JOIV : Int. J. Inform. Visualization, 7(1) - March 2023 241-248

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Impact of Data Balancing and Feature Selection on Machine Learning-based Network Intrusion Detection

Azhari Shouni Barkah [a,b,*], Siti Rahayu Selamat [b], Zaheera Zainal Abidin [b], Rizki Wahyudi [a]

[a] *Department of Informatics, Universitas Amikom Purwokerto, Purwokerto Utara, Banyumas, 55127, Indonesia*
[b] *Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Universiti Teknikal Malaysia Melaka, Melaka, Malaysia*
*Corresponding author: *azhari@amikompurwokerto.ac.id*

***Abstract*— Unbalanced datasets are a common problem in supervised machine learning. It leads to a deeper understanding of the majority of classes in machine learning. Therefore, the machine learning model is more effective at recognizing the majority classes than the minority classes. Naturally, imbalanced data, such as disease data and data networking, has emerged in real life. DDOS is one of the network intrusions found to happen more often than R2L. There is an imbalance in the composition of network attacks in Intrusion Detection System (IDS) public datasets such as NSL-KDD and UNSW-NB15. Besides, researchers propose many techniques to transform it into balanced data by duplicating the minority class and producing synthetic data. Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic (ADASYN) algorithms duplicate the data and construct synthetic data for the minority classes. Meanwhile, machine learning algorithms can capture the labeled data's pattern by considering the input features. Unfortunately, not all the input features have an equal impact on the output (predicted class or value). Some features are interrelated and misleading. Therefore, the important features should be selected to produce a good model. In this research, we implement the recursive feature elimination (RFE) technique to select important features from the available dataset. According to the experiment, SMOTE provides a better synthetic dataset than ADASYN for the UNSW-B15 dataset with a high level of imbalance. RFE feature selection slightly reduces the model's accuracy but improves the training speed. Then, the Decision Tree classifier consistently achieves a better recognition rate than Random Forest and KNN.**

***Keywords*— Intrusion detection; feature selection; imbalance; SMOTE; ADASYN.**

Manuscript received 23 Jul. 2022; revised 26 Dec. 2022; accepted 14 Jan. 2023. Date of publication 31 Mar. 2023.
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.

I. INTRODUCTION

With the current high level of internet usage, network attacks pose a serious threat. The attacks are evolving in line with the advance of computing capacity. To ensure the safety of data communication, defensive action must be taken. Therefore, researchers in network defense are working hard all the time to encounter new types of attacks.

The important task in network security is to recognize the type of attack. The attack dataset is evolving due to the introduction of new attack techniques. Even though the indicator variables (features) are similar, the type of network intrusion is evolving. Network security researchers provide datasets allowing the machine to recognize attack classes automatically. Many researchers provide KDD99 and NSL-KDD [1], [2], [3], [4], while other researchers provide UNSW-NB15 [2], [3], [5], [6], [7], [8] and Liu provide CICIDS2017 [3], and others provide CICDDS001 [9], [10].

Based on publicly available datasets, many researchers develop methods and tools to recognize network intrusions, such as random forest, decision tree, logistic regression, KNN, and ANN. The common problems identified in many academic papers are that certain classes of attacks have rarely happened. Therefore, the available data is limited, while other popular attacks, such as denial of service (DDOS), are dominated by network attacks. Natural data imbalances are observed in most of the IDS datasets.

The quality of the dataset is very important in the classification process, and certain imbalance classes dominate the dataset. According to Johnson and Khoshgoftaar [11], the existing classification model has a higher capability of recognizing the majority class and tends to fail to recognize the minority classes. The imbalance of class problems has been identified as the cause of low classification performance [12]. Therefore, it needs to run a pre-processing activity to make the training data into equal samples in each class [13].

It is possible to improve the dataset's quality by balancing the classes. This can be accomplished through resampling, which is an oversampling technique used to increase the number of minority classes [14]. The oversampling approach has two techniques: random oversampling, where the technique only duplicates data, which causes overfitting, and synthesis minority oversampling, which duplicates data by synthesizing minority class data so it can solve the overfitting problem that occurs in random oversampling [15], [16]. Previous research proposes balancing datasets to improve classification performance. SMOTE was proposed by some previous studies [4], [8], [17], [18], SMOTE combined with undersampling techniques [2], [7], [10], SMOTE with optimization techniques [1], [5], [6], [9], ADASYN [3] combine with undersampling [7].

To improve attack detection on machine learning models, balancing the data and reducing the features are two steps to ensure the quality of the training data. Once the data is ready, machine learning algorithms develop a model. The IDS dataset consists of a limited number of columns (features); therefore, it is still considered a simple model. Some research papers proposed using simple and light computation algorithms such as KNN [1], [8], [9], [17], [18], SVM [1], [9], Logistic Regression (LR) [8], [9], Random Forest (RF), Decision Tree (DT) [8], [9], [18], and ANN [4], [7], [8], [19]. In general, the accuracy achieved by classic machine learning algorithms ranges from 39% to 99.57%.

