 22.04.4 LTS with similar hardware configurations as the Ubuntu clients. They play critical roles in hosting and securing web applications.

To simulate realistic network traffic, several generators were implemented. These include a generator for creating traffic from the internal network to the Internet based on the "noisy" project. This project involves collecting and visiting links from specified root URLs recursively until no more links are available or a timeout is reached.

APACHE01 is protected by a reverse proxy server (APACHE-REVERSE-PROXY), which is exposed to the Internet through a firewall, enhancing the security of hosted applications. Additionally, traffic and activities are monitored using tools like Security Onion to provide insights into network traffic and potential security threats.

4.3. The Dataset

The dataset features and their meaning are depicted in Table 2.

Table 2. Description of variables in the cyberattack dataframe.

Variable Name	Description
`resp_pkts`	Number of packets sent by the responder during the connection.
`service`	The type of service being accessed (e.g., HTTP, FTP).
`local_resp`	Indicates whether the responder is local to the network.
`protocol`	Network protocol used in the connection (e.g., TCP, UDP).
`duration`	Duration of the connection in seconds.
`conn_state`	State of the connection (e.g., established, closed).
`orig_pkts`	Number of packets sent by the originator during the connection.
`dest_port`	Destination port number of the connection.
`orig_bytes`	Number of bytes sent by the originator during the connection.
`local_orig`	Indicates whether the originator is local to the network.
`resp_bytes`	Number of bytes sent by the responder during the connection.
`src_port`	Source port number of the connection.
`techniques_mitre`	MITRE ATTACK technique(s) associated with the cyberattack.

Initially, the dataset consisted of 436,404 rows, which were reduced to 307,658 after validation (many observations were duplicated, and the features {duration, orig_bytes, resp_bytes} included 98,844 *NaN* values) and normalization. This preprocessing brought the total number of rows in the dataset to 208,735.

Notably, the techniques_mitre variable takes the following values:

- network_service_discovery;
- benign;
- reconnaissance_vulnerability_scanning;
- reconnaissance_wordlist_scanning;
- remote_system_discovery;
- domain_trust_discovery;
- account_discovery_domain;
- reconnaissance_scan_ip_blocks.

Network service discovery refers to the process of identifying and characterizing services running on networked devices. Adversaries employ techniques such as port scanning and service enumeration to identify open ports, listening services, and their corresponding software versions. This information aids in pinpointing potential vulnerabilities within the network and constructing a comprehensive network topology map.

It is vital to distinguish between benign activities and malicious network reconnaissance. Benign activities encompass actions inherent to normal system operations, including legitimate software updates, routine maintenance procedures, and standard user behavior. In contrast, malicious network reconnaissance, as detailed in the subsequent sections, involves deliberate attempts to exploit vulnerabilities and compromise system security.

Reconnaissance vulnerability scanning involves the systematic interrogation of target systems to identify exploitable weaknesses. Adversaries leverage this technique to detect outdated software, misconfigurations, and other security gaps. The primary objective is to amass information that can be later utilized to gain unauthorized access or execute malicious actions.

This technique involves leveraging pre-defined lists of words or phrases (wordlists) to systematically probe potential points of interest within a target environment. Adversaries utilize wordlists to conduct brute-force attacks or attempt to guess critical information such as usernames, passwords, URLs, and other sensitive data. Reconnaissance wordlist scanning frequently complements other reconnaissance activities to enhance attack efficiency and accuracy.

Remote system discovery is the process of gathering information about remote systems on a network. This can involve identifying active hosts, network shares, and accessible resources. Techniques used for remote system discovery include ping sweeps, port scanning, and querying network services. The ultimate objective is to map the network layout and pinpoint potential targets for subsequent exploitation attempts.

Within domain environments, adversaries utilize domain trust discovery to comprehend the trust relationships established between various domains. Understanding these trust relationships can provide adversaries with pathways for lateral movement and privilege escalation. This may involve identifying trusted domains, domain controllers, and any cross-domain policies that govern access control.

Account discovery (domain) is a technique employed by adversaries to enumerate user accounts within a domain environment. This involves discovering usernames, associated user groups, and corresponding permissions. The information gleaned can be utilized to plan attacks that involve credential theft, privilege escalation, and lateral movement within the compromised domain. Common methods for account discovery include querying Active Directory and leveraging built-in domain commands.

Reconnaissance scan IP blocks involve systematically scanning large ranges of IP addresses to identify active devices and services. Adversaries utilize this technique to map the target network infrastructure and pinpoint potential targets for further exploitation. This type of scanning can reveal critical information such as the number of active hosts, operating systems in use, and network devices present within the target environment.

Finally, Group Policy Discovery refers to the process of identifying and analyzing Group Policy Objects (GPOs) within a Windows domain environment. Adversaries examine GPOs to gain insights into security configurations, administrative templates, and user policies. This information can be used to identify misconfigurations, understand the

deployed security controls, and pinpoint potential weaknesses that can be exploited to achieve their malicious goals.

As per Table 3, the dataset appears to be imbalanced due to a significant disproportion in the occurrences of different features, specifically within the "techniques_mitre distribution". Imbalanced datasets are commonly encountered in machine learning and statistical analysis and can lead to biased models that inadequately represent the minority classes. The feature "network_service_discovery" exhibits an overwhelming dominance with 144,279 occurrences, which is nearly 2.4 times that of the next most frequent category, "benign", which has 60,997 occurrences. This dominant feature may lead predictive models to exhibit a strong bias towards predicting this category, potentially at the expense of accuracy in other less frequent categories.

Table 3. Distribution of techniques_mitre.

Techniques_Mitre Distribution	Occurrences
network_service_discovery	144,279
benign	60,997
reconnaissance_vulnerability_scanning	1581
reconnaissance_wordlist_scanning	715
remote_system_discovery	554
domain_trust_discovery	411
account_discovery_domain	84
reconnaissance_scan_ip_blocks	80
group_policy_discovery	34

Moreover, categories such as "group_policy_discovery", "reconnaissance_scan_ip_blocks", and "account_discovery_domain" are extremely underrepresented with only 34, 80, and 84 occurrences, respectively. This sparse representation complicates the learning process for statistical models, as there are insufficient data to achieve a good generalization performance on new or unseen data falling into these categories. The vast range in feature distribution, from the most to the least frequent (144,279 occurrences vs. 34 occurrences), highlights the stark imbalance, indicating not only a skew towards certain features but also a significant under-representation of others.

Machine learning algorithms generally perform better when the numbers of instances for each class are approximately equal. An imbalanced dataset can result in models that are biased towards classes with more instances, increasing the likelihood of misclassification of minority class instances. This can severely affect the model's accuracy, particularly its ability to detect less frequent but potentially important categories.

4.4. Data Augmentation

Addressing this imbalance might involve employing techniques such as oversampling the minority classes, undersampling the majority classes, or using approaches like SMOTE, ADASYN, Borderline-SMOTE, Tomel-Links, SMOTEENN, and SMOTE Temek.

Table 4 offers a comprehensive comparison of various data augmentation techniques applied to an imbalanced dataset categorized under "techniques_mitre". These techniques include SMOTE, ADASYN, Borderline-SMOTE, Tomek Links, SMOTEENN, and a combination of SMOTE and Tomek Links, each tailored to modify the distribution of minority and majority classes through synthetic data generation or data cleaning.

