IJWR

A Combined Approach Of Adasyn And Tomeklink For Anomaly Network Intrusion Detection System Using Some Selected Machine Learning Algorithms

Nasiru Ige Salihu[a], Muhammed Nazeer Musa[b], Awujola J. Olalekan[c]

[a]Directorate of Information and Communication Technology, Nigerian Defence Academy, Nigerian Defence Academy, Kaduna, Nigeria; nasiru4real@gmail.com

[b]Cyber Security Department. Nigerian Defence Academy, Nigerian Defence Academy, Kaduna, Nigeria; muhammadmusa2502@nda.edu.ng

[c]Computer Science Department, Nigerian Defence Academy, Nigerian Defence Academy, Kaduna, Nigeria; ojawujoola@nda.edu.ng

ABSTRACT

Securing computer networks against malicious attacks requires an efficient Network Intrusion Detection System (IDS). While machine learning techniques are commonly used for anomaly-based intrusion detection, data imbalance challenges conventional algorithms, leading to biased predictions and reduced accuracy. This study introduces a novel approach that combines ADASYN and Tomek links to address this issue, along with specific machine learning algorithms. ADASYN generates synthetic samples for the minority class to achieve dataset balance, and Tomek links eliminate redundant instances from the majority class. Four supervised machine learning algorithms (Random Forest, J48, Multilayer Perceptron, and Bagging) were assessed on both imbalanced and balanced datasets. Results show Random Forest exhibited 99.67% accuracy, while J48 and Bagging yielded 99.30%, and MLP recorded 98.53%. Notably, Random Forest emerges as a highly effective algorithm for Intrusion Detection, demonstrating flawless accuracy with balanced data. These outcomes highlight the proposed approach's ability to enhance prediction accuracy in network intrusion detection compared to imbalanced datasets, validated through a comparative analysis with state-of-the-art solutions.

Keywords— Network Intrusion Detection System, Machine Learning, Data Imbalance, Adasyn, Tomek Links, Intrusion Detection System.

1. Introduction

In today's interconnected world, the rapid growth of cyber threats poses significant challenges to the security of computer networks [1]. Cyberattacks ranging from data breaches to system disruptions, can have severe consequences for individuals, organizations, and even nations. As a result, there is a pressing need for effective cybersecurity measures to detect and mitigate these threats. One crucial aspect of cybersecurity is the development of robust intrusion detection systems (IDS) capable of identifying anomalous activities in network traffic.

IDSs play a critical role in monitoring network traffic and identifying potential intrusions. They analyze data packets and network behavior to distinguish between normal and malicious activities [1]. Traditional IDSs fall into two main categories: signature-based detection (also known as misuse detection) and anomaly detection [1]. Signature-based detection relies on known patterns of attacks and can effectively identify previously documented threats. However, it struggles to detect unknown or novel attacks that lack predefined signatures. On the other hand, anomaly detection focuses on detecting deviations from the expected behavior of a network.

http://dx.doi.org/10.22133/ijwr.2024.486081.1245

Citation N. Ige Salihu, M. Nazeer Musa, A. J. Olalekan, "A Combined Approach Of Adasyn And Tomeklink For Anomaly Network Intrusion Detection System Using Some Selected Machine Learning Algorithms", *International Journal of Web Research*, vol.7, no.4, pp.51-64, 2024, doi: http://dx.doi.org/10.22133/ijwr.2024.486081.1245.

**Coressponding Author*

Article History: Received: 29 May 2024; Revised: 8 September 2024; Accepted: 16 September 2024.

It builds models of normal network behavior and identifies activities that significantly deviate from this baseline [1]. While anomaly detection has the potential to detect unknown attacks, it often generates a high number of false positives, which can overwhelm administrators and impact the efficiency of IDSs [2].

To address the limitations of anomaly-based IDSs, several studies have investigated the use of machine learning techniques for improving intrusion detection accuracy. Machine learning algorithms have the ability to learn from data patterns and make predictions or decisions without explicit programming. This capability makes them well-suited for identifying complex and evolving network intrusions.

The advantage of using machine learning algorithms for anomaly-based detection is their ability to analyze vast amounts of network traffic data and identify patterns that distinguish normal behavior from abnormal behavior [3].

Furthermore, a set of selected machine learning algorithms will be employed for intrusion detection. These algorithms include Random Forest, J48, Multilayer perceptron (MLP), and Bagging will be evaluated on KDD-Cup dataset. Each algorithm will be trained and evaluated on the balanced dataset using appropriate performance metrics.

Data imbalance is a significant challenge in intrusion detection, as network intrusion events are often rare compared to normal network traffic. Imbalanced datasets can lead to biased models and reduced detection performance. Various approaches have been proposed to tackle this issue, including data balancing techniques. Two popular techniques for addressing data imbalance are ADASYN (Adaptive Synthetic Sampling) and Tomek link. ADASYN generates synthetic samples for minority classes, while Tomek link identifies and removes instances near class boundaries to enhance class separation [2].

This research aims to address the limitations of current anomaly-based IDSs and propose solutions to enhance their effectiveness using machine learning techniques by building an hybrid data class balancing technique of combining Adasyn and Tomek link method. The focus is not only on improving the accuracy of detection on known datasets but also on developing models that can be deployed on real networks without biasedness and false rate alarm that normally arise from class imbalance. This research contributes to knowledge by introducing a holistic, hybrid approach that enhances anomaly detection systems' accuracy, robustness, and efficiency. It advances the state-of-the-art by addressing gaps in handling imbalanced datasets, reducing false alarms, and ensuring scalability for modern network environments, ultimately strengthening cybersecurity practices.

2. Instrusion Detection System

Intrusion Detection System (IDS) refers to a series of devices or software that play an essential role in combating intrusions and malicious behavior in modern organizations. Its function is not to eliminate but to guard against network attacks. It determines whether anomalies occur by detecting traffic or logs. If there is any abnormality, it will send an alarm to the system's management unit [4]. Intrusion detection is the process of monitoring network traffic and computer activities to identify unauthorized or malicious actions. Any device or software designed for this purpose is referred to as an Intrusion Detection System (IDS). Figure 1 illustrates how an IDS analyzes monitored data and generates alerts based on its knowledge, which may include databases, statistical models, or artificial intelligence. These alerts are either communicated to an administrator or aggregated through a Security Information and Event Management (SIEM) system. A SIEM system enables real-time analysis of alerts from various sources, providing a unified and comprehensive overview of IT security.

IDSs are often mistaken for two other security tools: firewalls and Intrusion Prevention Systems (IPSs). While all three mechanisms aim to safeguard network systems, they employ different approaches. Firewalls primarily focus on external intrusions by analyzing packet headers and applying predetermined rules to filter incoming and outgoing traffic. In contrast, IDSs monitor activities within the protected network, extending beyond the perimeter. However, IDSs solely serve a monitoring role and require an administrator to process their alerts since they cannot directly block suspicious activities. IPSs, on the other hand, function as IDSs but possess the capability to proactively block detected threats. Nevertheless, this automation introduces complexities as improper responses can potentially disrupt network operations.