The second problem is the features of the datasets. Not all features are relevant to the class label. Some features are intercorrelated, and therefore redundant information appears in the input. Redundant features make training take longer without making the model better. Researchers are working to solve this issue in various ways. Some researchers use statistical measures like intercorrelation between features like Information Gain (IG) [8], [20]. Principal component analysis (PCA) and Linear discriminant analysis (LDA) were proposed by Ibrahimi and Ouaddane [21]. Recursive feature elimination (RFE) research was conducted by some other previous studies [22], [23], [24]. The most commonly used feature selection method is information gain, a filter-based feature selection [25], [26]. Information gain starts with a basic attribute ranking, removes the background noise caused by unimportant features, and finds the feature with the most information about a certain class [20]. Calculating a feature's entropy is one way to evaluate which feature is superior to others. The entropy of a system is a measure of uncertainty that can be used to get a quick idea of how the system's characteristics are spread out [27]. PCA is the most popular method due to its computations' adaptability and approach's reversibility. PCA is useful for solving dimensionality reduction problems [28]. The PCA is accomplished by removing higher-dimensional space less significant attributes [29]. According to Gupta and Agrawal [24], using RFE in the training process can remove useless and redundant features to get higher accuracy and minimize training time.

IDS performance is improved by using modern and up-to-date datasets. As a result, modern network normal and attack operations necessitate the development of new cutting-edge datasets to evaluate IDS more efficiently and accurately. Using the UNSW-NB15 dataset, this research created a network attack detection framework [30], [31]. This dataset includes recent attacks. IDS benchmarked datasets previously used KDD99 and NSL-KDD. Aging datasets are less useful for understanding today's network traffic [32], [33]. Despite limited work, several researchers have used the new UNSW-NB15 data set to detect attacks.

This paper aims to determine the impact of balancing techniques (SMOTE, ADASYN, and feature selection using RFE) on the classification of results on machine learning-based IDS. Researchers also evaluate decision trees (DT), random forest (RF), logistic regression (LR), and K-nearest neighbor (KNN) to classify multiclass network attacks. This experiment will be carried out with two scenarios defined in the research framework. First, the selection feature is applied after the data has been balanced, and the second starts with feature selection and then initiates data balancing.

II. Materials and Methods

This research aims to observe the impact of balancing the data and feature selection toward the performance of the classification. Fig. 1 explains the first research framework, where the balancing dataset was carried out before the feature selection.

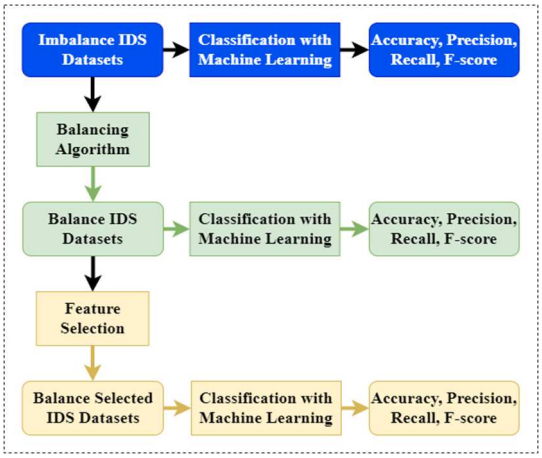

Fig. 1 Research Framework Balancing before RFE

This research also includes an experiment where the feature selection is carried out before the dataset balancing. Fig. 2 shows the second research framework where feature selection is carried out before imbalance dataset handling.

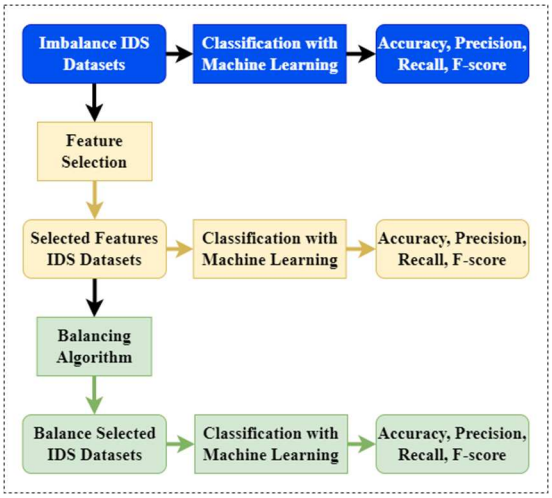

Fig. 2 Research Framework for RFE before imbalance handling.

