SAJSET
ISSN: 2992-3581

Savannah Journal of Science and Engineering Technology
Volume: 02; Issue: 04; Pages: 181-189; 2024.
www.sajsetjournal.com.ng

DOI: xx.xxxx.x/sajset-02-2024-0026

SAJSET
ISSN: 2992-3581

Enhancing network intrusion detection with SMOTE: A performance analysis

[1,2]Baba, U. A., [2]Garba, E. J. and [2]Ahmadu, A. S.
[1]*Department of Computer Science, North-Eastern University, Gombe*
[2]*Department of Computer Science, Modibbo Adama University, Adamawa*

Paper History
Received: 01st July, 2024
Accepted: 26th July, 2024
Published: July, 2023

Corresponding author
Baba, U. A.
usmanbabatela@gmail.com

Abstract:
Network Intrusion Detection Systems (NIDS) play a vital role in safeguarding computer networks against malicious activities. However, a significant challenge in developing effective NIDS is the inherent class imbalance in datasets, where normal traffic vastly outnumbers malicious instances. This study investigates the impact of the Synthetic Minority Over-sampling Technique (SMOTE) on machine learning models for network intrusion detection. The performance of seven popular algorithms (KNN, Naive Bayes, SVM, XGBoost, Logistic Regression, Random Forest, and Multi-Layer Perceptron) were evaluated on two benchmark datasets: UNSW-NB15 and NSL-KDD. These datasets represent different levels of class imbalance in network traffic data. The model's performance was evaluated using accuracy, precision, recall, and F1-score metrics, both with and without the application of SMOTE. The Naive Bayes (NB) classifier exhibited the most significant change, with precision increasing from 0.83 to approximately 1.00 after applying SMOTE, but at the cost of decreased recall and F1-score. The overall findings reveal that the effectiveness of SMOTE varies significantly depending on the dataset's inherent class distribution and the specific machine learning algorithm used. This study provides insights into when and how SMOTE should be applied in NIDS to enhance performance.

Keywords: *Accuracy, Intrusion detection system, K-Nearest Neighbors, NIDS, Precision, SMOTE, Support Vector Machine*

1. Introduction

The advent of the internet has brought about a profound transformation in all aspects of society, including but not limited to business transactions, information access, communication, and social interaction. The advancement of computer networks has made a significant contribution to the economic prosperity of nations and has had a profound effect on the increasing number of users in these networks. Nevertheless, the enhancement of internet accessibility is intrinsically hindered by the continuous advancement of information systems security [1].

In the field of information security, Intrusion Detection Systems (IDSs) have been conventionally categorized into two main types: Network IDS (NIDS) and Host IDS (HIDS) [2]. The first approach primarily involves the examination of network traffic flows or traces, typically obtained from firewalls, routers, switches, or other network devices. In contrast, the second approach considers data sources specific to individual hosts, such as Syslog files or CPU load. IDSs can be classified into two main categories based on their attack detection procedure. The first category is Signature-based IDS (SIDS), which identifies malicious behavior by comparing it to pre-established attack signatures. The second category is Anomaly-based IDS (AIDS), which detects anomalies in the system by identifying events that deviate from the normal behavior [3].

The evaluation of this type of system is generally conducted by utilizing pre-established datasets derived from simulated or emulated scenarios that closely resemble the ultimate application network or system [4]. Machine Learning (ML)-based methods that implement Network Intrusion Detection Systems (NIDSs) are frequently encountered in the literature [5]. Therefore, it is possible to distinguish between supervised, semi-supervised, and unsupervised techniques [6]. In the context of supervised models, it is necessary to have datasets that have labels assigned to each sample to classify known assaults. However, while unsupervised learning does not always require labelled data, it is still suggested to have labelled data to validate the performance of the algorithms. In the domain of semi-supervised learning, it is seen that both preceding methodologies are present. The utilization of the unsupervised component enables the identification of aberrant activity, which can subsequently be categorized within the supervised component [7].

Anomaly-based detection is currently a prominent topic of research and development in the field of IDS due to its significance [8]. Prior to implementing an anomaly-based intrusion detection system on a large scale, it is crucial to address certain unresolved concerns [3]. However, the current body of work is lacking in terms of comparative analysis on the performance of IDS when employing

supervised machine learning approaches [9]. Anomaly-based NIDS serve as a significant solution for safeguarding target systems and networks from hostile activities. In recent years, the literature has presented a range of anomaly-based network intrusion detection algorithms [3]. However, the implementation of security solutions with anomaly detection capabilities is still in its early stages, and there are unresolved issues that need to be addressed.

Various anomaly detection strategies have been suggested in the literature, such as Linear Regression, Genetic Algorithm, Support Vector Machines (SVM), Gaussian mixture model, Naive Bayes classifier, Decision Tree, and k-nearest neighbour algorithm, [10]. One of the most commonly employed learning algorithms is SVM, which has demonstrated its efficacy across several problem domains. One significant concern regarding anomaly-based detection is that while the proposed algorithms are capable of detecting new assaults, they tend to exhibit a high percentage of false alarms. The underlying factor is the intricacy involved in establishing profiles of typical practical behaviour through the process of learning from the training datasets [11].