The application of these augmentation methods aimed to normalize the occurrence rates across categories to a target number, approximately 115,474, for most methods, indicative of the level set to achieve class balance.

Table 4. Comparison of data augmentation techniques.

Techniques _MITRE	Original	SMOTE	ADASYN	Borderline-SMOTE	Tomek-Links	SMOTEENN	SMOTE Tomek
Benign	144,279	115,474	115,870	115,474	48,353	107,187	114,001
Account Discovery Domain	60,997	115,474	115,480	115,474	40	114,351	115,334
Domain Trust Discovery	1581	115,474	115,533	115,474	238	113,080	115,030
Group Policy Discovery	715	115,474	115,473	115,474	25	114,503	115,258
Network Service Discovery	554	115,474	115,474	115,474	115,467	115,405	115,470
Reconnaissance Scan IP Blocks	411	115,474	115,478	115,474	62	115,357	115,473
Reconnaissance Vulnerability Scanning	84	115,474	115,751	115,474	1004	111,585	114,734
Reconnaissance Wordlist Scanning	80	115,474	115,475	115,474	577	115,474	115,474
Remote System Discovery	34	115,474	115,475	115,474	431	113,998	115,324

4.5. Machine Learning Models

Machine learning models like Naïve Bayes, K-nearest neighbor (KNN), XGBoost (XGB), Gradient Boosting Machine (GBM), Logistic Regression, and Random Forest Classifier are often preferred in predictive analytics due to their diverse strengths and applicability across a wide range of problems. Each model brings a unique set of capabilities that makes it suitable for different types of data and predictive tasks. The preference for these models in various analytical scenarios stems from their ability to balance accuracy and computational efficiency while providing solutions that are easy to interpret and implement in real-world applications.

Table 5 reveals insightful trends regarding the behavior of these models under test conditions.

Table 5. Accuracy values for different classifiers and data augmentation methods.

Classifier	SMOTE	ADASYN	Borderline SMOTE	Tomek Links	SMOTEENN	SMOTE Tomek
Naïve Bayes	0.497	0.453	0.602	**0.718**	0.668	0.659
KNN	0.824	0.978	0.981	0.992	**0.993**	0.990
XGB	0.838	0.925	0.942	**0.993**	0.981	0.977
GBM	0.842	0.940	0.953	**0.989**	**0.989**	0.984
RF	0.833	0.985	0.985	0.994	**0.998**	0.996
Logistic	0.738	0.741	0.847	0.807	**0.860**	0.851
RNN	0.759	0.823	**0.888**	0.979	0.647	0.797
LSTM	0.819	0.875	0.916	**0.982**	0.945	0.944

Naïve Bayes, traditionally valued for its simplicity and efficiency in handling large datasets, shows moderate accuracy. This is expected given its assumption of feature independence, which might not always hold true in real-world datasets. KNN's performance is generally better, reflecting its capability to adapt its classification strategy based on the local data structure. However, KNN's reliance on feature scaling and the curse of dimensionality can sometimes affect its performance adversely.

XGB and GBM, both boosting models, exhibit high accuracy, underscoring their strength in dealing with complex datasets that involve non-linear relationships among features. These models build upon the errors of previous trees and, hence, can adaptively improve their predictions. The high performance is indicative of their robustness, but it also brings to light the need for careful parameter tuning to avoid fitting excessively to the training data.

Logistic Regression provides a reasonable accuracy that is useful in scenarios requiring probability estimation for binary outcomes. Its performance is generally less competitive compared to ensemble methods but offers valuable insights due to its interpretability.

Random Forest typically shows excellent accuracy due to its ability to reduce overfitting through averaging multiple decision trees. This model is effective in handling various types of data, including unbalanced datasets.

While the results suggest that ensemble methods like XGB, GBM, and Random Forest tend to provide higher accuracy, this must be balanced with the understanding that high accuracy can sometimes be a result of overfitting. Although overfitting is not the primary focus of this discussion, it is implicitly relevant when interpreting the high accuracy of complex models.

Finally, a CatBoost model was applied to the augmented dataset ("tomek_links.csv") obtained using the Tomek links approach. The result is shown in Figure 2.

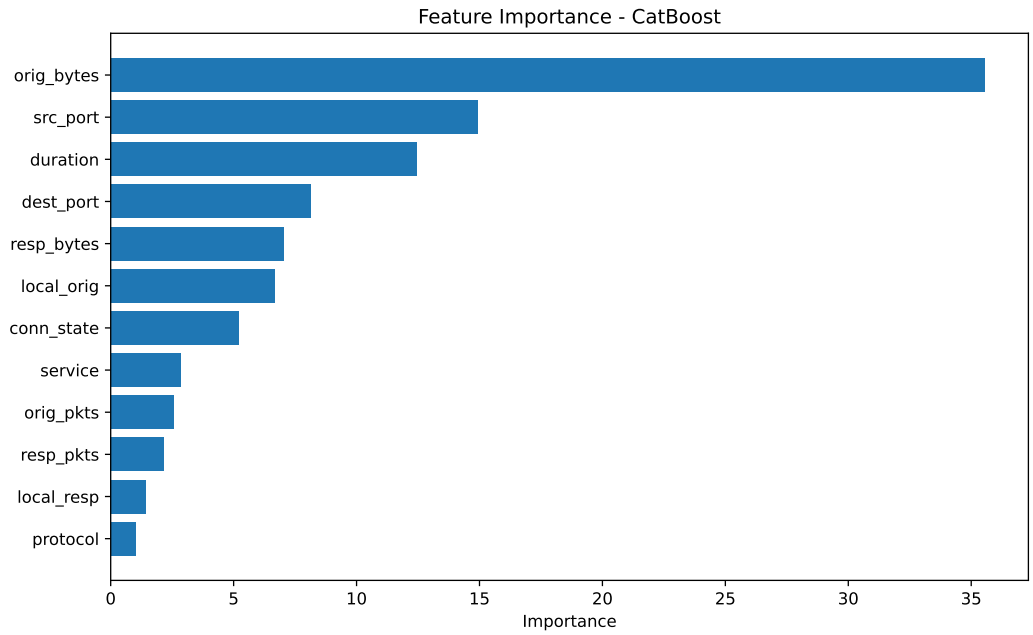

Figure 2. Features importance

4.6. Model Parameters

Table 6 shows the parameters employed by each model.

For GBM, key hyperparameters include the number of estimators, learning rate, and maximum depth. Tuning these involves the following:

- n_estimators: A higher number typically increases model complexity. Cross-validation helps find the optimal balance to avoid overfitting;
- learning_rate: Controls the contribution of each tree. Lower values typically require more trees;
- max_depth: Limits the depth of individual trees to control overfitting.

For KNN, the primary hyperparameter is the number of neighbors:

- _neighbors: A small number may lead to noisy predictions, while a large number can smooth out the prediction but may ignore local nuances. Grid search with cross-validation is commonly used to identify the optimal value.

Table 6. Models and their parameters.