A. Pre-processing

Pre-processing aims to prepare the dataset to enter the subsequent process. It includes data standardization and normalization. We used data standardization to transform the data from a normal distribution to a standard normal distribution because the dataset contained characteristics with a wide range of possible values [8]. Because of this, we had to adjust the data to follow a standard normal distribution instead of a normal distribution. c. The formula for making a standard score, also called a z-score, is as follows:

$$z = \frac{(x-\mu)}{\sigma} \quad (1)$$

Where x is the sample of data, μ represents the average, and σ represents the standard deviation [34]. During the data normalization process, the value of each continuous attribute is scaled so that the results of the attributes do not overlap [35]. This is done by assigning a value between 0 and 1 to the value of each continuous attribute and giving that value. This inquiry used the normalizer class that is available in Python programming. The utilization of this class paves the way for the successful normalization of a dataset.

B. Dataset UNSW-NB15

The Australian Centre put together the UNSW-NB15 dataset for Cyber Security [36]. The details of the dataset are presented in Table I. It covers nine attack types, which have a total of 49 attributes. The following categories of assaults are included in this dataset: Worms, shellcode, reconnaissance, port scans, generic, backdoor, DoS, exploits, and fuzzers [37].

TABLE I
DESCRIPTION OF UNSW-NB15 DATASET

Instances in training set	Instances in testing set	Attack category
175341 records	82332 records	9 attacks

The distribution of the training and testing dataset is shown in table II. The distribution shows that the data distribution is a high degree of imbalance.

TABLE II
DESCRIPTION OF UNSW-NB15 DATASET

Class	No. of Records Training	No. of Records Testing
Normal	56000	37000
Fuzzers	18184	6062
Analysis	2000	677
Backdoors	1746	583
DoS	12264	4089
Exploits	33393	11132
Generic	40000	18871
Reconnaissance	10491	3496
Shellcode	1133	378
Worms	130	44

C. Imbalance Dataset Treatment

This research observes SMOTE and ADASYN as two popular synthetic oversampling techniques. Researchers aim to identify the best synthetic data to serve the classification task of the highly degree-imbalanced UNSW-NB15 dataset.

According to Chawla et al. [38], it has been suggested that SMOTE be used as an oversampling method. The new synthetic data found in the underrepresented group was made using the oversampling method. SMOTE does not turn little amounts of data into a large number when it generates new data; rather, it creates synthetic data [10], [39], [40]. Generating new data by randomly picking cases from a minority class near the feature space requires significant labor. Then produce a new data point for the minority class by utilizing a linear combination of two samples from the minority class that are comparable. The newly obtained point value is interpreted in the same way between the instances that belong to the minority and those of their respective nearest neighbors. The position of the general data point in relation to the class that constitutes the majority is ignored by SMOTE. Because of this, the class may begin to overlap or become noisy [39].

A previous study by He et al. [41] suggested ADASYN as an oversampling method for underrepresented classes. Using the method of pseudo-probabilistic oversampling, new synthetic data were constructed and evaluated [40]. ADASYN calculates the weight distribution for each point in the various minority classes based on the difficulty that each minority group has in learning the material [40], [42]. If the difficulty level goes up, more synthesis data will be made than if the level of difficulty goes down.

D. Recursive Feature Elimination (RFE)

Recursive feature elimination is one of the simple methods to select only important features in the input space [24]. Unlike principal component analysis (PCA), RFE deletes the possible unnecessary columns. It may reduce the noise in the input, but with a risk of losing important information. However, PCA can reduce the input dimension with a transformation; therefore, they keep the information as much as possible with the risk of maintaining the noise in the input space. In this research, we use RFE as the feature selection technique. RFE selects features based on how they affect a particular model's performance. RFE works iteratively until the optimal number of features remains.

E. Classification Algorithm

Machine learning algorithms are responsible for forming a model to recognize the training data pattern and the new unknown class data. To do that, we choose four algorithms: KNN, RF, LR, and DT. Random Forest (RF) is a supervised machine learning architecture that may be used for classification and regression issues [43]. Random Forest is an ensemble classifier utilized to produce more accurate classification results [42]. It is simple to use, generates a decision forest using a Decision tree, and solves problems in this manner. For this purpose, it generates a random collection of trees. Throughout the procedure, many Decision trees are trained to produce the most accurate classification. The majority of the time, even without the usage of a hyperparameter, it is possible to obtain acceptable results. It is one of the most used techniques because it delivers accurate, rapid answers even for mixed, incomplete, and noisy datasets. RF has been shown to produce fewer classification errors compared to other classifiers. When building different trees in

RF, the optimum nodes for splitting selection are made by randomization to maximize efficiency [44].