One of the primary obstacles encountered when assessing the effectiveness of network IDS is the lack of a comprehensive dataset that encompasses network-based information. The majority of the anomaly-based approaches discussed in the literature were assessed using the KDD CUP 99 dataset [4]. In this research, the UNSW-NB15 and NSLKDD datasets were utilized as widely recognized benchmark datasets for network intrusion. The advancements and impact of machine learning thus far have been captivating. In today's landscape, optimizing processing and training procedures is imperative for constructing models that can effectively safeguard systems against dubious and spyware activities [12]. However, many contemporary Machine Learning-based IDS face limitations due to their reliance on small, outdated, and balanced datasets for model development [13].

Machine learning has become increasingly prevalent in several real-world applications that are now in use. The focus on smaller, often outdated datasets, coupled with imbalances in data distribution, raises questions regarding the practical applicability of these models in real-world scenarios, especially when dealing with big data. The achievable accuracy of such models often hinges on the intricacies of dataset preprocessing and the selection of suitable algorithms, adding an additional layer of complexity to their effectiveness [14].

It appears that machine learning is poised to exert significant influence over global affairs in the foreseeable future. Therefore, to address these challenges, there is a pressing need to develop and validate ML-based IDS that can handle large, imbalanced datasets encompassing all potential attack scenarios. The research emphasizes constructing a robust and well-structured framework for efficiently detecting network intrusions in substantial datasets. Data preprocessing techniques, specifically min-max normalization, followed by the application of the Synthetic Minority Over-sampling Technique (SMOTE) were employed to address the unique challenges posed by big datasets.

The use of SMOTE is particularly significant as it generates synthetic instances of the minority class by interpolating between existing minority samples, thereby balancing the class distribution without simply duplicating instances, which can lead to overfitting. This study evaluates the impact of SMOTE on the performance of ML-IDS by comparing results obtained with and without the application of SMOTE.

1.1. Objectives of the research

The key objectives of the research include:

a. Demonstrating the importance of handling imbalanced data in NIDS.
b. Implementing min-max normalization as a preprocessing step.
c. Applying SMOTE to mitigate class imbalance and enhance detection rates.
d. Evaluating the effectiveness of the proposed approach by comparing model performance with and without SMOTE.

This approach is rigorously evaluated using various machine learning classifiers, including, Random Forest (RF), Extreme Gradient Boosting (XGB), KNN, Naïve Bayes (NB), and SVM. The models were assessed using a comprehensive set of performance indicators, including precision, recall, F1-score, confusion matrix, and accuracy. The results indicate significant improvements in detection rates and overall model accuracy when SMOTE is applied, highlighting its potential as a valuable preprocessing step in NIDS.

1.2 Related work

The research by Le, *et al.* [15] utilized the KDD'99 dataset, employing the Ant Colony Algorithm to select an appropriate representative subset from the original dataset, which consisted of 550 samples. Subsequently, the researchers employed a novel technique called the Gradual Feature Removal (GFR) method to effectively decrease the dimensionality of the feature space, resulting in a reduced set of 19 features. The diminished characteristics are then integrated with SVMs for the purpose of classification. The first reported accuracy stands at 98.67% before feature selection, whereas the accuracy remains relatively unchanged following feature selection, yielding a result of 98.62%.

In the research by Rathore, *et al.* [16], the authors have presented an Intrusion Detection System (IDS) architecture that effectively tackles the emerging issues associated with the real-time management of large volumes of network data. To assess the system, the performance of attack classification was evaluated using widely recognized machine learning methods using the traditional KDDCup'99 dataset. In order to enhance the classification performance of attack detection in communication networks, a novel strategy utilizing a greedy feature selection (FS) technique was introduced by the authors in Le, *et al.* [15]. The researchers evaluated their idea by considering the NSL-KDD and ICSX12 datasets, along with several deep learning and classical machine learning methodologies. The researchers reached a conclusion regarding the viability of the methodology in relation to its classification performance

and memory usage. However, this paper did not specify or propose any feature extraction or data preprocessing techniques.

The authors of Farnaaz and Jabbar [17] addressed the issue of intrusion classification utilizing the NSL-KDD dataset in their study. Initially, the dataset undergoes preprocessing procedures to address any missing data concerns and to classify numerical attributes. Next, the dataset is divided into four distinct clusters and subsequently separated into training and test datasets. Next, the data is inputted into a random forest classifier, which is used for classification. The classification results are then evaluated by calculating the accuracy and false positive rate (FPR). The application of a feature selection strategy is also implemented utilizing the Symmetrical Uncertainty measure. The researchers saw a marginal improvement in performance after using feature selection. The findings of this study were compared to those of C4.5, and it was observed that random forests exhibited superior performance compared to C4.5. The average accuracy reported after feature selection is 99.67%, while the false alarm rate is 0.005.

On the other hand, Belavagi and Muniyal [9] focused on the NSL-KDD data set and employed a hold-out testing methodology without utilizing any feature selection technique. They evaluated four classification algorithms, namely random forest, SVM, logistic regression, and Gaussian mixture model. The results indicated that the random forest algorithm achieved the highest accuracy of 99%. Logistic regression emerged as the second-best performing classifier with an accuracy of 84%, as reported by the authors. The research in Shah and Trivedi [18] investigated the impact of lowering the dimensional space of the KDD'99 dataset on the overall classification performance. The researchers conducted an analysis on the feature selection process known as information gain based on entropy (IG). They then proceeded to merge the resulting feature vector, which consisted of 22 features, with the Back Propagation Neural Network (BBNN). The findings indicated that there were no significant alterations in accuracy after the implementation of feature reduction, as it consistently maintained a level of 91%.