Model	Parameters
GBM	n_estimators = 100, learning_rate = 0.1, max_depth = 3
KNN	n_neighbors = 5
Logistic Regression	max_iter = 10,000, class_weight = 'balanced'
LSTM	optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy']
GaussianNB	none
Random Forest	n_estimators = 100
RNN	optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy']
XGB	n_estimators = 100, learning_rate = 0.1, max_depth = 3, eval_metric = 'mlogloss'

Key hyperparameters include the maximum number of iterations and class weights:

- max_iter: Ensures convergence. Higher values allow the solver more iterations to converge, which is especially useful for complex datasets;
- class_weight: Balances the dataset by adjusting weights inversely proportional to class frequencies. It is particularly important in imbalanced datasets.

Critical hyperparameters for LSTM include the optimizer, loss function, and metrics:

- optimizer: "Adam" is commonly used for its adaptive learning rate capabilities;
- loss: "categorical_crossentropy" is used for multi-class classification problems;
- metrics: "accuracy" is a standard metric for evaluating classification performance.

GaussianNB typically requires no hyperparameter tuning as it is a straightforward probabilistic model. In contrast, important hyperparameters include the number of estimators:

- n_estimators: The number of trees in the forest. More trees generally improve performance but increase computation time.

Similar to LSTM, important hyperparameters include the optimizer, loss function, and metrics:

- optimizer: "Adam" is preferred for its efficiency and performance;
- loss: "categorical_crossentropy" for multi-class classification;
- metrics: "accuracy" for performance evaluation.

Finally, for XGBoost, key hyperparameters include the number of estimators, learning rate, maximum depth, and evaluation metric:

- n_estimators: Determines the number of boosting rounds;
- learning_rate: Lower rates require more boosting rounds;
- max_depth: Controls the depth of each tree to prevent overfitting;
- eval_metric: "mlogloss" is used for multi-class classification.

5. Discussion

5.1. Data Augmentation

The application of SMOTE and ADASYN was particularly effective in raising the number of instances in the minority classes to those in the majority classes, highlighting their capability to enhance intra-class variance through the generation of synthetic samples based on feature space similarities between existing minority samples. Similarly, Borderline-SMOTE, focusing on samples near the class borders, uniformly increased minority class counts, potentially aiding model accuracy in borderline cases.

The method involving Tomek Links, which removes pairs of closely situated opposite class samples, showed a substantial reduction in categories like "Benign", suggesting its efficacy in reducing major class sizes, thus deprioritizing the majority bias in data. SMOTEENN, which combines SMOTE's over-sampling with ENN's noise cleaning, appeared to both augment and cleanse the dataset by adding to the minorities and removing outliers or noise, respectively.

Combining SMOTE with Tomek Links resulted in a similar effect to SMOTE but with a slight reduction, indicating a cleaning effect on synthetic samples. These augmentation techniques signify a robust effort to address dataset imbalances, aiming to enhance the fairness and efficacy of predictive models.

5.2. Models

KNN achieved a high accuracy across all augmentation methods, with the highest accuracy of 0.993 using SMOTEENN. This model benefits from data augmentation, particularly with techniques like SMOTEENN and Tomek Links, which help in addressing class imbalance effectively. The accuracy of XGB also remained high across all augmentation methods, peaking at 0.993 with Tomek Links. XGB's robustness and ability to handle various types of augmented data contributed to its consistently high performance.

GBM showed a similar trend to XGB, with the highest accuracy of 0.989 using Tomek Links and SMOTEENN. The boosting approach in GBM makes it resilient to overfitting, even when augmented data are used. Random Forest (RF) achieved the highest accuracy of all models, with a maximum of 0.998 using SMOTEENN. The ensemble nature of RF allows it to generalize well across different augmented datasets.

Logistic Regressor displayed moderate performance, with the highest accuracy of 0.860 using SMOTEENN. The linear nature of this model might limit its ability to fully leverage the complex patterns introduced by some augmentation methods. RNN showed variable performance, with a peak accuracy of 0.979 using Tomek Links. The sequential nature of RNNs may benefit from Tomek Links' ability to clean noisy samples. LSTM, like RNN, showed improved performance with Tomek Links (0.982) and also benefited from other methods like SMOTEENN and SMOTETomek. LSTM's capability to capture long-term dependencies aids in leveraging augmented data effectively.

The choice of data augmentation technique has a significant impact on the performance of machine learning models. Techniques like SMOTEENN and Tomek Links generally yield higher accuracies, especially for models such as KNN, XGB, GBM, and RF. These findings highlight the importance of selecting appropriate data augmentation methods based on the machine learning model's characteristics and the dataset's nature.

Addressing the computational costs and efficiency of the data augmentation techniques and models involves several considerations. While data augmentation techniques can significantly improve the balance of the dataset and enhance model performance, they also introduce additional computational overhead. This overhead stems from the need to generate synthetic samples, which can be resource-intensive, especially for large datasets. To mitigate these costs, this study explored optimization strategies that streamline the augmentation process without compromising the quality of the generated data. This involves selecting appropriate parameters for each technique to balance the trade-off between computational efficiency and the effectiveness of the augmentation. For instance, optimizing the number of nearest neighbors in SMOTE or adjusting the density distribution in ADASYN can reduce unnecessary computations.

5.3. Features Importance

Regarding the various features commonly involved in network traffic data, which are crucial for identifying potentially malicious activities, the CatBoost model provided the following ranked features by importance:

- orig_bytes : This feature, representing the number of bytes that originated from the source, is identified as the most significant predictor. The high importance of this feature suggests that the volume of data sent from the source is a critical indicator of anomalous behavior.
- src_port and dest_port: The source and destination ports also play vital roles, indicating that particular ports may be more susceptible to exploitation or are commonly used by attackers.

- duration: The duration of the connection is another key feature, with longer connections possibly being indicative of data exfiltration activities.
- resp_bytes and resp_pkts: These features represent the response bytes and packets, respectively, highlighting the importance of the response size and frequency in detecting unusual responses that could signify a breach.

Regarding typical attack patterns, large orig_bytes values combined with extended duration are typical indicators. Effective detection rules should flag high data volume transfers, especially if they occur during off-hours or from unexpected sources. Unusual src_port and dest_port activity can signify reconnaissance efforts. Detection rules should monitor for spikes in port activity or access attempts to ports not typically used by legitimate applications within the organization. Moreover, long duration sessions should be scrutinized, especially if coupled with high resp_bytes. This pattern can indicate persistent attackers attempting to maintain access or exfiltrate data over extended periods. Based on these observations, different strategies can be considered to develop more effective detection rules. For example, it is recommended to establish thresholds for orig_bytes and duration that, when exceeded, trigger alerts. Specifically, a rule might flag any outgoing connection exceeding a certain data volume within a specific timeframe. Rules can also identify unusual port usage patterns, such as multiple access attempts to non-standard ports or a high frequency of connection attempts within a short period. This would help detect port scanning and early stages of attacks. ML models can be trained to learn normal patterns of src_port, dest_port, orig_bytes, and resp_pkts. Any significant deviations from these learned patterns can be flagged as potential threats. More sophisticated attack patterns can be detected by combining multiple features. For instance, a rule could flag connections with high orig_bytes and a long duration originating from an uncommon src_port and targeting an uncommon dest_port. Finally, user and system behaviors over time should be monitored. Sudden changes in data transfer volumes or connection durations that deviate from established behavior profiles can indicate compromised accounts or systems.