In the 1970s, the expansion of computer networks presented challenges regarding user activity monitoring and access control. In a publication by [5], the United States Air Forces (USAF) acknowledged the growing awareness of computer security issues affecting their operations

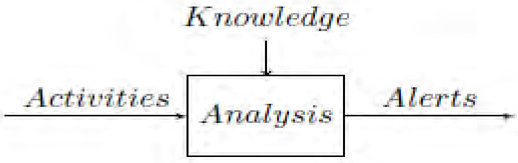


Figure. 1. Intrusion Detection System function.

and administration. The USAF faced unique difficulties as users with varying security clearances shared the same computer systems [6]. Anderson further outlined methods to enhance computer security threat monitoring and surveillance in another paper published in 1980 [5]. He introduced the concept of automating intrusion detection within a network to identify covert users. The proposed IDS aimed to assist administrators in reviewing system event logs, file access logs, and user access logs. A model for real-time intrusion detection was presented in 1986 [7]. This study draws on the development of a prototype known as the Intrusion Detection Expert System (IDES), created between 1984 and 1986. IDES combined a rule-based expert system for identifying known attacks with statistical anomaly detection applied to user and network data. The system produced alerts in a standardized format, ensuring interoperability with diverse systems.

2.1. Machine Learning

According to [8] Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment. They are considered the working horse in the new era of the so-called big data. Techniques based on machine learning have been applied successfully in diverse fields ranging from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, and computational biology to biomedical and medical applications.

Machine learning models can perform several tasks relevant to intrusion detection, including classification, regression, and reconstruction. Classification involves assigning inputs to categories such as "normal" or "attack" or distinguishing between various attack types. Regression, also called "prediction," estimates continuous values, such as the likelihood of an input being an attack. Reconstruction, often used in specific neural network architectures, compresses and decompresses input data to help the model learn underlying features and representations.

Machine learning algorithms are trained using either supervised or unsupervised methods, with some models, like neural networks, supporting both approaches. In supervised training, the algorithm learns from a dataset containing inputs paired with correct outputs, enabling it to model the relationship between them. Tasks such as classification and regression fall under supervised training. Conversely, unsupervised training works without labeled outputs, focusing on uncovering patterns or structures within the input data, with reconstruction being a common unsupervised task.

Once trained, machine learning models must undergo testing to assess their performance.

2.2. Dataset Description

Datasets play a crucial role in training machine learning algorithms, as they provide the necessary quantity and quality of data. The effectiveness of machine learning models heavily relies on the availability of high-quality data. However, acquiring such datasets can be challenging and expensive.

KDD Cup 99 Data set

The KDD Cup 99 dataset, also known as the 1999 DARPA Intrusion Detection Evaluation dataset, is a widely used benchmark dataset for evaluating intrusion detection systems. It was created for the KDD Cup 1999 data mining competition, which aimed to develop effective methods for detecting network intrusions.

The dataset contains network traffic data captured from a simulated military network environment. It includes both normal network traffic instances and various types of simulated attacks, such as Denial of Service (DoS), probing, and unauthorized access attempts. The dataset consists of approximately five million connection records, and each record is described by 41 features, including protocol type, service, source and destination addresses, and others, [9].

2.3. Data Balancing

Data balancing plays a crucial role in improving the performance, fairness, and generalization of machine learning models, especially in domains with imbalanced class distributions like intrusion detection. By addressing the imbalance issue, data balancing techniques help to ensure that the machine learning algorithms can learn from and effectively detect both majority and minority class instances, leading to more accurate and reliable predictions or classifications [1O].

The choice of data balancing technique depends on the specific characteristics of the dataset and the requirements of the problem at hand. It is important to carefully select and evaluate the appropriate technique to avoid introducing bias or overfitting [10].

ADASYN (Adaptive Synthetic Sampling)

ADASYN is an oversampling technique used to address class imbalance in machine learning. It is specifically designed to handle imbalanced datasets by generating synthetic samples for the minority class based on their density distribution in the feature space. The ADASYN algorithm adapts to the characteristics of the dataset, adjusting the synthesis of samples according to the density distribution. It provides a data-driven approach that can be effective in scenarios where the imbalance between classes is more complex and varies across the feature space. ADASYN can be particularly useful when used in conjunction with other techniques, such as Tomek links or undersampling methods, to create a more balanced dataset for training machine learning models. By generating synthetic samples based on the density distribution, ADASYN helps to alleviate the bias caused by imbalanced data and improve the model's ability to generalize and detect minority class instances accurately [17].

Tomek Links

Tomek links is an undersampling technique used to address the issue of class imbalance in machine learning. It focuses on identifying and removing instances that form pairs of samples from different classes and are located in close proximity to each other in the feature space. These pairs are known as Tomek links.

The concept behind Tomek links is that if two instances from different classes are very close to each other, they are likely to be near the decision boundary, making classification more challenging. By removing these instances, the aim is to enhance the separation between the classes and improve the performance of the machine learning algorithm.

Tomek links alone may not always achieve a perfect balance between classes, as they only focus on removing instances forming links. However, when combined with other data balancing techniques, such as oversampling methods, Tomek links can contribute to creating a more balanced and representative dataset for training the machine learning model.

2.4. Model Selection

The selection of appropriate machine learning algorithms is crucial for achieving accurate and effective network intrusion detection. This section outlines the model selection process for the research study.

Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to create a robust and accurate model. It leverages the concept of bagging and random feature selection to reduce overfitting and improve generalization [12].

J48 (C4.5 Decision Tree)

J48, also known as C4.5, is a widely used decision tree algorithm that employs the concept of information gain to split the dataset based on attribute values. It recursively builds a decision tree by selecting the most informative features and splitting the data at each node. J48 is known for its interpretability and simplicity, making it suitable for understanding the underlying patterns and rules in the data [13].

Multilayer Perceptron (MLP)

Multilayer Perceptron is a type of artificial neural network with multiple layers of interconnected nodes (neurons). It is capable of learning complex nonlinear relationships between input features and output classes. MLP utilizes backpropagation to adjust the weights and biases during training, allowing it to model intricate decision boundaries. MLP is known for its flexibility and ability to capture intricate patterns in the data [14].

Bagging

Bagging, short for bootstrap aggregating, is an ensemble learning technique that combines multiple models trained on different subsets of the training data. It reduces variance and improves robustness by averaging the predictions of individual models. Bagging can be applied to various base classifiers, including decision trees and random forests, to enhance their performance and mitigate overfitting [12].