The authors of Hussain and Lalmuanawma [19] conducted a comprehensive analysis of traditional machine learning methods for the purpose of detecting anomalies in network systems. To achieve the objectives of the research, the researchers utilized two widely recognized network datasets: KDDCup'99 and NSL-KDD. Additionally, an examination was conducted to assess the impact of introducing noise to the data in the NSL-KDD dataset. On one side, it was observed that the performance of nearly all algorithms improved when applied to datasets that were more representative of real-world scenarios, such as the NSL-KDD dataset and its variations that included noise. Conversely, each algorithm possessed distinct and pertinent characteristics that contributed to improved outcomes. Additionally, a Performance-based Method of Ranking (PMR) FS algorithm was presented for this purpose.

A research by Divekar, *et al.* [20] identified certain deficiencies in the KDDCup'99 dataset when used for evaluating contemporary NIDS. In response, they proposed a study to assess the efficacy of various traditional Machine Learning (ML) models on two alternative benchmarks. This study aimed to determine the appropriateness of the KDDCup'99 dataset for evaluating modern NIDS. The Mean Decrease Impurity approach was employed to eliminate irrelevant features, with the string values of certain characteristics being converted into integer numbers. In addition to employing the SMOTE technique for oversampling, a subsequent random under-sampling procedure was utilized to create a dataset of appropriate dimensions, ensuring a balanced representation of samples across all classes. The hyper-parameters of the ML models were configured through the utilization of randomized grid search and 5-fold cross-validation.

The research article published by Siddique, *et al.* [21] brought attention to the shortcomings identified in prior studies that focused on classifying network attacks. These studies often utilized an obsolete dataset, such as KDDCup'99, for evaluating the performance of their methods. Other studies in the same domain typically attain notable levels of classification accuracy, primarily driven by the identified limitations present in the KDDCup'99 dataset.

The accuracy of ten algorithms for classification on the NSL-KDD dataset was assessed by Prachi, *et al.* [22], utilizing a feature selection strategy that differed from that of Farnaaz and Jabbar [17]. The feature selection methodology relies on the utilization of attribute assessors and filtering techniques. The researchers employed a variety of algorithms in their study, including Naive Bayes, Bayes-Net, Logistics, Random tree, Random Forest, J48, Bagging, OneR, PART, and ZERO. The classification algorithm that exhibits the highest performance is a random forest, which achieves an accuracy of 99.9% and demonstrates a minimal false alarm rate of 0.001. The classifier that achieved the second highest performance is Bagging, which demonstrated an accuracy rate of 99.8%. Similarly, the PART method also exhibited a comparable level of performance.

In their study, Le, *et al.* [15] put up a novel approach to classification that combines the artificial fish swarm (AFS) and artificial bee colony (ABC) algorithms. The application of the Fuzzy C-Means Clustering (FCM) and Correlation-based Feature Selection (CFS) methods involves the utilization of these approaches to partition the training dataset and eliminate features that are deemed unnecessary, respectively. Furthermore, the generation of If-Then rules is facilitated by employing the Classification and Regression Trees (CART) technique, which considers the specific attributes that have been chosen. This approach aims to effectively differentiate between normal and anomalous records.

Similarly, the hybrid approach that is being presented is trained using the rules that have been generated. The experimental findings obtained from conducting simulations on the NSL-KDD and UNSW-NB15 datasets provide evidence that the suggested approach exhibits superior performance in relation to several performance metrics. Specifically, the proposed method attains a detection rate of 99% and a false positive rate of 0.01%. Table 1 summarizes some of the existing research in terms of dataset, data preprocessing, feature selection and results.

Table 1. Summary of the related work

Authors	Dataset	Data Preprocessing	Resampling Method	Results
Farnaaz and Jabbar [17]	NSLKDD	No	No	FPR = 0.005 %; Accuracy = 99.7 %
Prachi, *et al.* [22]	NSLKDD	No	No	FPR = 0.001%; Accuracy = 99.9 %
Le, *et al.* [15]	UNSW-NB15 and NSLKDD	No	No	UNSW-NB15 Accuracy = 98.9% DR = 98.6%, FPR =0.13; NSLKDD Accuracy = 99.0% DR = 99%, FPR =0.01.
Li, *et al.* [23]	KDDCup99	Yes	No	Accuracy=98.7%; MCC = 86.9
Belavagi and Muniyal [9]	NSLKDD	Yes	No	Accuracy = 99.0%; F1 score =99.0; Precision = 99.0; Recall = 99.0%
Shah and Trivedi [18]	KDDCup 99	Yes	No	Accuracy = 91, Precision = 99.7, Recall = 90.1; Fscore = 94.60
Rathore, *et al.* [16]	KDD	No	No	NB Accuracy = 94.1%; SVM Accuracy = 97.7% RF Accuracy = 98.9%
Hussain and Lalmuanawma [19]	NSLKDD and KDCup99	No	No	JRIP accuracy = 74.10%, FAR= 31 and J48 accuracy = 73%, FAR= 38%.
Li, *et al.* [24]	NSL-KDD, BGP	Yes	No	LSTM Accuracy = 82.78; Fscore = 83.34; GRU Accuracy = 82.87, Fscore = 83.05

Despite the significant advancements in NIDS and the application of machine learning techniques to improve their efficacy, many studies still rely on small, outdated, and balanced datasets as summarized earlier. This reliance limits the generalizability and real-world applicability of these models, particularly in handling large-scale, imbalanced datasets commonly encountered in modern network environments.