Updating models in response to evolving cyber threats requires a dynamic and continuous approach to ensure that detection systems remain effective against new and sophisticated attack patterns. One potential strategy is the implementation of a continuous learning framework. In this framework, the model is periodically retrained using recent data, which helps incorporate the latest threat patterns and anomalies observed in the network traffic. Another strategy involves the use of ensemble learning techniques. By combining multiple models, each trained on different aspects or time frames of the data, the system can achieve greater robustness against varying attack strategies. Ensemble methods can also incorporate new models trained on recent data, allowing the system to integrate fresh insights without completely discarding the knowledge from older models. Moreover, implementing a feedback loop from the security operations center (SOC) can be highly beneficial. When a potential threat is detected, the SOC can provide feedback on whether it was a true positive or a false positive. This feedback can be used to fine-tune the model, improving its accuracy over time. Data augmentation techniques play a crucial role in updating models. By continuously generating synthetic data that reflect the latest attack patterns and scenarios, the training dataset can be expanded and diversified. This approach helps in maintaining the model's effectiveness against a wide range of threats, including those that may not be prevalent in the historical data. Anomaly detection can be enhanced by integrating unsupervised learning methods alongside supervised ones. While supervised models are trained on labeled data, unsupervised models can identify new and unusual patterns without prior knowledge. By combining these approaches, the detection system can adapt to novel attack methods that deviate from known patterns.

6. Conclusions and Future Work

The adoption of advanced data augmentation techniques within supervised learning models significantly enhances the robustness and efficacy of cyberattack detection systems. This research demonstrates that by integrating SMOTE, ADASYN, and Tomek links, not

only can the predictive accuracy be improved, but the generalizability of the models across diverse and evolving cyber threat landscapes can also be substantially enhanced.

Furthermore, our findings underscore the importance of leveraging a hybrid approach to data augmentation, which meticulously addresses the challenges of imbalanced datasets prevalent in cybersecurity applications. By employing these techniques, we successfully minimized the overfitting potential and improved the detection rates of cyberattacks.

As cybersecurity threats continue to evolve in complexity and subtlety, the ability of detection systems to adapt and respond with nuanced understanding becomes increasingly critical. The techniques developed and tested in this study support more sophisticated, adaptive responses to cyber threats, empowering security professionals with tools that are both reactive and preemptively adaptive.

Future work will explore the impact of the computational cost and efficiency of the augmentation methods, providing deeper insights into their practical applications. Further analysis on tailored cost-sensitive learning strategies will be pursued, where the cost of misclassifying minority classes is set higher than that of the majority classes to compel the model to pay more attention to the underrepresented classes. These measures are crucial for building robust models that perform well across all categories and are not biased toward the majority. Finally, more complex mechanisms of explanation exploiting consolidated techniques, such as LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations), will be included to acquire a deeper understanding of cyberattacks.

Author Contributions: Conceptualization, K.M. and E.B.; methodology, A.G.; software, E.B.; validation, K.M., T.T. and A.G.; formal analysis, E.B.; investigation, T.T.; resources, T.T.; data curation, K.M.; writing—original draft preparation, K.M.; writing—review and editing, E.B.; visualization, E.B.; supervision, E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The original data presented in this study, including the software to reproduce the results, are openly available at https://github.com/EBarbierato/cyberattack_classification (accessed on 3 July 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism
ADASYN	Adaptive Synthetic Sampling
CNN	Convolutional Neural Networks
DL	Deep Learning
DDoS	Distributed Denial of Service
DT	Decision Trees
GBM	Gradient Boosting Machines
IDS	Intrusion Detection Systems
KNN	K-Nearest Neighbor
KPCA	Kernel Principal Component Analysis
LSTM	Long Short-Term Memory
PCA	Principal Component Analysis
RNN	Recurrent Neural Networks
RF	Random Forest
SMOTE	Synthetic Minority Over-sampling Technique
SMOTEENN	Combination of SMOTE and Edited Nearest Neighbors

SVM	Support Vector Machines
XGB	eXtreme Gradient Boost

Appendix A

Appendix A.1. Additional Details

This section details the precision achieved when predicting the values of the techniques_mitre distribution.

Appendix A.1.1. GBM

SMOTE

Table A1. Accuracy: 0.8420.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.68	0.71	0.69	23,003
benign	0.55	0.47	0.51	23,309
domain_trust_discovery	0.54	0.54	0.54	23,141
group_policy_discovery	0.97	0.96	0.96	23,008
network_service_discovery	0.95	0.97	0.96	23,029
reconnaissance_scan_ip_blocks	0.98	0.98	0.98	23,036
reconnaissance_vulnerability_scanning	0.90	0.98	0.94	22,971
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,201
remote_system_discovery	0.99	0.99	0.99	23,156

ADASYN

Table A2. Accuracy: 0.9402.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.96	0.99	0.98	23,170
benign	0.93	0.55	0.69	23,287
domain_trust_discovery	0.93	0.97	0.95	23,051
group_policy_discovery	0.97	0.99	0.98	22,987
network_service_discovery	1.00	1.00	1.00	22,985
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,138
reconnaissance_vulnerability_scanning	0.75	0.97	0.85	23,251
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,209
remote_system_discovery	0.97	0.99	0.98	22,924

Borderline SMOTE

Table A3. Accuracy: 0.9539.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.99	1.00	1.00	23,067
benign	0.96	0.62	0.75	23,253
domain_trust_discovery	0.90	0.99	0.94	22,971
group_policy_discovery	1.00	1.00	1.00	23,220
network_service_discovery	1.00	1.00	1.00	22,987
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,023
reconnaissance_vulnerability_scanning	0.79	0.98	0.87	22,968
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	1.00	1.00	1.00	23,163

Tomek Links

Table A4. Accuracy: 0.9895.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.40	0.29	0.33	7
benign	0.97	0.99	0.98	9535
domain_trust_discovery	0.00	0.00	0.00	58
group_policy_discovery	1.00	0.25	0.40	4
network_service_discovery	1.00	1.00	1.00	23,225
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	16
reconnaissance_vulnerability_scanning	0.59	0.43	0.50	192
reconnaissance_wordlist_scanning	1.00	1.00	1.00	115
remote_system_discovery	0.00	0.00	0.00	88

SMOTEENN

Table A5. Accuracy: 0.9892.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.98	1.00	0.99	22,784
benign	0.98	0.93	0.96	21,615
domain_trust_discovery	1.00	0.99	0.99	22,536
group_policy_discovery	0.99	0.99	0.99	23,026
network_service_discovery	1.00	1.00	1.00	22,956
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,013
reconnaissance_vulnerability_scanning	0.96	1.00	0.98	22,359
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,256
remote_system_discovery	1.00	1.00	1.00	22,643

SMOTETomek

Table A6. Accuracy: 0.9842.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.97	1.00	0.98	23,164
benign	0.98	0.90	0.94	22,718
domain_trust_discovery	1.00	0.98	0.99	22,862
group_policy_discovery	0.99	0.99	0.99	23,048
network_service_discovery	1.00	1.00	1.00	23,213
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,065
reconnaissance_vulnerability_scanning	0.94	1.00	0.97	22,871
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	1.00	0.99	0.99	23,077