2.5. Review of Related work

Machine learning techniques have been widely explored for intrusion detection in computer networks. A comprehensive review evaluated various algorithms, including decision trees, support vector machines (SVM), random forests, and neural networks, concluding that ensemble methods like AdaBoost and stacking improve the accuracy and robustness of intrusion detection systems [24]. Another study analyzed Intrusion Detection Systems (IDSs) using machine learning (ML) techniques, focusing on the KDD CUP-'99' dataset. The study highlighted the effectiveness of SVM, achieving an accuracy of 98.08% [26].

Similarly, a framework employing a Wrapper feature selection approach and a Bayesian classifier achieved an accuracy of 98.3% with a false positive rate of 0.7% [15].

To optimize performance, a hybrid anomaly detection model combined decision trees and k-NN algorithms. This approach utilized feature selection techniques to extract optimized information from the NSL-KDD dataset, achieving a detection accuracy of 99.7% with a false alarm rate of 0.2% [16]. Another model, based on stacking ensemble techniques, correctly identified several attack classes with an accuracy of 92.55% [17].

A hybrid model integrating probabilistic BayesNet and IBK was compared to traditional approaches like BayesNet, J48, JRip, IBK, and SMO. When tested on the KDDCUP99 dataset, the proposed method achieved a performance accuracy of 96.1% [16]. Enhanced Intrusion Detection Systems (IDSs) using feature selection methods and ensemble learning algorithms have also demonstrated significant improvements. For example, one study achieved an accuracy of

99.7984% using a reduced feature set of 19 attributes and product probability rules [18].

Deep learning models, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), have shown significant potential in detecting and classifying malware samples. However, these approaches require large and diverse datasets to ensure robustness and effectiveness [19]. Various evaluation metrics for intrusion detection, such as detection rate, false positive rate, precision, recall, and F1 score, were also examined. The importance of standardized evaluation metrics for fair comparisons and benchmarking was emphasized in recent reviews [20].

The application of data mining and machine learning techniques for intrusion detection has also been explored extensively. Comparisons of algorithms to analyze network traffic and detect potential intrusions. Common algorithms applied in this area include decision trees, support vector machines, and k-nearest neighbors, which help to efficiently handle large volumes of network data [21].

A deep learning-based hybrid approach combining convolutional neural networks (CNN) and bidirectional long short-term memory (BiLSTM) was proposed to enhance intrusion detection accuracy. [22] demonstrated its effectiveness using the NSL-KDD dataset, achieving a detection accuracy of 99.5% and showcasing its ability to handle complex attack patterns.

An ensemble-based intrusion detection model that integrates Random Forest and Gradient Boosting algorithms was developed to identify advanced persistent threats. Validated on the CICIDS2017 dataset, this model achieved a 98.9% accuracy rate [23].

To address computational inefficiencies, Recursive Feature Elimination (RFE) was combined with LightGBM to develop an optimized feature selection framework. This approach reduced computational overhead by 40% while achieving a detection efficiency of 96% on the UNSW-NB15 dataset [24].

Generative Adversarial Networks (GAN)-based data augmentation has been successfully applied to imbalanced datasets to improve model performance. Using the KDD99 dataset, this approach achieved a 99.7% accuracy [6].

Federated Learning (FL) has addressed privacy challenges in IoT network intrusion detection. A privacy-preserving FL-based model achieved a 97.5% detection rate while maintaining data confidentiality [25].

To optimize computational efficiency for resource-constrained IoT devices, a lightweight deep learning-based intrusion detection system was proposed. This model, trained on the BoT-IoT dataset, achieved 98.3% accuracy [26].

Oversampling techniques like ADASYN, combined with Tomek Links to remove noisy data, have improved model performance in intrusion detection. When applied to the NSL-KDD dataset, the approach achieved a 99.6% detection accuracy [22].

Finally, integrating SMOTE with Tomek Links has proven effective in reducing false positives. This method reduced false positives by 25% compared to traditional oversampling techniques in anomaly-based intrusion detection [3].

2.6. Research Gap

While ADASYN and Tomek Links have been individually applied to address class imbalance and noisy data issues, limited studies have explored their combined potential. Existing models often fail to comprehensively address these challenges, leading to reduced detection accuracy for minority class anomalies in intrusion datasets.

Many studies, such as those using Recursive Feature Elimination (RFE) or Principal Component Analysis (PCA), have focused on dimensionality reduction to improve computational efficiency, [24]. However, the integration of advanced sampling techniques like ADASYN with feature selection methods has been underexplored.

ADASYN and Tomek Links for handling imbalanced and noisy datasets remains unexplored.

Existing studies, such as those using datasets like KDD CUP-99 and NSL-KDD, have demonstrated high detection accuracies [23]. However, these datasets may not fully represent modern, complex network environments. There is a need to test proposed methods on contemporary datasets like CICIDS2018 or UNSW-NB15 for broader applicability.

Techniques such as SMOTE combined with Tomek Links have proven effective in reducing false positives [3], but similar studies leveraging ADASYN and Tomek Links for this purpose are scarce. High false alarm rates remain a significant challenge in current intrusion detection systems.

Ensemble techniques like Random Forest and Gradient Boosting and deep learning models such as CNNs and BiLSTM have shown promising results [23, 3], their performance has not been studied in combination with ADASYN and Tomek Links.

Despite the availability of diverse evaluation metrics (e.g., detection rate, false positive rate, precision, recall), there is a lack of standardized comparisons among models incorporating ADASYN and Tomek Links, limiting consistent [3].

3. Research Process

The research process is an essential component of any research study, providing a comprehensive overview of the methods and procedures employed to achieve the research objectives. It also outlines the systematic approach and framework utilized to collect, analyze, and interpret the data, ensuring the validity and reliability of the research findings. This section further provides a brief overview of the specific methods and techniques employed within each component of the research methodology. Additionally, it may highlight the importance of data preprocessing techniques, such as cleaning, handling missing values, feature selection, and normalization, to ensure the quality and suitability of the data for analysis. The flow of this research methodology is visualized in Figure 2

3.1. Data Preprocessing

Once the datasets are collected, the next step is to preprocess the data. Data preprocessing involves handling missing values, outliers, and inconsistencies in the datasets. Techniques such as imputation, outlier detection, and data normalization are applied to ensure the data's quality and integrity.

Data Cleaning

Data cleaning is the process of identifying and correcting or removing errors, inconsistencies, and inaccuracies in the dataset. It ensures that the data is reliable and of high quality for subsequent analysis.

Feature Selection Process

The feature selection process typically involves the following steps:

1) Dataset Preparation: Ensure that the dataset is appropriately preprocessed, including handling missing values, outliers, and categorical variables. Data normalization or standardization may also be performed.

2) Evaluation Metric Selection: Select an appropriate evaluation metric for feature selection, such as information gain, chi-square, or correlation coefficient. The metric should align with the specific goals and characteristics of the research study.