While various resampling techniques, including SMOTE, have been explored in different domains to address class imbalance, there is a notable gap in the comprehensive evaluation of SMOTE's impact specifically within the context of NIDS. Existing research often overlooks the critical step of integrating SMOTE with appropriate preprocessing techniques, such as min-max normalization, to optimize the performance of machine learning models in detecting minority attack classes.

Furthermore, there is a lack of systematic comparison of model performance with and without the application of SMOTE in NIDS. This gap hinders the understanding of SMOTE's true potential and effectiveness in enhancing detection rates and overall model robustness in real-world NIDS deployments.

This study aims to fill this gap by thoroughly investigating the effects of applying SMOTE in conjunction with min-max normalization on the performance of various machine learning classifiers in NIDS. By conducting a detailed comparative analysis, this research seeks to provide valuable insights into the practical benefits and limitations of using SMOTE for handling imbalanced data in NIDS, thereby contributing to the development of more robust and effective IDS.

2. Methodology

The suggested framework is presented in this section. The framework encompasses data pre-processing via the minimum-maximum normalization, addressing the imbalance issue using the synthetic minority oversampling technique (SMOTE), and classification techniques. The specifics of this approach are illustrated in Figure 1.

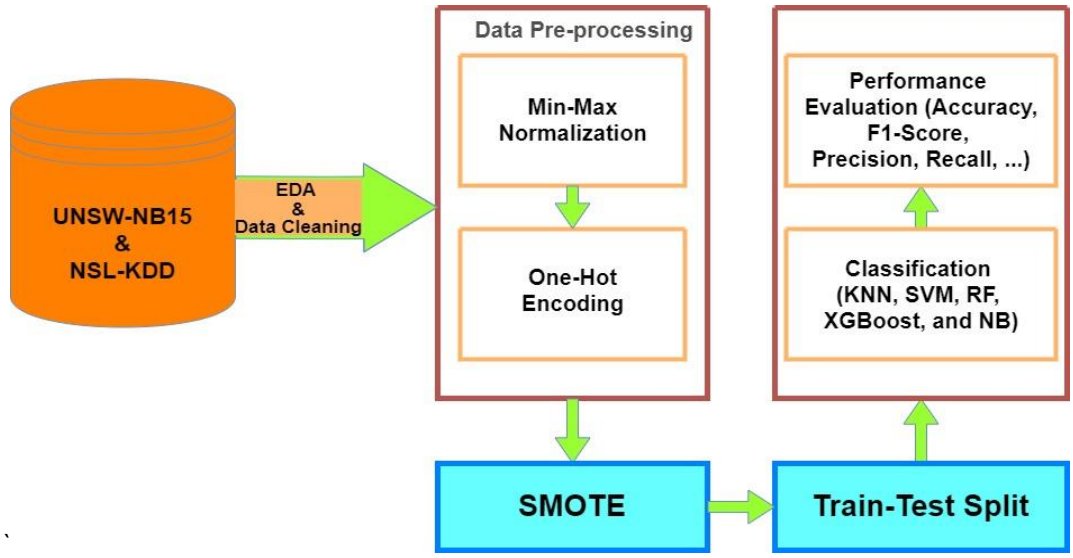

Figure 1: Proposed Framework

2.1 UNSW-NB15 Dataset

The UNSWNB15 dataset employs the IXIA Perfect Storm, an automated attack generating tool, to execute a variety of actual and up-to-date attacks on various servers. In early 2015, traces of TCP dump were collected from network traffic over a period of 31 hours. The UNSWNB15 dataset comprises 49 distinct features that can be categorized into many types, namely flow features, basic features, feature features, temporal features, further generated features, and connection features. Regarding the classification, the dataset consists of two distinct categories of labels: normal and assault. The assault can be further classified into nine distinct categories, including fuzzers, analysis, backdoors, denial of service (DoS), exploits, generic attacks, reconnaissance, shellcode, and worms [25]. Figure 2 shows how the data is distributed in the original UNSW-NB15 dataset (before the application of SMOTE).

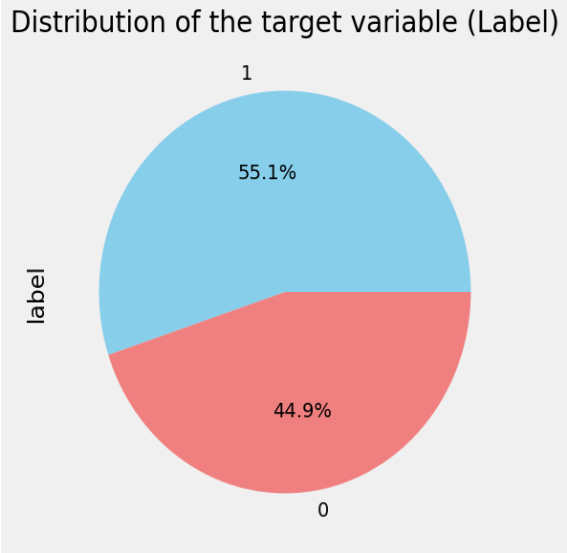

Figure 2: Distribution of target variable in the original UNSW-NB15 dataset.

2.2 NSL-KDD Dataset

The NSL-KDD dataset is a refined version of the KDD Cup 1999 dataset, which was originally created for the purpose of evaluating NIDS. The NSL-KDD dataset addresses some of the significant issues found in the KDD Cup 1999 dataset, such as the presence of a large number of redundant records. These redundant records were removed to ensure a more balanced and representative dataset, making the evaluation of NIDS more accurate and realistic. Each record is described by 41 features, which include basic features of individual connections, content features within a connection, and traffic features computed using a two-second time window [21]. The distribution of target variable in the original NSL-KDD dataset is 51.9% and 48.1% for normal and attack classes respectively.