A decision tree (DT) is a supervised learning method used to classify numerical and class data. In the DT algorithm, the classes' labels are stored on leaf nodes, and attributes are evaluated on interior nodes of the tree. The branches show the results of the evaluations of the attributes [45]. Methods of attribute selection are utilized in the process of identifying nodes. It has a goal variable that has already been predefined. In addition, it consists of leaf nodes maintained by decision-making processes to accomplish one of the top-down objectives of the algorithm structure [46]. It processes enormous volumes of data quickly due to its straightforward architecture, which allows it to do so. There are situations in which more complicated trees are required to cope with the categorization of datasets. In these kinds of circumstances, decision trees get more complicated, and achieving any of the goals becomes more challenging. Another issue that might arise with decision tree algorithms is overfitting. In order to find a solution to this issue, certain leaf nodes of the DT will need to be removed. An entropy and information gain must be calculated on a decision tree [47].

An example of a supervised learning algorithm is the K Nearest Neighbor (KNN) Algorithm. It is unique among supervised learning algorithms in that it does not include a stage for training the model [48]. For K-nearest neighbors to function, new data points must first be connected to the training set's existing data points before being given a value based on the strength of that connection. The forecast is made based on comparable features. In KNN, the Euclidean, Manhattan, or Hamming distances can be used to determine the separation between a set of test data and each record of training data [49]. After that, the rows are arranged in descending order based on the value of the distance. The first K rows from the top of those rows are the ones that are chosen. Classes are allocated to the test points in accordance with the rows' classes in which they occur the most frequently.

Logistic Regression (LR) is another classification model that uses a linear algebra approach to classify data [50]. The LR model is based on the likelihood of a class instance. This probability is calculated by applying a logistic function to each class data set. The logistic function is derived from linear regression, in which a linear function represents the probability of a specific data point in the class. However, s function, the logistic regression model can also be represented by the logit function [50]. These are commonly known as logit functions, and their classification is known as log-linear classification [50].

F. Classification Evaluation

In order to evaluate the effectiveness of different machine learning algorithms, a confusion matrix is typically utilized. The values of False Positive (FP), True Negative (TN), False Negative (FN), and True Positive (TP) are combined in this matrix in order to provide a variety of different metrics. These metrics are created by combining the values of TN, TP, and FN, FP [51]. The following is a list of some of the performance metrics that may be used to evaluate models by making use of a confusion matrix: The degree to which a model's estimated value corresponds accurately or closely to the model's real or correct value is referred to as its accuracy, and it is measured as the percentage of the total number of samples that are correctly categorized [52]. Formula eq. 2 is used to calculate the accuracy of the model.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FN+FP}} \tag{2}$$

Precision indicates what percentage of the relevant occurrences from the chosen instances genuinely exhibit better characteristics [53]. To calculate precision using the formula eq.3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3}$$

Recall, also known as TPR represents True Positive Rate, which is a calculation that determines the percentage of genuine positives that are accurately detected [54]. To find recall, use the formula eq.4:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{4}$$

The harmonic mean of precision and recall, which combines the weighted average of precision and recall, is what is meant by the F1 score. The F1-score is calculated using the following equation, eq.5:

$$\text{F1} - \text{Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision+Recall}} \tag{5}$$

III. Result and Discussion

The flow of the experimental design is shown in Fig. 1 and Fig. 2. The training and testing data is adopted from UNSW-NB15 original dataset. Before being tested with the dataset model, it is processed first through the standardized and normalized pre-processing stages. After the pre-processing was carried out, we carried out two experiment scenarios with different balancing and feature selection tasks. We evaluate the classification result in each dataset modification due to balancing and feature simplification. Then the data set is tested using the RF, DT, LR, and KNN models.

A. Balancing Prior to Feature Selection

The first scenario, handling the imbalance dataset, was carried out by utilizing SMOTE and ADASYN. The result of the balanced dataset with full features (columns) consists of 56.000 rows and 44 columns in training data. After the dataset with synthetics data is created, the recursive feature elimination (RFE) is responsible for reducing the feature and maintaining the information held by the input side of the training dataset. Fig. 3 shows the accuracy of multiclass classification.

As can be seen in Table III, the decision tree achieves the best accuracy, followed by Random Forest, K-Nearest Neighbors, and Logistic Regression. Regarding training time, the model that recorded the best performance was DT with 7 seconds, followed by RF, LR, and KNN, which had the longest training time with 521 seconds.

The classification performance on the balanced dataset with complete features is better than that imbalance dataset. However, it was paid for by the heavier computing load, as indicated by the slower training time. As can be seen in Table IV and Table V, the computation time of the balanced dataset both in SMOTE and ADASYN are much longer compared to that of the imbalance dataset in Table III. This is caused by

the impact of the number of rows in the balanced dataset much higher than the original data. A higher number of balanced samples usually lead to a better model but need a longer training duration due to the iterative process.