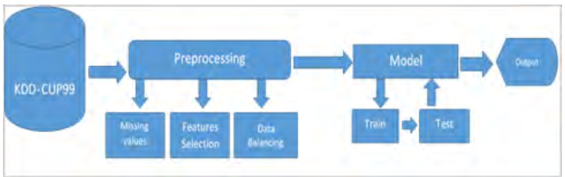


Figure. 2. Research methodology flow

3) Model Training and Evaluation: Train the machine learning models using the selected feature subset and evaluate their performance using appropriate metrics such as accuracy, precision, recall, F1-score, or area under the ROC curve.

Handling Missing Values

Missing values in the dataset can negatively impact the performance of machine learning algorithms. They need to be appropriately handled to ensure accurate and reliable analysis. Common techniques include using mean, median, or mode for numerical data, or using the most frequent category for categorical data.

Encoding Categorical Variables

Many datasets contain categorical variables that need to be encoded into numerical values for analysis by machine learning algorithms. In this work, the three non-numeric attributes were converted into numerical ones using one-hot encoding.

3.2. Data Balancing

Addressing class imbalance is a critical step in enhancing the effectiveness of network intrusion detection systems. This section provides a detailed overview of the data balancing techniques employed in the research study on enhancing cybersecurity through anomaly-based network intrusion detection using a combined approach of ADASYN and Tomek link.

Class Imbalance Problem

In network intrusion detection datasets, the occurrence of normal instances (negative class) significantly outweighs the instances of network intrusions (positive class). This class imbalance can lead to biased model performance, with the classifier favoring the majority class and performing poorly in detecting intrusions.

Combined Approach

The combined approach of ADASYN and Tomek link is utilized to achieve a more balanced representation of classes in the dataset. ADASYN oversamples the minority class, generating synthetic instances to increase its representation, while Tomek link further enhances class separation by removing instances that are close to the decision boundary.

Flowchart representation for Adasyn + Tomeklink

```
                 Start
                   ↓
         Check Class Imbalance
                   ↓
Is Dataset Imbalanced? -- No --> Use
            Original Dataset
                 ↓ Yes
              Apply ADASYN
                   ↓
```

```
Generate Synthetic Samples for Minority
                Class
                  ↓
           Apply Tomek Links
                  ↓
      Remove Overlapping Data Points
                  ↓
      Balanced and Cleaned Dataset
                  ↓
      Train Machine Learning Model
                  ↓
                 End
```

3.3. Model Evaluation Metrics

For each selected model, the evaluation will be conducted using various performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve. These metrics provide insights into the model's ability to correctly classify normal and anomalous network traffic, its robustness against false positives and false negatives, and its overall effectiveness in detecting network intrusions.

3.4. Comparison and Selection

The performance of the selected models (Random Forest, J48, MLP, and Bagging) will be compared based on the evaluation metrics. The models will be trained and evaluated on the balanced dataset obtained after applying the combined approach of ADASYN and Tomek link. The model with the highest overall performance, considering the specific requirements and constraints of the research, will be selected as the primary model.

4. Findings

4.1. Experimental Results

After analyzing the dataset, it was discovered to be imbalanced, with a significant disparity in the number of samples between the classes. To mitigate this issue, two data class balancing techniques were combined: Adasyn and Tomek link to form an hybrid technique. ADASYN (Adaptive Synthetic Sampling) is an oversampling technique used to address class imbalance in machine learning. It is specifically designed to handle imbalanced datasets by generating synthetic samples for the minority class based on their density distribution in the feature space. The ADASYN algorithm adapts to the characteristics of the dataset, adjusting the synthesis of samples according to the density distribution. It provides a data-driven approach that can be effective in scenarios where the imbalance between classes is more complex and varies across the feature space. While Tomek links on the other hand is an undersampling technique used to address the issue of class imbalance in machine learning. It focuses on identifying and removing instances that form pairs of samples from different classes and are located in close proximity to each other in the feature space. These pairs are known as Tomek links.

Figure 3 illustrates the data class distribution, revealing a significant imbalance. Specifically, the "Non Neptune" class contains a total of 84,759(67.28%) instances, while the "Neptune" class consists of 41,214(32.78%) instances.

In contrast in Figure 4, subsequent to the application of the class balance algorithm, the resultant distribution shows that Non Neptune is 84,838 (50.02%) while the Neptune: 84,759 (49.98%). This adjustment highlights a significant improvement in the balance between the classes, successfully mitigating the initial issue of imbalance.

4.2. Intrusion Detection experiments with Imbalance Data Class

Four distinct experiments were conducted using imbalanced data classes, each employing a different classifier. The outcomes of these experiments are outlined below.Table 1 offers an in-depth analysis of performance metrics for the Random Forest classifier.

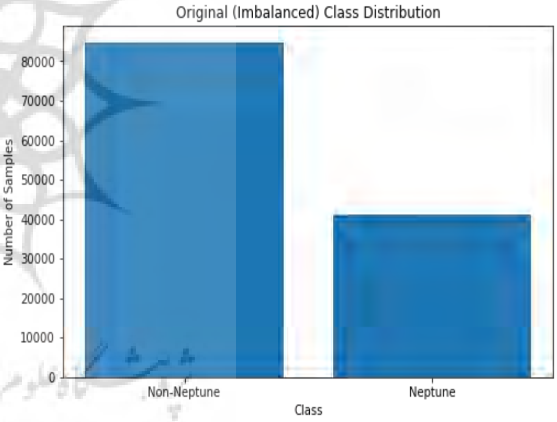


Figure. 3. Illustrates dataset class distribution before balancing

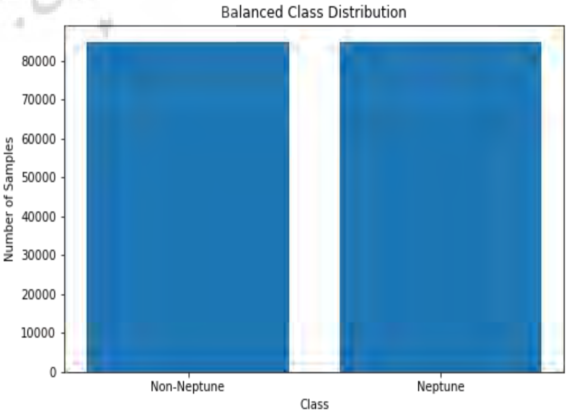


Figure. 4. Displays the class distribution after the application of Adasyn and Tomek link

In Figure 5, the confusion matrix shows True Positives (TP):15,830, False Positives (FP): 60, False Negatives (FN): 63 True Negatives (TN): 6,591.

The classification report, as depicted in Table 2, offers an in-depth analysis of the J48 classifier's performance metrics.

The matrix in Figure 6, clearly delineates the classification outcomes, with "Actual Non-Neptune" signifying instances truly belonging to the non-Neptune class, and "Actual Neptune" indicating instances from the Neptune class.

Table 3 shows the precision, recall, and F1-score values for Non-neptune and Neptune indicate exceptional accuracy in classifying instances, especially in the presence of class imbalance.