Appendix A.1.2. KNN

SMOTE

Table A7. Accuracy: 0.8245.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.59	0.64	0.61	23,003
benign	0.46	0.39	0.42	23,309
domain_trust_discovery	0.50	0.51	0.50	23,141
group_policy_discovery	0.97	0.97	0.97	23,008
network_service_discovery	0.97	0.98	0.97	23,029
reconnaissance_scan_ip_blocks	0.98	0.98	0.98	23,036
reconnaissance_vulnerability_scanning	0.95	0.97	0.96	22,971

Table A7. *Cont.*

Techniques_Mitre	Precision	Recall	F1-Score	Support
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,201
remote_system_discovery	0.99	0.99	0.99	23,156

ADASYN

Table A8. Accuracy: 0.9782.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.98	1.00	0.99	23,170
benign	0.96	0.85	0.90	23,287
domain_trust_discovery	0.94	0.99	0.96	23,051
group_policy_discovery	1.00	1.00	1.00	22,987
network_service_discovery	1.00	1.00	1.00	22,985
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,138
reconnaissance_vulnerability_scanning	0.94	0.98	0.96	23,251
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,209
remote_system_discovery	0.99	1.00	1.00	22,924

Borderline SMOTE

Table A9. Accuracy: 0.9812.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.99	1.00	1.00	23,067
benign	0.95	0.88	0.91	23,253
domain_trust_discovery	0.92	0.98	0.95	22,971
group_policy_discovery	1.00	1.00	1.00	23,220
network_service_discovery	1.00	1.00	1.00	22,987
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,023
reconnaissance_vulnerability_scanning	0.97	0.98	0.98	22,968
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	1.00	1.00	1.00	23,163

Tomek Links

Table A10. Accuracy: 0.9925.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.75	0.43	0.55	7
benign	0.99	0.99	0.99	9535
domain_trust_discovery	0.69	0.84	0.76	58
group_policy_discovery	0.33	0.25	0.29	4
network_service_discovery	1.00	1.00	1.00	23,225
reconnaissance_scan_ip_blocks	0.93	0.88	0.90	16
reconnaissance_vulnerability_scanning	0.59	0.45	0.51	192
reconnaissance_wordlist_scanning	1.00	0.99	1.00	115
remote_system_discovery	0.75	0.92	0.83	88

SMOTEENN

Table A11. Accuracy: 0.9938.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.98	1.00	0.99	22,784
benign	0.99	0.96	0.97	21,615

Table A11. *Cont.*

Techniques_Mitre	Precision	Recall	F1-Score	Support
domain_trust_discovery	1.00	0.99	1.00	22,536
group_policy_discovery	0.99	1.00	1.00	23,026
network_service_discovery	1.00	1.00	1.00	22,956
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,013
reconnaissance_vulnerability_scanning	0.99	1.00	0.99	22,359
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,256
remote_system_discovery	0.99	1.00	1.00	22,643

SMOTETomek

Table A12. Accuracy: 0.9902.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.97	0.99	0.98	23,164
benign	0.98	0.94	0.96	22,718
domain_trust_discovery	0.99	0.99	0.99	22,862
group_policy_discovery	0.99	1.00	1.00	23,048
network_service_discovery	1.00	1.00	1.00	23,213
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,065
reconnaissance_vulnerability_scanning	0.98	1.00	0.99	22,871
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	0.99	1.00	0.99	23,077

Appendix A.1.3. Logistic Regressor

SMOTE

Table A13. Accuracy: 0.7380.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.63	0.72	0.67	23,003
benign	0.35	0.12	0.18	23,309
domain_trust_discovery	0.47	0.17	0.25	23,141
group_policy_discovery	0.92	0.90	0.91	23,008
network_service_discovery	0.62	0.89	0.73	23,029
reconnaissance_scan_ip_blocks	0.88	0.96	0.92	23,036
reconnaissance_vulnerability_scanning	0.89	0.91	0.90	22,971
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,201
remote_system_discovery	0.61	0.97	0.75	23,156

ADASYN

Table A14. Accuracy: 0.7414.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.64	0.47	0.54	23,170
benign	0.74	0.16	0.26	23,287
domain_trust_discovery	0.88	0.85	0.87	23,051
group_policy_discovery	0.54	0.63	0.58	22,987
network_service_discovery	1.00	1.00	1.00	22,985
reconnaissance_scan_ip_blocks	0.90	0.98	0.94	23,138
reconnaissance_vulnerability_scanning	0.72	0.96	0.82	23,251
reconnaissance_wordlist_scanning	0.99	1.00	0.99	23,209
remote_system_discovery	0.42	0.64	0.51	22,924

Borderline SMOTE

Table A15. Accuracy: 0.8470.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.84	0.69	0.76	23,067
benign	0.84	0.23	0.36	23,253
domain_trust_discovery	0.84	0.94	0.89	22,971
group_policy_discovery	0.84	0.92	0.88	23,220
network_service_discovery	1.00	1.00	1.00	22,987
reconnaissance_scan_ip_blocks	0.96	1.00	0.98	23,023
reconnaissance_vulnerability_scanning	0.75	0.96	0.84	22,968
reconnaissance_wordlist_scanning	0.99	1.00	1.00	23,202
remote_system_discovery	0.65	0.88	0.75	23,163

Tomek Links

Table A16. Accuracy: 0.8070.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.02	0.43	0.03	7
benign	0.99	0.34	0.50	9535
domain_trust_discovery	0.20	0.86	0.32	58
group_policy_discovery	0.00	0.50	0.00	4
network_service_discovery	1.00	1.00	1.00	23,225
reconnaissance_scan_ip_blocks	0.01	1.00	0.03	16
reconnaissance_vulnerability_scanning	0.15	0.96	0.25	192
reconnaissance_wordlist_scanning	0.98	0.99	0.99	115
remote_system_discovery	0.03	0.97	0.06	88

SMOTEENN

Table A17. Accuracy: 0.8609.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.84	0.58	0.69	22,784
benign	0.76	0.46	0.57	21,615
domain_trust_discovery	0.96	0.93	0.95	22,536
group_policy_discovery	0.75	0.83	0.79	23,026
network_service_discovery	1.00	1.00	1.00	22,956
reconnaissance_scan_ip_blocks	0.93	0.99	0.96	23,013
reconnaissance_vulnerability_scanning	0.94	0.96	0.95	22,359
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,256
remote_system_discovery	0.64	0.98	0.78	22,643

SMOTETomek

Table A18. Accuracy: 0.8512.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.84	0.57	0.67	23,164
benign	0.74	0.42	0.54	22,718
domain_trust_discovery	0.95	0.93	0.94	22,862
group_policy_discovery	0.73	0.83	0.78	23,048
network_service_discovery	1.00	1.00	1.00	23,213
reconnaissance_scan_ip_blocks	0.92	0.99	0.95	23,065
reconnaissance_vulnerability_scanning	0.92	0.95	0.94	22,871
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	0.64	0.98	0.77	23,077