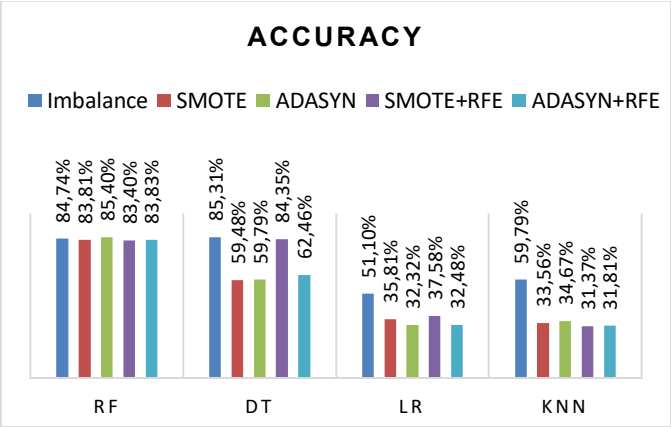

Fig. 3 Accuracy of Classification on First Experiment Scenario

TABLE III
CLASSIFICATION RESULT IMBALANCE DATA

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	84.74%	86.19%	84.74%	85.05%	36
DT	85.31%	88.49%	85.31%	86.69%	7
LR	51.10%	46.81%	51.10%	48.26%	65
KNN	59.79%	68.74%	59.79%	62.84%	521

TABLE IV
CLASSIFICATION RESULT OF BALANCE DATA SMOTE

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	83.81%	87.09%	83.81%	84.70%	200
DT	59.48%	66.52%	59.48%	57.21%	34
LR	35.81%	63.18%	35.81%	41.71%	112
KNN	33.56%	69.86%	33.56%	41.88%	1609

TABLE V
CLASSIFICATION RESULT OF BALANCE DATA ADASYN

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	85.40%	89.40%	85.40%	86.94%	211
DT	59.79%	66.25%	59.79%	60.86%	38
LR	32.32%	60.65%	32.32%	39.63%	103
KNN	34.67%	66.01%	34.67%	43.57%	1411

TABLE VI
CLASSIFICATION RESULT OF BALANCE DATASET USING SMOTE AND AFTER FEATURE SELECTION

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	83.40%	86.45%	83.40%	84.04%	97
DT	84.35%	89.43%	84.35%	86.30%	6
LR	37.58%	63.61%	37.58%	45.84%	104
KNN	31.37%	28.85%	31.37%	28.16%	7

Tables VI and VII show the classification result of a selected feature on the balanced dataset. RFE has taken its role in simplifying the dataset by eliminating the potential noise in the synthetic dataset. The number of columns decreased from 44 to 13, reducing lot training duration, as can be seen in table VI compared to table VII. The accuracy, recall, precision, and F1-score are slightly decreased. On the other hand, the training duration is much faster due to the data reduction.

TABLE VII
CLASSIFICATION RESULT OF BALANCE DATASET USING ADASYN AND AFTER FEATURE SELECTION

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	83.83%	87.30%	83.83%	84.82%	110
DT	62.46%	77.28%	62.46%	61.44%	8
LR	32.48%	48.75%	32.48%	34.92%	110
KNN	31.81%	29.74%	31.81%	29.01%	8

B. Feature Selection Prior to Balancing

The feature selection was carried out in the second scenario before the balancing task. The recursive feature elimination gets the original dataset with 13 features remaining in the list. Fig. 4 shows the accuracy of multiclass classification.

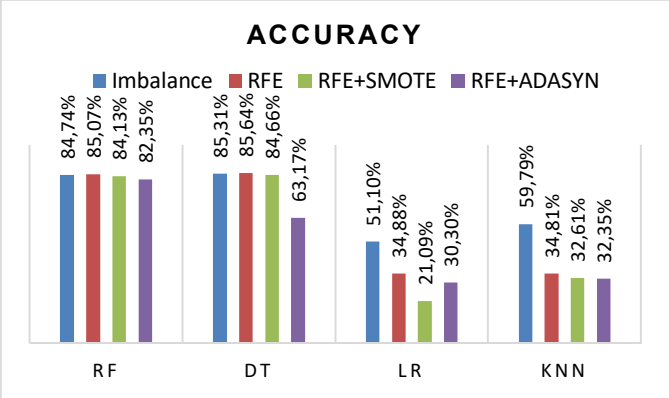

Fig. 4 Accuracy classification of the second scenario

Comparing the classifier's performance, it is obvious that logistic regression gets the worst capability to capture the pattern on the training dataset. Among the four compared algorithms, KNN and LR show low accuracy. Decision Tree is generally better than its competitors and recorded as the quickest to execute. Decision Tree achieves slightly better accuracy, recall, precision, and F1-score than Random Forest except in the balanced dataset with ADASYN.