As obtained in Figure 7, the classifier correctly identified 15830 instances as intrusions (True Positives). It misclassified 60 legitimate data instances as intrusions (False Positives). The classifier correctly identified 6622 instances as legitimate data (True Negatives). It misclassified 32 intrusions as legitimate data (False Negatives).

Table 4 report shows precision, recall, and F1-score metrics for Non-neptune and Neptune reflect outstanding accuracy in classifying instances, even in the presence of class imbalance.

The classifier correctly identified True Positives (TP):15,789, False Positives (FP): 101, False Negatives (FN): 11 True Negatives (TN): 6,645, as shown in Figure 8.

The summarized results, detailed in Table 5, offer insights into the performance of each classification algorithm.

Clearly shown in Figure 9, the performance variation of the four classification algorithms used with imbalance data class.

4.3. Intrusion Detection experiments with Balanced Data Class

Four separate experiments were executed, each utilizing distinct classifiers and balanced data classes. The subsequent outcomes of these experiments are delineated as follows

The classification report in Table 6 showcases the Random Forest algorithm's robust performance in handling balanced data classes. Its exceptional precision, recall, and F1-score values underscore its potential as a reliable tool for accurate classification in scenarios characterized by balanced datasets.

In Figure 10, the classifier correctly identified True Positives (TP):15,830, False Positives (FP): 60, False Negatives (FN): 49 True Negatives (TN): 15,853.

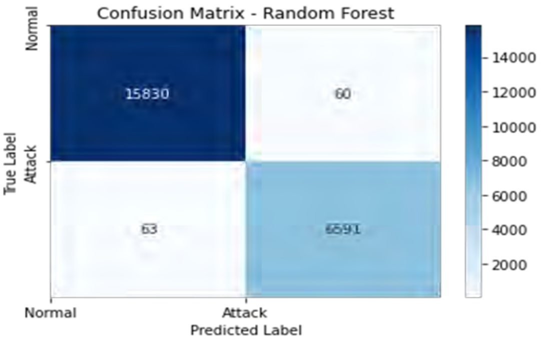


Figure. 5. Confusion Matrix for Random Forest

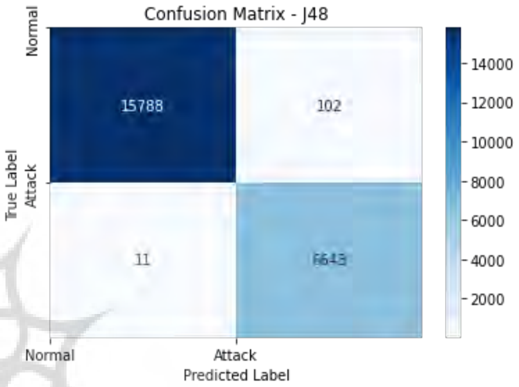


Figure. 6. Confusion Matrix for J48

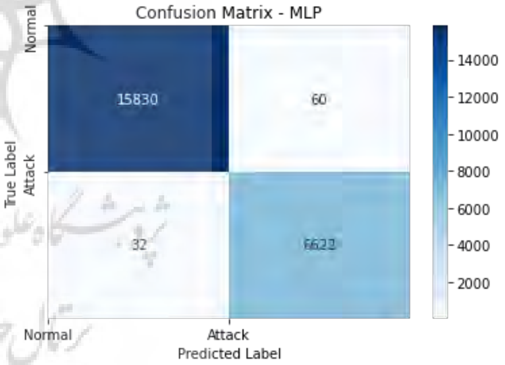


Figure. 7. Confusion Matrix for MLP

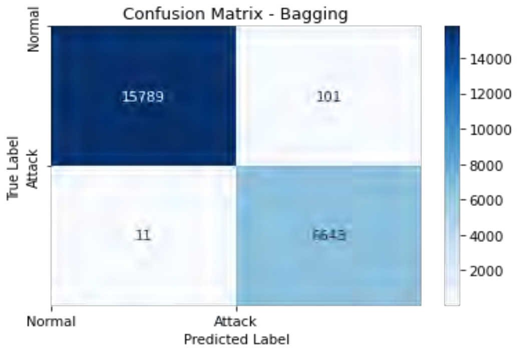


Figure. 8. Confusion Matrix for Bagging

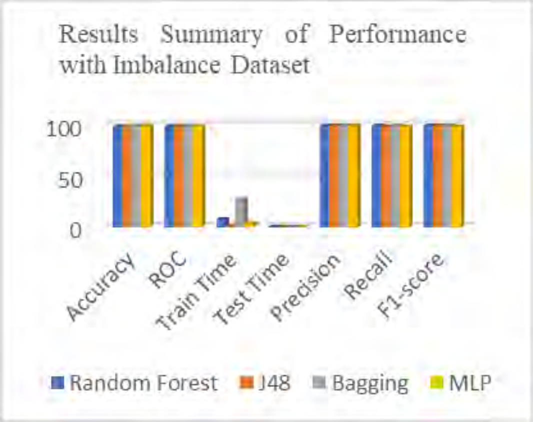


Figure. 9. Classifiers Results summary with Imbalance data class

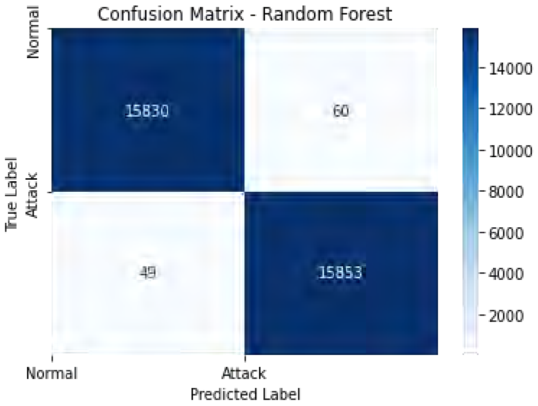


Figure. 10. Confusion Matrix for Random Forest

Table 1. Classification Report for Random Forest

	Precision	*Recall*	*Fi-score*	*Support*
Non Neptune	1.00	1.00	1.00	15890
Neptune	0.99	0.99	0.99	6654
Accuracy			0.99	22544
Macro avg	0.99	0.99	0.99	22544
Weighted avg	0.99	0.99	0.99	22544

Table 2. Classification Report for J48

	Precision	*Recall*	*Fi-score*	*Support*
Non Neptune	1.00	1.00	1.00	15890
Neptune	0.99	0.99	0.99	6654
Accuracy			0.99	22544
Macro avg	0.99	0.99	0.99	22544
Weighted avg	0.99	0.99	0.99	22544

Table 3. Classification Report for MLP

	Precision	*Recall*	*Fi-score*	*Support*
Non Neptune	1.00	1.00	1.00	15890
Neptune	0.99	1.00	0.99	6654
Accuracy			1.00	22544
Macro avg	0.99	1.00	1.00	22544
Weighted avg	1.00	1.00	1.00	22544

Table 4. Classification Report for Bagging

	Precision	*Recall*	*Fi-score*	*Support*
Non Neptune	1.00	0.99	1.00	15890
Neptune	0.99	1.00	0.99	6654
Accuracy			1.00	22544
Macro avg	0.99	1.00	0.99	22544
Weighted avg	1.00	1.00	1.00	22544

Table 5. Results Summary Obtained from the Experiment with Imbalance Data Class.