Appendix A.1.4. LSTM

SMOTE

Table A19. Accuracy: 0.8194.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.68	0.70	0.69	23,003
benign	0.51	0.42	0.46	23,309
domain_trust_discovery	0.52	0.46	0.49	23,141
group_policy_discovery	0.95	0.93	0.94	23,008
network_service_discovery	0.91	0.96	0.93	23,029
reconnaissance_scan_ip_blocks	0.97	0.98	0.97	23,036
reconnaissance_vulnerability_scanning	0.90	0.98	0.94	22,971
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,201
remote_system_discovery	0.85	0.97	0.90	23,156

ADASYN

Table A20. Accuracy: 0.8754.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.88	0.77	0.82	23,170
benign	0.93	0.37	0.53	23,287
domain_trust_discovery	0.91	0.92	0.92	23,051
group_policy_discovery	0.92	0.93	0.92	22,987
network_service_discovery	1.00	1.00	1.00	22,985
reconnaissance_scan_ip_blocks	0.97	1.00	0.99	23,138
reconnaissance_vulnerability_scanning	0.71	0.99	0.83	23,251
reconnaissance_wordlist_scanning	1.00	0.98	0.99	23,209
remote_system_discovery	0.70	0.94	0.80	22,924

Borderline SMOTE

Table A21. Accuracy: 0.9166.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.97	0.87	0.92	23,067
benign	0.81	0.45	0.58	23,253
domain_trust_discovery	0.86	0.98	0.92	22,971
group_policy_discovery	0.97	1.00	0.98	23,220
network_service_discovery	1.00	1.00	1.00	22,987
reconnaissance_scan_ip_blocks	0.98	0.99	0.99	23,023
reconnaissance_vulnerability_scanning	0.75	0.99	0.85	22,968
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,202
remote_system_discovery	0.93	0.99	0.96	23,163

Tomek Links

Table A22. Accuracy: 0.9825.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.00	0.00	0.00	7
benign	0.95	0.99	0.97	9535
domain_trust_discovery	0.00	0.00	0.00	58
group_policy_discovery	0.00	0.00	0.00	4
network_service_discovery	1.00	0.99	1.00	23,225
reconnaissance_scan_ip_blocks	0.00	0.00	0.00	16
reconnaissance_vulnerability_scanning	0.00	0.00	0.00	192

Table A22. *Cont.*

Techniques_Mitre	Precision	Recall	F1-Score	Support
reconnaissance_wordlist_scanning	0.90	0.99	0.94	115
remote_system_discovery	0.00	0.00	0.00	88

SMOTEENN

Table A23. Accuracy: 0.9459.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.89	0.85	0.87	22,784
benign	0.95	0.80	0.87	21,615
domain_trust_discovery	0.99	0.95	0.97	22,536
group_policy_discovery	0.94	0.95	0.95	23,026
network_service_discovery	1.00	1.00	1.00	22,956
reconnaissance_scan_ip_blocks	0.99	0.99	0.99	23,013
reconnaissance_vulnerability_scanning	0.95	0.99	0.97	22,359
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,256
remote_system_discovery	0.83	0.98	0.90	22,643

SMOTETomek

Table A24. Accuracy: 0.9447.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.93	0.81	0.87	23,164
benign	0.90	0.81	0.85	22,718
domain_trust_discovery	0.98	0.96	0.97	22,862
group_policy_discovery	0.95	0.96	0.96	23,048
network_service_discovery	1.00	1.00	1.00	23,213
reconnaissance_scan_ip_blocks	0.98	1.00	0.99	23,065
reconnaissance_vulnerability_scanning	0.93	0.99	0.96	22,871
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	0.84	0.98	0.90	23,077

Appendix A.1.5. Naïve Bayes

SMOTE

Table A25. Accuracy: 0.4979.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.06	0.00	0.00	23,003
benign	0.14	0.00	0.00	23,309
domain_trust_discovery	0.23	0.02	0.03	23,141
group_policy_discovery	0.68	0.87	0.77	23,008
network_service_discovery	0.19	0.22	0.20	23,029
reconnaissance_scan_ip_blocks	0.73	0.48	0.57	23,036
reconnaissance_vulnerability_scanning	0.35	0.96	0.51	22,971
reconnaissance_wordlist_scanning	0.97	0.99	0.98	23,201
remote_system_discovery	0.46	0.94	0.62	23,156

ADASYN

Table A26. Accuracy: 0.4532.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.35	0.11	0.17	23,170
benign	0.31	0.04	0.07	23,287
domain_trust_discovery	0.85	0.87	0.86	23,051
group_policy_discovery	0.54	0.06	0.11	22,987
network_service_discovery	0.98	1.00	0.99	22,985
reconnaissance_scan_ip_blocks	0.21	0.95	0.35	23,138
reconnaissance_vulnerability_scanning	0.69	0.23	0.34	23,251
reconnaissance_wordlist_scanning	0.73	0.61	0.66	23,209
remote_system_discovery	0.29	0.22	0.25	22,924

Borderline SMOTE

Table A27. Accuracy: 0.6020.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.60	0.17	0.26	23,067
benign	0.28	0.04	0.07	23,253
domain_trust_discovery	0.86	0.93	0.89	22,971
group_policy_discovery	0.98	0.16	0.28	23,220
network_service_discovery	0.98	1.00	0.99	22,987
reconnaissance_scan_ip_blocks	0.32	1.00	0.49	23,023
reconnaissance_vulnerability_scanning	0.71	0.70	0.71	22,968
reconnaissance_wordlist_scanning	0.86	0.76	0.81	23,202
remote_system_discovery	0.49	0.66	0.56	23,163

Tomek Links

Table A28. Accuracy: 0.7185.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.00	0.14	0.00	7
benign	0.98	0.03	0.06	9535
domain_trust_discovery	0.12	0.86	0.21	58
group_policy_discovery	0.06	0.25	0.10	4
network_service_discovery	0.99	1.00	0.99	23,225
reconnaissance_scan_ip_blocks	0.00	0.81	0.01	16
reconnaissance_vulnerability_scanning	0.04	0.97	0.07	192
reconnaissance_wordlist_scanning	0.93	0.99	0.96	115
remote_system_discovery	0.34	0.94	0.50	88

SMOTEENN

Table A29. Accuracy: 0.6688.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.40	0.35	0.37	22,784
benign	0.34	0.03	0.06	21,615
domain_trust_discovery	0.90	0.90	0.90	22,536
group_policy_discovery	0.81	0.07	0.13	23,026
network_service_discovery	0.99	1.00	0.99	22,956
reconnaissance_scan_ip_blocks	0.36	0.92	0.51	23,013
reconnaissance_vulnerability_scanning	0.70	0.74	0.72	22,359
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,256
remote_system_discovery	0.79	0.98	0.88	22,643

SMOTETomek

Table A30. Accuracy: 0.6599.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.40	0.35	0.37	23,164
benign	0.33	0.03	0.05	22,718
domain_trust_discovery	0.88	0.90	0.89	22,862
group_policy_discovery	0.81	0.08	0.14	23,048
network_service_discovery	0.99	1.00	0.99	23,213
reconnaissance_scan_ip_blocks	0.34	0.91	0.50	23,065
reconnaissance_vulnerability_scanning	0.68	0.74	0.71	22,871
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,202
remote_system_discovery	0.80	0.94	0.87	23,077