TABLE VIII
CLASSIFICATION RESULT OF IMBALANCE DATA AFTER FEATURE SELECTION (RFE)

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	85.07%	87.24%	85.07%	85.69%	20
DT	85.64%	88.58%	85.64%	86.87%	2
LR	34.88%	37.95%	34.88%	32.11%	82
KNN	34.81%	26.57%	34.81%	29.85%	5

Table VIII shows the simplified imbalance dataset after RFE was executed to the original training dataset. The recognition rate slightly decreased compared to the original imbalance dataset. However, it was paid for by the efficiency in computing load. As can be seen in table VIII, the computation time halved in the selected features classification compared to the imbalance with full features in table IX.

TABLE IX
CLASSIFICATION RESULT OF BALANCE DATASET USING SMOTE AND AFTER FEATURE SELECTION

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	84.13%	87.37%	84.13%	84.95%	145
DT	84.66%	88.07%	84.66%	86.03%	18
LR	21.09%	44.39%	21.09%	25.84%	147
KNN	32.61%	27.77%	32.61%	29.14%	8

Tables IX and X show the balance dataset's classification after synthetics data created by SMOTE and ADASYN. Although the accuracy, precision, and recall did not touch the achieved value on the complete dataset, the gap got smaller. It shows that the impact of selecting a subset of the features slightly decreases the classifier performance but is still acceptable due to the small gap. Balancing the dataset give a positive impact on the classifier's performance. However, it leads to slower training time. It is acceptable since the training of the model has no hardware limitation. We can train the model in high-performance computing infrastructures. Once the model is trained, it can be implemented in various lower computing resource devices without sacrificing speed.

TABLE X
CLASSIFICATION RESULT OF BALANCE DATASET USING ADASYN AND AFTER FEATURE SELECTION

Model	Accuracy	Precision	Recall	F1-score	Times (s)
RF	82.35%	86.27%	82.35%	83.86%	157
DT	63.17%	77.74%	63.17%	63.27%	20
LR	30.30%	47.78%	30.30%	35.20%	124
KNN	32.35%	33.56%	32.35%	28.91%	8

Table X presents the classifier performance on the balanced dataset. The balanced dataset was produced by ADASYN technique. According to previous research [55], it is observed that SMOTE overperforms ADASYN in helping the classifier improve its accuracy in a high level of imbalance class. UNSW-NB15 dataset also has a high level of imbalance, as shown in Table III. Therefore, it is reasonable to see a better classification result on SMOTE balanced dataset. In [37], they reported comparing ADASYN and SMOTE in 5 binary class classifications. According to their result, in 5 datasets under their investigation, they found that ADASYN balance dataset leads to better classification performance. Their dataset is imbalance, but the degree of imbalance is not as high as UNSW-NB15. Our finding is in line with [55], where the SMOTE balanced dataset achieved better classification performance as the degree of imbalance improved.

In the experiment, we discovered that SMOTE provides better synthetic data for the UNSW-B15 dataset with high imbalances levels than ADASYN. DT has the maximum accuracy, at 84%, according to Tables VI and VII, whereas KNN has the lowest accuracy, at 31%, when balancing data with feature selection. Another interesting finding was that, even though KNN indicated almost the same time, it produced poor measurement findings. The best time consumption was determined to be between 6 and 8 seconds and was acquired by DT. According to tables IX and X, where feature selection trials were carried out with data balancing, DT obtained the maximum accuracy with a value of 84.6%, while LR obtained the lowest gain with 21.09%. Other findings in Tables 9 and 10 show DT achieves better measurement results than KNN, but KNN reduces training time, with a value of 8 seconds compared to a DT value of 18 to 20 seconds.

The order of pre-processing play's an important role in terms of reducing computing complexity. It was reflected in training time. First, the balancing dataset carries out before the feature selection task. It leads to a balanced dataset with much more rows and complete features. More columns highly affect to the data size, and it is directly slowing down the training.

The findings of this study have to be seen in the light of some limitations. We only conducted experiments related to imbalanced data using UNSW-NB15. However, it is quite difficult to determine whether the results depend on a single dataset. In other words, it is necessary to prove whether it can be applied to other imbalanced data on other datasets.