2.3 Data Preprocessing

This stage represents the preliminary step of the methodology. The purpose of pre-processing data is to convert the raw data into a format that is more convenient and efficient for further processing stages. During the initial stage, the data was explored to gain more insights and subsequently it was cleaned to get rid of duplicate records, handle missing values, and filter out irrelevant features.

2.3.1 Normalization

This is a fundamental step in data modelling where attributes within a dataset are organized and categorized to enhance the relationships between different entity types. Normalization is a process that enhances the flexibility of data. A data set that has undergone a high level of normalization reduces the likelihood of encountering inconsistent data [26]. The process is conducted to center the data around a particular value. Here, data normalization is performed using the min-max method. The implementation of normalization can enhance training efficiency by ensuring that all training data is standardized to a consistent scale, often within the range of 0 and 1. The MinMaxScaler method is widely acknowledged as a prominent strategy for normalizing data. Each component is assigned a numerical value based on a scale where the lowest estimation is represented as 0, the highest value is represented as 1, and all other values are represented as decimals between 0 and 1 [26]. Equation 1 shows how values are normalized using min-max.

$$X_{Normalized} = \frac{X - X_{Min}}{X_{Max} - X_{Min}} \quad (1)$$

2.3.2 Categorical Data Encoding

Some columns in the NSL-KDD and UNSW-NB15 datasets are categorical. These categorical variables were converted into numerical values using One-Hot Encoding (OHE). OHE transforms categorical features into a format that can be provided to machine learning algorithms to improve predictions. This technique creates binary columns for each category, allowing the models to process the categorical information effectively [27]. It can be represented as in equation 2.

$$OHE(x_i) = \begin{cases} 1 & if\ x_i = category \\ 0 & otherwise \end{cases} \quad (2)$$

2.3.3 Synthetic minority oversampling technique

Synthetic Minority Over-sampling Technique (SMOTE) is a highly effective method for addressing imbalanced data sets in classification issues. It specifically addresses the challenges associated with poor prediction of the minority class and the misleading display of classifier performance. One of the primary challenges associated with data sets containing minority data samples is the inadequacy of the training model due to the underrepresentation of the minority class, which is considered to possess the "true population values" [28]. The utilization of the Synthetic Minority Over-sampling Technique (SMOTE) enables the expansion of the decision region pertaining to the minority class, it involves generating additional samples rather than perturbing the existing data as highlighted by Chawla et al. [29]. Figure 3 shows how the data is distributed in both UNSW-NB15 and NSL-KDD datasets after the application of SMOTE.

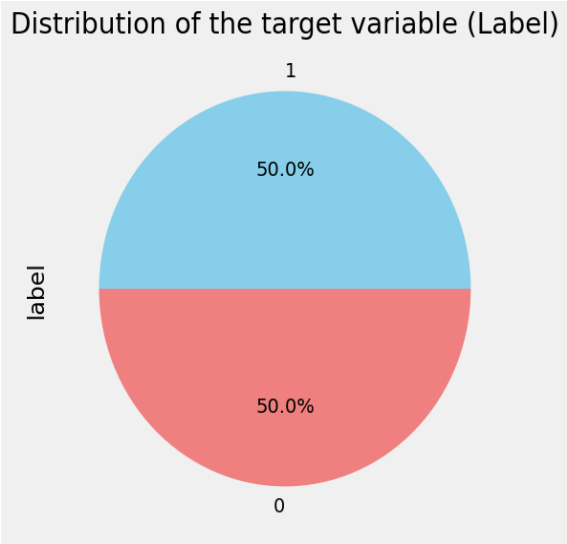

Figure 3: Distribution of target variable after applying SMOTE to the NSL-KDD and UNSWNB15 datasets.

2.4 Classification and performance evaluation

According to Wolpert and Macready [30], no classification algorithm is universally superior. Practitioners often need to try multiple algorithms to determine which one works best for their specific problem. Several algorithms have been widely used in NIDS domain including Random Forest (RF), SVM, KNN, Naïve Bayes (NB), Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), and Multilayer Perceptron (MLP). The classifiers were selected to build the models and achieve the objectives of this research. These models were trained on each of the datasets (UNSW-NB15 and NSL-KDD) with and without the application of SMOTE to compare their performance. The training process involved splitting the data (train-test split), model training – where each classifier was trained on the preprocessed dataset, and finally the trained models were evaluated on the testing set using various performance metrics:

a. Accuracy: The proportion of correctly classified instances out of the total instances.
b. Precision: The proportion of true positive instances out of all instances classified as positive.
c. Recall (Sensitivity): The proportion of true positive instances out of all actual positive instances.
d. F1-Score: The harmonic mean of precision and recall, providing a balance between the two.
e. Confusion Matrix: A matrix showing the true positive, true negative, false positive, and false negative classifications.

3. Result and Discussion

The performance of binary classification on the UNSW-NB15 and NSL-KDD datasets evaluated using various metrics are shown in Table 1 and Table 2 respectively. The results demonstrate the impact of applying the Synthetic Minority Over-sampling Technique (SMOTE) on various machine learning models for network intrusion detection across the two datasets. Importantly, these results must be considered in light of the original class distributions in each dataset.

The NSL-KDD dataset has a near-balanced distribution with 51.9% normal and 48.1% attack instances. This relative balance might explain why the application of SMOTE had minimal impact on most models' performance for this dataset. The consistent high performance across all models, with or without SMOTE, suggests that the original class distribution was already conducive to effective learning.