Appendix A.1.6. Random Forest

SMOTE

Table A31. Accuracy: 0.8338.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.62	0.62	0.62	23,003
benign	0.45	0.42	0.43	23,309
domain_trust_discovery	0.58	0.61	0.59	23,141
group_policy_discovery	0.97	0.97	0.97	23,008
network_service_discovery	0.97	0.96	0.97	23,029
reconnaissance_scan_ip_blocks	0.97	0.97	0.97	23,036
reconnaissance_vulnerability_scanning	0.94	0.98	0.96	22,971
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,201
remote_system_discovery	1.00	1.00	1.00	23,156

ADASYN

Table A32. Accuracy: 0.9856.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	1.00	1.00	1.00	23,170
benign	0.97	0.90	0.93	23,287
domain_trust_discovery	0.94	0.98	0.96	23,051
group_policy_discovery	1.00	1.00	1.00	22,987
network_service_discovery	1.00	1.00	1.00	22,985
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,138
reconnaissance_vulnerability_scanning	0.96	0.99	0.98	23,251
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,209
remote_system_discovery	1.00	1.00	1.00	22,924

Borderline SMOTE

Table A33. Accuracy: 0.9856.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	1.00	1.00	1.00	23,067
benign	0.98	0.89	0.93	23,253
domain_trust_discovery	0.91	0.98	0.95	22,971
group_policy_discovery	1.00	1.00	1.00	23,220
network_service_discovery	1.00	1.00	1.00	22,987
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,023
reconnaissance_vulnerability_scanning	0.98	1.00	0.99	22,968

Table A33. *Cont.*

Techniques_Mitre	Precision	Recall	F1-Score	Support
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	1.00	1.00	1.00	23,163

Tomek Links

Table A34. Accuracy: 0.9947.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.75	0.43	0.55	7
benign	0.99	0.99	0.99	9535
domain_trust_discovery	0.74	0.74	0.74	58
group_policy_discovery	1.00	0.25	0.40	4
network_service_discovery	1.00	1.00	1.00	23,225
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	16
reconnaissance_vulnerability_scanning	0.72	0.52	0.61	192
reconnaissance_wordlist_scanning	1.00	1.00	1.00	115
remote_system_discovery	1.00	0.98	0.99	88

SMOTEENN

Table A35. Accuracy: 0.9981.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	1.00	1.00	1.00	22,784
benign	1.00	0.98	0.99	21,615
domain_trust_discovery	1.00	1.00	1.00	22,536
group_policy_discovery	1.00	1.00	1.00	23,026
network_service_discovery	1.00	1.00	1.00	22,956
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,013
reconnaissance_vulnerability_scanning	0.99	1.00	0.99	22,359
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,256
remote_system_discovery	1.00	1.00	1.00	22,643

SMOTETomek

Table A36. Accuracy: 0.9966.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	1.00	1.00	1.00	23,164
benign	1.00	0.97	0.98	22,718
domain_trust_discovery	1.00	1.00	1.00	22,862
group_policy_discovery	1.00	1.00	1.00	23,048
network_service_discovery	1.00	1.00	1.00	23,213
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,065
reconnaissance_vulnerability_scanning	0.98	1.00	0.99	22,871
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	1.00	1.00	1.00	23,077

Appendix A.1.7. RNN

SMOTE

Table A37. Accuracy: 0.7594.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.69	0.68	0.69	23,003
benign	0.48	0.08	0.14	23,309
domain_trust_discovery	0.47	0.50	0.48	23,141
group_policy_discovery	0.96	0.87	0.92	23,008
network_service_discovery	0.71	0.91	0.80	23,029
reconnaissance_scan_ip_blocks	0.96	0.86	0.91	23,036
reconnaissance_vulnerability_scanning	0.80	0.97	0.88	22,971
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,201
remote_system_discovery	0.66	0.97	0.79	23,156

ADASYN

Table A38. Accuracy: 0.8231.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.80	0.68	0.74	23,170
benign	0.92	0.29	0.44	23,287
domain_trust_discovery	0.91	0.89	0.90	23,051
group_policy_discovery	0.78	0.93	0.85	22,987
network_service_discovery	1.00	0.99	0.99	22,985
reconnaissance_scan_ip_blocks	0.96	0.85	0.90	23,138
reconnaissance_vulnerability_scanning	0.69	0.95	0.80	23,251
reconnaissance_wordlist_scanning	0.94	1.00	0.97	23,209
remote_system_discovery	0.62	0.83	0.71	22,924

Borderline SMOTE

Table A39. Accuracy: 0.8881.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.95	0.77	0.85	23,067
benign	0.93	0.34	0.50	23,253
domain_trust_discovery	0.86	0.96	0.91	22,971
group_policy_discovery	0.97	0.98	0.97	23,220
network_service_discovery	1.00	1.00	1.00	22,987
reconnaissance_scan_ip_blocks	0.90	0.99	0.94	23,023
reconnaissance_vulnerability_scanning	0.74	0.99	0.84	22,968
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,202
remote_system_discovery	0.78	0.99	0.87	23,163

Tomek Links

Table A40. Accuracy: 0.9790.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.00	0.00	0.00	7
benign	0.95	0.98	0.97	9535
domain_trust_discovery	0.00	0.00	0.00	58
group_policy_discovery	0.00	0.00	0.00	4
network_service_discovery	0.99	0.99	0.99	23,225
reconnaissance_scan_ip_blocks	0.00	0.00	0.00	16
reconnaissance_vulnerability_scanning	0.00	0.00	0.00	192

Table A40. *Cont.*

Techniques_Mitre	Precision	Recall	F1-Score	Support
reconnaissance_wordlist_scanning	0.80	0.98	0.88	115
remote_system_discovery	0.00	0.00	0.00	88

SMOTEENN

Table A41. Accuracy: 0.6472.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.79	0.31	0.45	22,784
benign	0.25	0.65	0.36	21,615
domain_trust_discovery	0.89	0.94	0.91	22,536
group_policy_discovery	0.88	0.58	0.70	23,026
network_service_discovery	1.00	0.96	0.98	22,956
reconnaissance_scan_ip_blocks	0.47	0.92	0.62	23,013
reconnaissance_vulnerability_scanning	0.98	0.46	0.62	22,359
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,256
remote_system_discovery	0.00	0.00	0.00	22,643

SMOTETomek

Table A42. Accuracy: 0.7977.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.69	0.34	0.45	23,164
benign	0.57	0.61	0.59	22,718
domain_trust_discovery	0.99	0.93	0.96	22,862
group_policy_discovery	0.84	0.59	0.69	23,048
network_service_discovery	1.00	0.99	0.99	23,213
reconnaissance_scan_ip_blocks	0.85	0.89	0.87	23,065
reconnaissance_vulnerability_scanning	0.89	0.86	0.87	22,871
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,202
remote_system_discovery	0.55	0.98	0.70	23,077

Appendix A.1.8. XGB

SMOTE

Table A43. Accuracy: 0.8387.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.69	0.69	0.69	23,003
benign	0.55	0.49	0.51	23,309
domain_trust_discovery	0.53	0.53	0.53	23,141
group_policy_discovery	0.96	0.95	0.95	23,008
network_service_discovery	0.93	0.97	0.95	23,029
reconnaissance_scan_ip_blocks	0.98	0.98	0.98	23,036
reconnaissance_vulnerability_scanning	0.90	0.98	0.94	22,971
reconnaissance_wordlist_scanning	0.99	0.99	0.99	23,201
remote_system_discovery	0.99	0.98	0.98	23,156