IV. Conclusion

The type of intrusion to the network (attack) is naturally imbalance. Popular attacks like DDOS dominate attack incidents and are reflected in the IDS dataset. The machine learning model cannot work well in the imbalanced training dataset, leading to failure to recognize the minority class. Imbalance dataset handling eases the imbalance problem and improve the classifier performance. We observed the performance of four classic machine learning algorithms and found that Decision Tree consistently achieved the best accuracy compared to RF, KNN, and LR. The result was obtained using UNSW-NB15 IDS dataset. SMOTE and ADASYN handle the imbalanced dataset by creating synthetic data to increase the number of minority samples. On UNSW-NB15, a high-level imbalanced dataset, SMOTE provides better synthetics data for the classification task. In this research, we also observed process order's impact and found that carrying out feature selection before creating synthetic data leads to more efficient computation without sacrificing the recognition rate. Currently, we cover one publicly available dataset. In future research, observing the impact of the imbalance handling mechanism and feature selection impact on the recognition rate on various datasets would be interesting. Experimenting with more advanced synthetics data creation algorithms such as GAN, CGAN is also our future direction. Implementing neural networks and deep learning algorithms to reduce data dimension without losing too much information, like Autoencoder, would be a future research direction.

Acknowledgment

The authors thank the financial support from Universitas AMIKOM Purwokerto; and Fakulti Teknologi Maklumat dan Komunikasi (FTMK), and Universiti Teknikal Malaysia Melaka (UTeM) for their assistance in this research.

References

[1] J. H. Seo and Y. H. Kim, "Machine-learning approach to optimize smote ratio in class imbalance dataset for intrusion detection," *Computational Intelligence and Neuroscience*, vol. 2018. 2018. doi: 10.1155/2018/9704672.

[2] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, 2020.

[3] J. Liu, Y. Gao, and F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," *Comput Secur*, 2021.

[4] R. Ahsan, W. Shi, and J. P. Corriveau, "Network intrusion detection using machine learning approaches: Addressing data imbalance," *IET Cyber-Physical Systems: Theory and Applications*, vol. 7, no. 1, pp. 30–39, Mar. 2022, doi: 10.1049/cps2.12013.

[5] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks*, vol. 177. 2020. doi: 10.1016/j.comnet.2020.107315.

[6] X. Jiao and J. Li, "An Effective Intrusion Detection Model for Class-imbalanced Learning Based on SMOTE and Attention Mechanism," *2021 18th International Conference on Privacy, Security and Trust, PST 2021*. 2021. doi: 10.1109/PST52912.2021.9647756.

[7] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *Journal of Big Data*. journalofbigdata.springeropen.com, 2021. doi: 10.1186/s40537-020-00390-x.

[8] H. A. Ahmed, A. Hameed, and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Computer Science*, vol. 8. 2022. doi: 10.7717/PEERJ-CS.820.

[9] D. Gonzalez-Cuautle *et al.*, "Synthetic minority oversampling technique for optimizing classification tasks in botnet and intrusion-detection-system datasets," *Applied Sciences (Switzerland)*, vol. 10, no. 3. 2020. doi: 10.3390/app10030794.

[10] S. Al and M. Dener, "STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment," *Comput Secur*, 2021.

[11] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J Big Data*, vol. 6, no. 1, p. 27, Dec. 2019, doi: 10.1186/s40537-019-0192-5.

[12] Shuo Wang and Xin Yao, "Multiclass Imbalance Problems: Analysis and Potential Solutions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012, doi: 10.1109/TSMCB.2012.2187280.

[13] C. Romero, J. R. Romero, and S. Ventura, "A Survey on Pre-Processing Educational Data," 2014, pp. 29–64. doi: 10.1007/978-3-319-02738-8_2.

[14] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," *Inf Sci (N Y)*, vol. 465, pp. 1–20, Oct. 2018, doi: 10.1016/j.ins.2018.06.056.

[15] G. Wei, W. Mu, Y. Song, and J. Dou, "An improved and random synthetic minority oversampling technique for imbalanced data," *Knowl Based Syst*, vol. 248, p. 108839, Jul. 2022, doi: 10.1016/j.knosys.2022.108839.

[16] S. Feng, J. Keung, X. Yu, Y. Xiao, and M. Zhang, "Investigation on the stability of SMOTE-based oversampling techniques in software defect prediction," *Inf Softw Technol*, vol. 139, p. 106662, Nov. 2021, doi: 10.1016/j.infsof.2021.106662.

[17] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain, and A. Nawaz, "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, 2019.

[18] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8. pp. 32150–32162, 2020. doi: 10.1109/ACCESS.2020.2973219.

[19] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A New Intrusion Detection System Based on Fast Learning Network and Particle Swarm Optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018, doi: 10.1109/ACCESS.2018.2820092.

[20] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.

[21] K. Ibrahimi and M. Ouaddane, "Management of intrusion detection systems based-KDD99: Analysis with LDA and PCA," in *Proceedings - 2017 International Conference on Wireless Networks and Mobile Communications, WINCOM 2017*, 2017. doi: 10.1109/WINCOM.2017.8238171.