Table 1: Results for UNSW-NB15 dataset

UNSW-NB15 Dataset									
S/N	**Model**	**Accuracy**		**Precision**		**Recall**		**F1-score**	
		No SMOTE	**SMOTE**	**No SMOTE**	**SMOTE**	**No SMOTE**	**SMOTE**	**No SMOTE**	**SMOTE**
1	**KNN**	0.92	0.92	0.98	0.98	0.88	0.88	0.93	0.93
2	**NB**	0.64	0.65	0.83	1.00	0.43	0.36	0.57	0.53
3	**SVM**	0.90	0.92	0.95	0.96	0.86	0.88	0.90	0.92
4	**XGB**	0.97	0.93	0.97	0.96	0.97	0.92	0.97	0.94
5	**LR**	0.91	0.91	0.95	0.95	0.86	0.87	0.90	0.91
6	**RF**	0.97	0.94	0.98	0.96	0.97	0.92	0.97	0.94
7	**MLP**	0.96	0.96	0.96	0.97	0.96	0.96	0.96	0.97

Table 2: Results for NSL-KDD dataset

NSL-KDD Dataset									
S/N	**Model**	**Accuracy**		**Precision**		**Recall**		**F1-score**	
		No SMOTE	**SMOTE**	**No SMOTE**	**SMOTE**	**No SMOTE**	**SMOTE**	**No SMOTE**	**SMOTE**
1	**KNN**	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
2	**NB**	0.81	0.81	0.99	0.99	0.62	0.62	0.76	0.76
3	**SVM**	0.96	0.96	0.97	0.97	0.93	0.94	0.95	0.96
4	**XGB**	0.99	1	0.99	1	0.99	1	0.99	1
5	**LR**	0.96	0.96	0.96	0.96	0.95	0.95	0.96	0.96
6	**RF**	0.99	1	0.99	1	0.99	0.99	0.99	0.99
7	**MLP**	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

In contrast, the UNSW-NB15 dataset shows a slightly more imbalanced distribution, with 55.1% normal and 44.9% attack instances as depicted in Figure 2. This modest imbalance could account for the more varied effects of SMOTE observed in the results for this dataset.

For the UNSW-NB15 dataset as depicted in Figure 4, the application of SMOTE showed mixed effects across different models and evaluation metrics. The Naive Bayes (NB) classifier exhibited the most significant change, with precision increasing from 0.83 to 1.00 after applying SMOTE, but at the cost of decreased recall and F1-score. This dramatic shift might indicate that SMOTE helped NB better identify certain attack patterns, but potentially at the expense of over fitting to these synthetic examples.

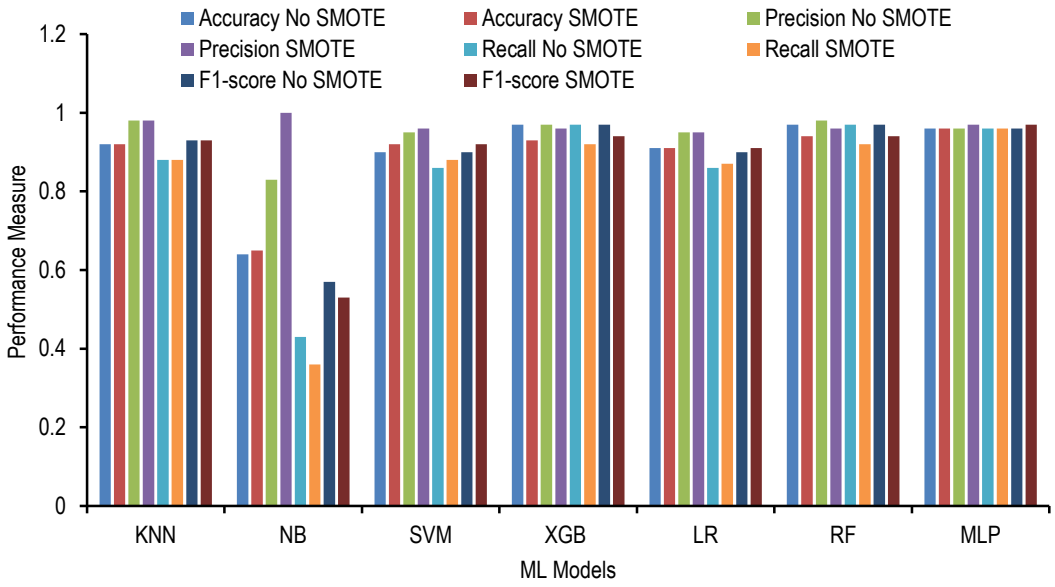

Figure 4: A Clustered column chart depicting the performance of the models built using UNSW-NB15 dataset