ADASYN

Table A44. Accuracy: 0.9254.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.93	0.98	0.96	23,170
benign	0.96	0.45	0.61	23,287
domain_trust_discovery	0.92	0.95	0.93	23,051
group_policy_discovery	0.96	0.98	0.97	22,987
network_service_discovery	1.00	1.00	1.00	22,985
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,138
reconnaissance_vulnerability_scanning	0.72	0.99	0.84	23,251
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,209
remote_system_discovery	0.94	0.98	0.96	22,924

Borderline SMOTE

Table A45. Accuracy: 0.9427.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.98	1.00	0.99	23,067
benign	0.96	0.51	0.67	23,253
domain_trust_discovery	0.87	0.99	0.92	22,971
group_policy_discovery	1.00	1.00	1.00	23,220
network_service_discovery	1.00	1.00	1.00	22,987
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,023
reconnaissance_vulnerability_scanning	0.75	1.00	0.86	22,968
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	1.00	1.00	1.00	23,163

Tomek Links

Table A46. Accuracy: 0.9932.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	1.00	0.29	0.44	7
benign	0.98	0.99	0.99	9535
domain_trust_discovery	0.64	0.88	0.74	58
group_policy_discovery	0.00	0.00	0.00	4
network_service_discovery	1.00	1.00	1.00	23,225
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	16
reconnaissance_vulnerability_scanning	0.63	0.33	0.44	192
reconnaissance_wordlist_scanning	1.00	1.00	1.00	115
remote_system_discovery	0.99	0.97	0.98	88

SMOTEENN

Table A47. Accuracy: 0.9819.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.97	0.98	0.98	22,784
benign	0.96	0.90	0.93	21,615
domain_trust_discovery	1.00	0.97	0.98	22,536
group_policy_discovery	0.97	0.99	0.98	23,026
network_service_discovery	1.00	1.00	1.00	22,956
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,013
reconnaissance_vulnerability_scanning	0.95	1.00	0.97	22,359
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,256
remote_system_discovery	0.99	0.99	0.99	22,643

SMOTETomek

Table A48. Accuracy: 0.9771.

Techniques_Mitre	Precision	Recall	F1-Score	Support
account_discovery_domain	0.95	0.99	0.97	23,164
benign	0.96	0.87	0.91	22,718
domain_trust_discovery	0.99	0.97	0.98	22,862
group_policy_discovery	0.97	0.98	0.98	23,048
network_service_discovery	1.00	1.00	1.00	23,213
reconnaissance_scan_ip_blocks	1.00	1.00	1.00	23,065
reconnaissance_vulnerability_scanning	0.93	1.00	0.96	22,871
reconnaissance_wordlist_scanning	1.00	1.00	1.00	23,202
remote_system_discovery	0.99	0.99	0.99	23,077

References

1. Apruzzese, G.; Laskov, P.; Montes de Oca, E.; Mallouli, W.; Brdalo Rapa, L.; Grammatopoulos, A.V.; Di Franco, F. The role of machine learning in cybersecurity. *Digit. Threat. Res. Pract.* **2023**, *4*, 1–38. [CrossRef]
2. Mijwil, M.; Salem, I.E.; Ismaeel, M.M. The significance of machine learning and deep learning techniques in cybersecurity: A comprehensive review. *Iraqi J. Comput. Sci. Math.* **2023**, *4*, 87–101.
3. Bagui, S.; Mink, D.; Bagui, S.; Ghosh, T.; McElroy, T.; Paredes, E.; Khasnavis, N.; Plenkers, R. Detecting reconnaissance and discovery tactics from the MITRE ATT&CK framework in Zeek conn logs using spark's machine learning in the big data framework. *Sensors* **2022**, *22*, 7999. [CrossRef] [PubMed]
4. Tufan, E.; Tezcan, C.; Acartürk, C. Anomaly-based intrusion detection by machine learning: A case study on probing attacks to an institutional network. *IEEE Access* **2021**, *9*, 50078–50092. [CrossRef]
5. Ravi, V.; Chaganti, R.; Alazab, M. Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Comput. Electr. Eng.* **2022**, *102*, 108156. [CrossRef]
6. Verkerken, M.; D'hooge, L.; Wauters, T.; Volckaert, B.; De Turck, F. Towards model generalization for intrusion detection: Unsupervised machine learning techniques. *J. Netw. Syst. Manag.* **2022**, *30*, 1–25. [CrossRef]
7. Hwang, R.H.; Peng, M.C.; Huang, C.W.; Lin, P.C.; Nguyen, V.L. An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access* **2020**, *8*, 30387–30399. [CrossRef]
8. Aamir, M.; Zaidi, S.M.A. Clustering based semi-supervised machine learning for DDoS attack classification. *J. King Saud-Univ.-Comput. Inf. Sci.* **2021**, *33*, 436–446. [CrossRef]
9. Maharana, K.; Mondal, S.; Nemade, B. A review: Data pre-processing and data augmentation techniques. *Glob. Transitions Proc.* **2022**, *3*, 91–99. [CrossRef]
10. Naik, B.; Mehta, A.; Yagnik, H.; Shah, M. The impacts of artificial intelligence techniques in augmentation of cybersecurity: A comprehensive review. *Complex Intell. Syst.* **2022**, *8*, 1763–1780. [CrossRef]
11. Agrawal, G.; Kaur, A.; Myneni, S. A review of generative models in generating synthetic attack data for cybersecurity. *Electronics* **2024**, *13*, 322. [CrossRef]
12. Mohammad, R.; Saeed, F.; Almazroi, A.A.; Alsubaei, F.S.; Almazroi, A.A. Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach. *Systems* **2024**, *12*, 79. [CrossRef]
13. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
14. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; IEEE: Piscataway Township, NJ, USA, 2008; pp. 1322–1328.
15. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Proceedings of the International Conference on Intelligent Computing, Hefei, China, 23–26 August 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887.
16. Swana, E.F.; Doorsamy, W.; Bokoro, P. Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset. *Sensors* **2022**, *22*, 3246. [CrossRef] [PubMed]
17. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
18. Yang, H.; Li, M. Software Defect Prediction Based on SMOTE-Tomek and XGBoost. In *Bio-Inspired Computing: Theories and Applications*; Pan, L., Cui, Z., Cai, J., Li, L., Eds.; Springer: Singapore, 2022; pp. 12–31.
19. Handa, A.; Sharma, A.; Shukla, S.K. Machine learning in cybersecurity: A review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**, *9*, e1306. [CrossRef]
20. Dasgupta, D.; Akhtar, Z.; Sen, S. Machine learning in cybersecurity: A comprehensive survey. *J. Def. Model. Simul.* **2022**, *19*, 57–106. [CrossRef]

21. Martínez Torres, J.; Iglesias Comesaña, C.; García-Nieto, P.J. Machine learning techniques applied to cybersecurity. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2823–2836. [CrossRef]
22. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 6639–6649.