Classifiers	*Accuracy*	*ROC*	*Train Time*	*Test Time*	*Precision*	*Recall*	*F1-score*
Random Forest	99	99.33	7.58	0.23	100	100	100
J48	99	99.60	0.53	0.016	100	99	100
Bagging	100	99.56	27.47	0.032	100	100	100
MLP	100	99.60	3.94	0.13	100	99	99

Table 7 highlights the J48 classifier's robust performance on a balanced dataset. This analysis showcases its capability to maintain a high level of accuracy and balanced trade-offs between precision and recall, making it a suitable choice for Intrusion Detection tasks involving balanced data classes.

The classifier in Figure 11, correctly identified True Positives (TP):15,789, False Positives (FP): 112, False Negatives (FN): 100 True Negatives (TN): 15,802.

The precision-recall report in Table 8 provides a detailed evaluation of the MLP classifier's performance on the balanced dataset. The Non-

Neptune Class, the classifier exhibits a precision, recall, and F1-score of 0.97 and 1.00 respectively.

The classifier in Figure 12, correctly identified True Positives (TP):15,832, False Positives (FP): 58, False Negatives (FN): 407, True Negatives (TN): 15,495.

The classification report in Table 9 provides a comprehensive evaluation of the Bagging classifier's performance on the balanced dataset. For Non-Neptune Class, the classifier attains a precision, recall, and F1-score of 0.99. This demonstrates its proficiency in accurately identifying instances from this class while maintaining balanced trade-offs between false positives and false negatives. Similarly, for Neptune Class, the classifier showcases comparable precision, recall, and F1-score values of 0.99, signifying its capacity to effectively manage instances from this class.

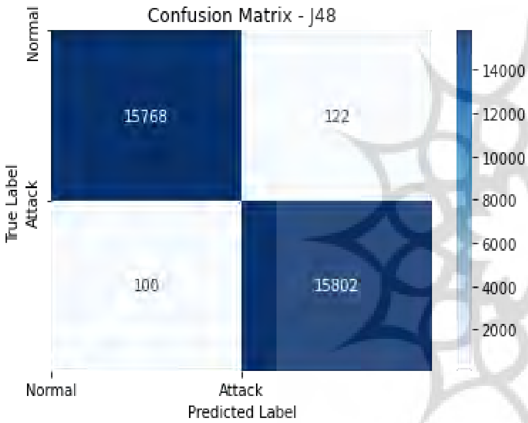


Figure. 11. Confusion Matrix for J48

Table 6. Classification Report for Random Forest

	Precision	Recall	Fi-score	Support
Non Neptune	1.00	1.00	1.00	15890
Neptune	1.00	1.00	1.00	15902
Accuracy			1.00	31792
Macro avg	1.00	1.00	1.00	31792
Weighted avg	1.00	1.00	1.00	31792

Table 7. Classification Report for j48

	Precision	Recall	Fi-score	Support
Non Neptune	0.99	0.99	0.99	15890
Neptune	0.99	0.99	0.99	15902
Accuracy			0.99	31792
Macro avg	0.99	0.99	0.99	31792
Weighted avg	0.99	0.99	0.99	31792

The classifier in Figure 13, correctly identified True Positives (TP):15,752, False Positives (FP): 138, False Negatives (FN): 83 True Negatives (TN): 15,819.

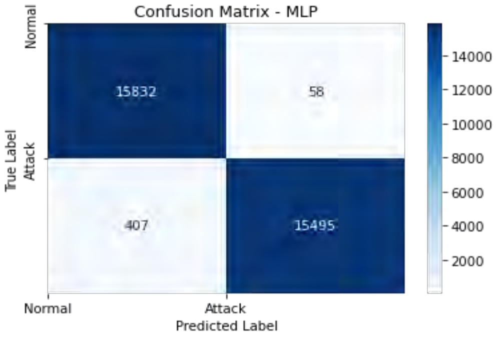


Figure. 12. Confusion Matrix for MLP

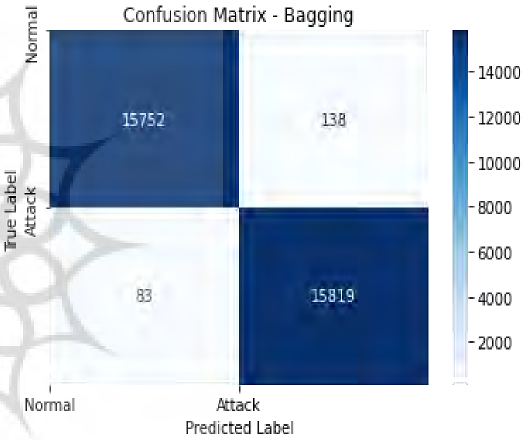


Figure. 13. Confusion Matrix for Bagging

Table 8. Classification Report for MLP

	Precision	Recall	Fi-score	Support
Non Neptune	0.97	1.00	0.99	15890
Neptune	1.00	0.97	1.00	15902
Accuracy			0.99	31792
Macro avg	0.99	0.99	0.99	31792
Weighted avg	0.99	0.99	0.99	31792

Table 9. Classification Report for Bagging

	Precision	Recall	Fi-score	Support
Non Neptune	0.99	0.99	0.99	15890
Neptune	0.99	0.99	0.99	15902
Accuracy			0.99	31792
Macro avg	0.99	0.99	0.99	31792
Weighted avg	0.99	0.99	0.99	31792

4.3.5. Summary of Results obtained from the four classification algorithms on Intrusion detection with balanced class.

This section provides a comprehensive summary of the results achieved from the Intrusion Detection experiments using Adasyn combined with Tomek Link techniques to balance the data classes. The outcomes of these experiments, as presented in Table 10, encompass key performance metrics for each classifier, including accuracy, Receiver Operating Characteristic (ROC), training time, testing time, precision, recall, and F1-score.

Table 10 offer insights into the performance of each classifier under balanced data conditions. Random Forest classifier attains a perfect accuracy of 99.67%, and its high ROC score of 99.66% signifies its capability to discriminate between classes effectively. With optimal precision, recall, and F1-score values. Also, J48 achieved an accuracy of 99.30% and a commendable ROC score of 99.32%, J48 exhibits consistent precision, recall, and F1-score metrics. Its relatively low training and testing times make it an efficient option. Multilayer Perceptron: The classifier demonstrates an accuracy of 98.53%, with a respectable ROC score of 98.54%. Despite a longer training time, its precision, recall, and F1-score values highlight its ability to effectively classify instances. Lastly, Bagging similar to J48, Bagging achieves an accuracy of 99.30% and an impressive ROC score of 99.30%. Its balanced precision, recall, and F1-score metrics reinforce its suitability for balanced data scenarios.