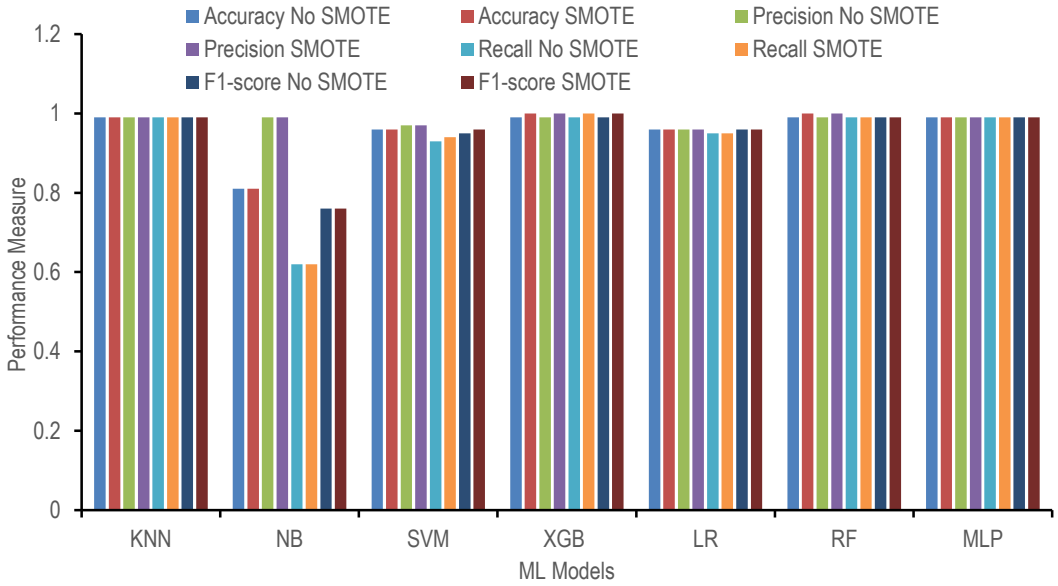

Figure 5: A Clustered column chart depicting the performance of the models built using NSL-KDD dataset

Interestingly, high-performing models like XGBoost (XGB) and Random Forest (RF) on the UNSW-NB15 dataset showed slight decreases in performance across most metrics after applying SMOTE. This suggests that these ensemble methods were already capable of handling the modest class imbalance effectively, and the introduction of synthetic samples may have added unnecessary noise to their decision boundaries.

The near-perfect performance of XGBoost and Random Forest on the NSL-KDD dataset as seen in figure 5, both with and without SMOTE, further underscores how well these models can handle the intrusion detection task when class distributions are relatively balanced. The Multi-Layer Perceptron (MLP) showed remarkable consistency across both datasets and conditions, suggesting it may be a robust choice for network intrusion detection tasks regardless of minor class imbalances or the application of SMOTE.

These findings highlight the importance of considering the original class distribution when deciding

whether to apply balancing techniques like SMOTE. In cases of severe class imbalance, SMOTE may provide significant benefits. However, for datasets with only slight imbalances, as seen here, the benefits of SMOTE may be limited or even counterproductive for some models. The results generally underscore the need for careful evaluation when applying techniques like SMOTE in network intrusion detection. The effectiveness of SMOTE appears to be not only model-dependent but also heavily influenced by the inherent class distribution of the dataset.

4. Conclusion and Future Work

The experiment and analysis of SMOTE's impact on network intrusion detection models revealed several key findings that can be summarized as follows:

a. The effectiveness of SMOTE is heavily dependent on the dataset's original class distribution. For the nearly balanced NSL-KDD dataset (51.9% normal, 48.1% attack), SMOTE had minimal impact on model performance. In contrast, the slightly more imbalanced UNSW-NB15 dataset (55.1% normal, 44.9% attack) showed more varied effects from SMOTE application.
b. Different machine learning algorithms respond differently to SMOTE. Naive Bayes showed the most dramatic changes with SMOTE on the UNSW-NB15 dataset, while ensemble methods like XGBoost and Random Forest sometimes showed slight performance decreases after SMOTE application.
c. High-performing models can effectively handle modest class imbalances without additional techniques. This was particularly evident in the consistent performance of the Multi-Layer Perceptron across all conditions.
d. The application of SMOTE does not universally improve model performance in network intrusion detection. In some cases, it may introduce noise or lead to overfitting, especially when the original class imbalance is not severe.

These findings underscore the importance of careful consideration when applying class balancing techniques in network security applications. The decision to use SMOTE should be based on a thorough understanding of the dataset's characteristics and the specific requirements of the intrusion detection task. Future work could explore threshold levels of class imbalance at which SMOTE becomes beneficial for different types of models in the context of network security applications. Evaluating advanced variants of SMOTE as well as investigating the combination of SMOTE and other techniques like Feature Selection and Feature Extraction could also be explored.

References

[1]. Mebawondu, J. O., Alowolodu, O. D., Mebawondu, J. O. and Adetunmbi, A. O., (2020). Network intrusion detection system using supervised learning paradigm. *Scientific African*, 9, 1-11. https://doi.org/10.1016/j.sciaf.2020.e00497

[2]. Magán-Carrión, R., Urda, D., Díaz-Cano, I. and Dorronsoro, B., (2020). Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Applied Sciences (Switzerland)*, *10*(5), 1–21. https://doi.org/10.3390/app10051775

[3]. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E., (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers and Security*, *28*(1–2), 18–28. https://doi.org/10.1016/j.cose.2008.08.003

[4]. Moustafa, N. and Slay, J., (2015). *UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)*. 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, Canberra, Australia, 1-6. https://doi.org/10.1109/MilCIS.2015.7348942

[5]. Liu, H. and Lang, B., (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. In *Applied Sciences (Switzerland)* 9(20), 1-28. https://doi.org/10.3390/app9204396

[6]. Bhuyan, M. H., Bhattacharyya, D. K. and Kalita, J. K., (2014). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys and Tutorials*, *16*(1), 303–336. https://doi.org/10.1109/SURV.2013.052213.00046

[7]. Camacho, J., Macia-Fernandez, G., Fuentes-Garcia, N. M. and Saccenti, E., (2019). Semi-supervised multivariate statistical network monitoring for learning security threats. *IEEE Transactions on Information Forensics and Security*, *14*(8), 2179–2189. https://doi.org/10.1109/TIFS.2019.2894358

[8]. Taher, K. A., Mohammed Yasin Jisan, B. and Rahman, M. M., (2019). *Network intrusion detection using supervised machine learning technique with feature selection.* 1st International Conference on Robotics, Electrical and Signal Processing Techniques, ICREST 2019, Dhaka, Bangladesh, 643–646. https://doi.org/10.1109/ICREST.2019.8644161

[9]. Belavagi, M. C. and Muniyal, B., (2016). Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. *Procedia Computer Science*, *89*, 117–123. https://doi.org/10.1016/j.procs.2016.06.016

[10]. Saber, M., Chadli, S., Emharraf, M. and El Farissi, I., (2015). Modeling and implementation approach to evaluate the intrusion detection system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9466*, 513–517. https://doi.org/10.1007/978-3-319-26850-7_41

[11]. Schmidhuber, J., (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

[12]. Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P. and Ng, A., (2020).

Cybersecurity data science: an overview from machine learning perspective. *Journal of Big Data*, *7*(1), 1-29. https://doi.org/10.1186/s40537-020-00318-5

[13]. Istiaque, S. M., Khan, A. I., Hassan, Z. A. I. and Waheed, S., (2021). Performance Evaluation of a Smart Intrusion Detection System (IDS) Model. *European Journal of Engineering and Technology Research*, *6*(2), 148–152. https://doi.org/10.24018/ejers.2021.6.2.2371

[14]. Norwahidayah, S., Noraniah, Farahah, N., Amirah, A., Liyana, N. and Suhana, N., (2021). Performances of Artificial Neural Network (ANN) and Particle Swarm Optimization (PSO) Using KDD Cup '99 Dataset in Intrusion Detection System (IDS). *Journal of Physics: Conference Series*, *1874*(1), 1-8. https://doi.org/10.1088/1742-6596/1874/1/012061

[15]. Le, T. T. H., Kim, Y. and Kim, H., (2019). Network intrusion detection based on novel feature selection model and various recurrent neural networks. *Applied Sciences (Switzerland)*, *9*(7), 1-29. https://doi.org/10.3390/app9071392

[16]. Rathore, M. M., Ahmad, A. and Paul, A., (2016). Real time intrusion detection system for ultra-high-speed big data environments. *Journal of Supercomputing*, *72*(9), 3489–3510. https://doi.org/10.1007/s11227-015-1615-5

[17]. Farnaaz, N. and Jabbar, M. A., (2016). Random Forest Modeling for Network Intrusion Detection System. *Procedia Computer Science*, *89*, 213–217. https://doi.org/10.1016/j.procs.2016.06.047

[18]. Shah, B. and Trivedi, B. H., (2015). *Reducing features of KDD CUP 1999 dataset for anomaly detection using back propagation neural network*. International Conference on Advanced Computing and Communication Technologies, ACCT, 2015-April(1), Haryana, India, 247–251. https://doi.org/10.1109/ACCT.2015.131

[19]. Hussain, J. and Lalmuanawma, S., (2016). Feature Analysis, Evaluation and Comparisons of Classification Algorithms Based on Noisy Intrusion Dataset. *Procedia Computer Science*, *92*, 188–198. https://doi.org/10.1016/j.procs.2016.07.345

[20]. Divekar, A., Parekh, M., Savla, V., Mishra, R. and Shirole, M., (2018). *Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives*, Proceedings on 2018 IEEE 3rd International Conference on Computing, Communication and Security, ICCCS 2018, Kathmandu, Nepal, 1–8. https://doi.org/10.1109/CCCS.2018.8586840

[21]. Siddique, K., Akhtar, Z., Aslam Khan, F. and Kim, Y., (2019). KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research. *Computer*, *52*(2), 41–51. https://doi.org/10.1109/MC.2018.2888764

[22]. Prachi, Malhotra, H. and Sharma, P., (2019). Intrusion Detection using Machine Learning and Feature Selection. *International Journal of Computer Network and Information Security*, *11*(4), 43–52. https://doi.org/10.5815/ijcnis.2019.04.06

[23]. Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X. and Dai, K., (2012). An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Systems with Applications*, *39*(1), 424–430. https://doi.org/10.1016/j.eswa.2011.07.032

[24]. Li, Z., Rios, A. L. G., Xu, G. and Trajković, L., (2019). *Machine learning techniques for classifying network anomalies and intrusions,* Proceedings - IEEE International Symposium on Circuits and Systems, 2019-May, Sapporo, Japan, 1–5. https://doi.org/10.1109/ISCAS.2019.8702583

[25]. Ahmad, T. and Aziz, M. N., (2019). Data preprocessing and feature selection for machine learning intrusion detection systems. *ICIC Express Letters*, *13*(2), 93–101. https://doi.org/10.24507/icicel.13.02.93

[26]. Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A. and Padma, V., (2020). *Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification*. Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Tirunelveli, India, 729–735. https://doi.org/10.1109/ICSSIT48917.2020.9214160

[27]. Raschka, S. and Mirjalili, V., (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*, Packt Publishing, Birmingham-Mumbai. 1-741.

[28]. Blagus, R. and Lusa, L., (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14(1),1-13. https://doi.org/10.1186/1471-2105-14-106

[29]. [Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P., (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357. https://doi.org/10.1613/jair.953

[30]. Wolpert, D. H. and Macready, W. G., (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893