Considering the collective performance metrics "Random Forest" classifier emerges as the recommended choice for Intrusion Detection when using balanced data classes.

Figure 14, further visualize the result summary.

In terms of the accuracy, the classifiers consistently maintain high accuracy levels, indicating their ability to handle diverse data distributions. Also, in training and testing time - the classifiers vary in terms of training and testing times. While some classifiers exhibit faster training times, others excel in testing efficiency.

Precision and F1-score in general, the precision and F1-score values remain consistent across both scenarios, demonstrating the classifiers' ability to achieve a balance between correct classifications and minimizing false positives and negatives(Table 11).

Notably, the "Random Forest" classifier consistently maintains a high accuracy, precision, and F1-score across both scenarios. Its relatively short training and testing times further underscore its efficiency.

Considering the overall consistency in performance and efficiency across both balanced and imbalanced data classes, the "Random Forest" classifier emerges as the most versatile and effective option for Intrusion Detection.

5. Conclusions

The inception of this research was marked by the recognition of class imbalance as a pervasive challenge in Intrusion Detection. The subtle presence of malicious activities amidst an overwhelming sea of normal behaviors underscores the need for refined techniques to identify and thwart potential threats. Adasyn and Tomek Link emerged as dynamic tools in rectifying this imbalance, providing classifiers with a more balanced platform to discern patterns and anomalies.

The expedition into balanced data scenarios reaffirmed the capabilities of the selected classifiers. Across the spectrum, each classifier demonstrated a consistent ability to achieve high accuracy, precision, recall, and F1-score values. This showcases their proficiency in deciphering intricate patterns within well-distributed classes, thereby ensuring robust cybersecurity in scenarios where class imbalance is not the dominant concern.

In the realm of imbalanced data, the application of Adasyn and Tomek Link yielded remarkable results. The classifiers showcased exceptional precision, recall, and accuracy in detecting instances from both classes. Notably, the "Random Forest" classifier emerged as a standout performer, boasting perfect accuracy and a harmonious balance between precision and recall. This underscores the potential of advanced classifiers coupled with strategic balancing techniques in addressing class imbalance.

5.1. Limitation of the Study

The research has several limitations. Firstly, its results are heavily dependent on the quality of datasets, such as KDDCUP 99, which may not reflect modern cyber threats accurately. Secondly,

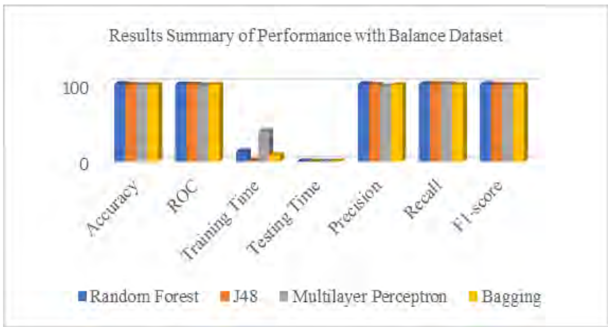
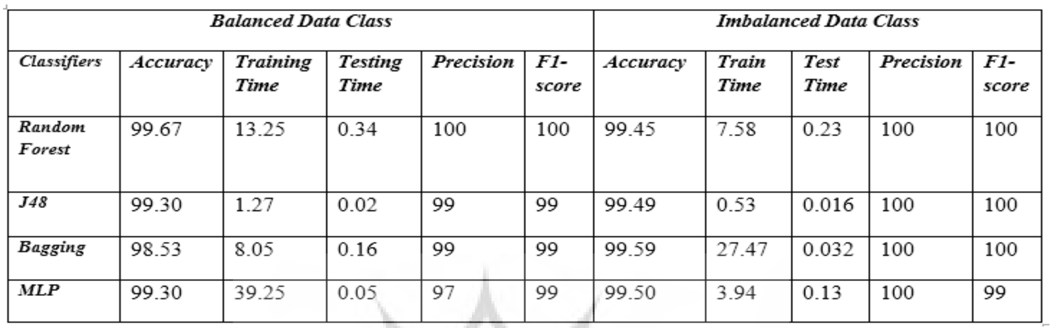


Figure. 14. Results Summary of Performance with Balance Dataset

Table 10. Summary Intrusion Detection experiments with Balanced Data Class using Adasyn combined with Tomek link.

Classifiers	*Accuracy*	*ROC*	*Train Time*	*Test Time*	*Precision*	*Recall*	*F1-score*
Random Forest	100	99.66	13.25	0.34	100	100	100
J48	99.00	99.32	1.27	0.02	99	100	99
Bagging	99.00	98.54	39.25	0.05	97	100	99
MLP	99.00	99.30	8.05	0.16	99	99	99

Table 11. Overall summary of results obtained from both Imbalance and balance data class



Balanced Data Class						*Imbalanced Data Class*				
Classifiers	*Accuracy*	*Training Time*	*Testing Time*	*Precision*	*F1-score*	*Accuracy*	*Train Time*	*Test Time*	*Precision*	*F1-score*
Random Forest	99.67	13.25	0.34	100	100	99.45	7.58	0.23	100	100
J48	99.30	1.27	0.02	99	99	99.49	0.53	0.016	100	100
Bagging	98.53	8.05	0.16	99	99	99.59	27.47	0.032	100	100
MLP	99.30	39.25	0.05	97	99	99.50	3.94	0.13	100	99

the preprocessing techniques, including ADASYN and Tomek Link, add computational complexity, making real-time deployment challenging. The framework also struggles with scalability in high-speed or large-scale networks. Additionally, it has a limited focus on identifying advanced and evolving threats, such as zero-day attacks. Overfitting risks arise from the synthetic data generated by ADASYN. Lastly, the framework has been tested primarily in offline environments, with no validation conducted in live, real-time settings.

5.2. Suggestions for Future Work

1) Test the framework in real-time settings under dynamic traffic conditions.

2) Integrate deep learning (e.g., CNNs, transformers) for complex attack patterns.

3) Use modern datasets like CICDDoS2019 and UNSW-NB15 for validation.

4) Optimize preprocessing techniques for resource-constrained IoT environments.

5) Ensure transparency by leveraging explainable AI (XAI) techniques.

Declarations

Funding

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Authors' contributions

Nasiru Ige Salihu: Study design, acquisition of data, drafting the manuscript;

Muhammed Nazeer Musa: Study design, interpretation of the results, drafting the manuscript, revision of the manuscript.

Awujoola J. Olalekan: interpretation of the results, statistical analysis, revision of the manuscript.

Conflict of interest

The authors declare that no conflicts of interest exist. All information from other sources (published or unpublished) were referenced accordingly.

References

[1] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021, https://doi.org/10.1016/j.procs.2021.05.025

[2] U. S. Musa, S. Chakraborty, M. M. Abdullahi, and T. Maini, "A review on intrusion detection system using machine learning techniques," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2021, pp. 541–549 https://doi.org/10.1109/ICCCIS51004.2021.9397121

[3] Z. Ahmed and S. Das, "A Novel Hybrid Resampling Approach to Address Class-Imbalanced Issues," *SN Computer Science*, vol. 5, no. 7, p. 865, 2024, https://doi.org/10.1007/s42979-024-03227-z

[4] C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, "Comparative research on network intrusion detection methods based on machine learning," *Computers & Security*, vol. 121, p. 102861, 2022

https://doi.org/10.1016/j.cose.2022.102861

[5] J. P. Anderson, "Computer security technology planning study," *ESD-TR-73-51*, vol. 1, p. 4, 1972, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=07356c5477c83773bd062b525f45c433e5b044e8

[6] F. M. Alotaibi, "Network Intrusion Detection Model Using Fused Machine Learning Technique," *Computers, Materials & Continua*, vol. 75, no. 2, pp. 2479-2490, 2023. https://doi.org/10.32604/cmc.2023.033792

[7] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987, https://doi.org/10.1109/TSE.1987.232894

[8] I. El Naqa and M. J. Murphy, *What is Machine Learning?*, pp. 3–11, Springer International Publishing, 2015.

https://doi.org/10.1007/978-3-319-18305-3_1

[9] R. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. Kendall, D. McClung, D. Weber, and D. Wyschogrod, "The 1999 DARPA Off-line Intrusion Detection Evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579–595, 2000, https://doi.org/10.1016/S1389-1286(00)00139-0

[10] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018,pp.883–892. https://proceedings.mlr.press/v80/chen18j.html

[11] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, and H. Yuanyue, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017, https://doi.org/10.1016/j.eswa.2016.12.035

[12] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no.1 ,pp. 5–32, 2001, https://doi.org/10.1023/A:1010933404324

[13] H. Alqahtani, I. H. Sarker, A. Kalim, S. M. Minhaz Hossain, S. Ikhlaq, and S. Hossain, "Cyber intrusion detection using machine learning classification techniques," in *Computing Science, Communication and Security: First International Conference, COMS2 2020, Gujarat, India, March 26–27, 2020, Revised Selected Papers 1*, Springer Singapore, 2020, pp. 121–131. https://doi.org/10.1007/978-981-15-6648-6_10

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

[15] M. Belouch, S. El Hadaj, and M. Idhammad, "Performance evaluation of intrusion detection based on machine learning using Apache Spark," *Procedia Computer Science*, vol. 127, pp. 1–6, 2018, https://doi.org/10.1016/j.procs.2018.01.091

[16] S. Zafar, M. Kamran, and X. Hu, "Intrusion-Miner: A Hybrid Classifier for Intrusion Detection using Data Mining," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, 2019, https://thesai.org/Downloads/Volume10No4/Paper_40-Intrusion_Miner_A_Hybrid_Classifier_for_Intrusion_Detection.pdf

[17] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Hong Kong, China, Jun. 2008, pp. 1322–1328.

https://doi.org/10.1109/IJCNN.2008.4633969

[18] M. Conti, T. Dargahi, and A. Dehghantanha, *Cyber Threat Intelligence: Challenges and Opportunities*. Springer International Publishing, 2018, pp. 1–6. https://link.springer.com/chapter/10.1007/978-3-319-73951-9_1

[19] M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Paris, France, IEEE, 2018, pp. 1–5.

https://doi.org/10.1109/NTMS.2018.8328749

[20] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Feature selection using genetic algorithm to improve classification in network intrusion detection system," in *International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, Surabaya, Indonesia, 2017, pp. 46–49, https://doi.org/10.1109/KCIC.2017.8228458

[21] F. Salo, M. Injadat, A. B. Nassif, A. Shami and A. Essex, "Data Mining Techniques in Intrusion Detection Systems: A Systematic Literature Review," in *IEEE Access*, vol. 6, pp. 56046-56058, 2018, https://doi.org/10.1109/ACCESS.2018.2872784

[22] H. A. Ahmed, A. Hameed, and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Computer Science*, vol. 8, p. e820, 2022, https://doi.org/10.7717/peerj-cs.820

[23] A. D. J. Atul, R. Kamalraj, G. Ramesh, K. S. Sankaran, S. Sharma, and S. Khasim, "A machine learning based IoT for providing an intrusion detection system for security," *Microprocessors and Microsystems*, vol. 82, p. 103741, 2021,

https://doi.org/10.1016/j.micpro.2020.103741

[24] S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, "Enhancing IoT network security through deep learning-powered Intrusion Detection System," *Internet of Things*, vol. 24, p. 100936, 2023.

https://doi.org/10.1016/j.iot.2023.100936

[25] M. Esmaeili, M. Rahimi, H. Pishdast, D. Farahmandazad, M. Khajavi, and H. J. Saray, "Machine Learning-Assisted Intrusion Detection for Enhancing Internet of Things Security," *arXiv preprint arXiv:2410.01016*, 2024. https://doi.org/10.48550/arXiv.2410.01016

[26] X. Zhou, S. Wang, K. Wen, B. Hu, X. Tan, and Q. Xie, "Security-Enhanced Lightweight and Anonymity-Preserving User Authentication Scheme for IoT-Based Healthcare," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 9599-9609, 2023. https://doi.org/10.1109/JIOT.2023.3323614

Nasiru Ige Salihu is an experienced system analyst and ICT professional specializing in cybersecurity, computer science, and educational instruction, currently working as a Principal System Analyst/Programmer at the Directorate of Information and Communication Technology, Nigerian Defence Academy, Kaduna. He holds an MSc (Professional) in Cyber Security from the Nigerian Defence Academy post graduate school, kaduna (2023), a BSc in Computer Science from Kogi State University, Kogi state, Nigeria (2012).

MN Musa had B.Tech in Computer Science from Federal University of Technology (FUT) Minna in the year 2015. He furthered his studies and got his MSc in Computer Science from Nigerian Defence Academy (NDA) Kaduna in 2020. His areas of interest include data mining, machine learning, computer vision, deep learning, cyber security, and digital forensics. He currently lectures in the Department of Cyber Security, Nigerian Defence Academy, Kaduna, Nigeria.

Awujola Joel Olalekun is a highly educated professional currently serving as a Chief System Analyst/Programmer at the Nigerian Defence Academy Kaduna, Nigeria. He holds a PhD in Computer Science and has studied MSc in both Nuclear and Radiation Physics at the Nigerian Defence Academy and Computer Science at Ahmadu Bello University. Additionally, he earned a BTech in Mathematics and Computer Science from the Federal University of Technology Minna. Nigeria.