[22] N. V Sharma and N. S. Yadav, "An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers," *Microprocess Microsyst*, vol. 85, p. 104293, Sep. 2021, doi: 10.1016/j.micpro.2021.104293.

[23] S. Ustebay, Z. Turgut, and M. A. Aydin, "Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier," *... congress on big data, deep ...*, 2018.

[24] A. R. B. Gupta and J. Agrawal, "Machine Learning-Based Intrusion Detection System with Recursive Feature Elimination," 2021, pp. 157–172. doi: 10.1007/978-981-33-4305-4_13.

[25] T. A. Alhaj, M. M. Siraj, A. Zainal, H. T. Elshoush, and F. Elhaj, "Feature Selection Using Information Gain for Improved Structural-Based Alert Correlation," *PLoS One*, vol. 11, no. 11, p. e0166017, Nov. 2016, doi: 10.1371/journal.pone.0166017.

[26] Z. Karimi, M. Mansour Riahi Kashani, and A. Harounabadi, "Feature Ranking in Intrusion Detection Dataset using Combination of Filtering Methods," *Int J Comput Appl*, vol. 78, no. 4, pp. 21–27, Sep. 2013, doi: 10.5120/13478-1164.

[27] P. Bereziński, B. Jasiul, and M. Szpyrka, "An Entropy-Based Network Anomaly Detection Method," *Entropy*, vol. 17, no. 4, pp. 2367–2408, Apr. 2015, doi: 10.3390/e17042367.

[28] K. Keerthi Vasan and B. Surendiran, "Dimensionality reduction using Principal Component Analysis for network intrusion detection," *Perspect Sci (Neth)*, vol. 8, pp. 510–512, Sep. 2016, doi: 10.1016/j.pisc.2016.05.010.

[29] P. Nskh, M. N. Varma, and R. R. Naik, "Principle component analysis based intrusion detection system using support vector machine," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, May 2016, pp. 1344–1350. doi: 10.1109/RTEICT.2016.7808050.

[30] B. A. Tama and K.-H. Rhee, "An in-depth experimental study of anomaly detection using gradient boosted machine," *Neural Comput Appl*, vol. 31, no. 4, pp. 955–965, Apr. 2019, doi: 10.1007/s00521-017-3128-z.

[31] N. Belhadj aissa, M. Guerroumi, and A. Derhab, "NSNAD: negative selection-based network anomaly detection approach with relevant feature subset," *Neural Comput Appl*, vol. 32, no. 8, pp. 3475–3501, Apr. 2020, doi: 10.1007/s00521-019-04396-2.

[32] H. N. Viet, Q. N. Van, L. L. T. Trang, and S. Nathan, "Using Deep Learning Model for Network Scanning Detection," in *Proceedings of the 4th International Conference on Frontiers of Educational Technologies - ICFET '18*, 2018, pp. 117–121. doi: 10.1145/3233347.3233379.

[33] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Comput*, vol. 23, no. 2, pp. 1397–1418, Jun. 2020, doi: 10.1007/s10586-019-03008-x.

[34] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," *IEEE Access*, vol. 7. pp. 42210–42219, 2019. doi: 10.1109/ACCESS.2019.2904620.

[35] D. Gupta, S. Singhal, S. Malik, and A. Singh, "Network intrusion detection system using various data mining techniques," in *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, May 2016, pp. 1–6. doi: 10.1109/RAINS.2016.7764418.

[36] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

[37] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, doi: 10.1080/19393555.2015.1125974.

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

[39] J. H. Lee and K. H. Park, "GAN-based imbalanced data intrusion detection system," *Pers Ubiquitous Comput*, vol. 25, no. 1, pp. 121–128, Feb. 2021, doi: 10.1007/s00779-019-01332-y.

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

[41] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 1322–1328. doi: 10.1109/IJCNN.2008.4633969.

[42] J. Liu, Y. Gao, and F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," *Comput Secur*, vol. 106, p. 102289, Jul. 2021, doi: 10.1016/j.cose.2021.102289.

[43] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, Apr. 2016, doi: 10.1016/j.isprsjprs.2016.01.011.

[44] J. Jiang, Q. Wang, Z. Shi, B. Lv, and B. Qi, "RST-RF: A Hybrid Model based on Rough Set Theory and Random Forest for Network Intrusion Detection," in *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy*, Mar. 2018, pp. 77–81. doi: 10.1145/3199478.3199489.

[45] S. Afraei, K. Shahriar, and S. H. Madani, "Developing intelligent classification models for rock burst prediction after recognizing significant predictor variables, Section 2: Designing classifiers," *Tunnelling and Underground Space Technology*, vol. 84, pp. 522–537, Feb. 2019, doi: 10.1016/j.tust.2018.11.011.

[46] G. H. Nicholas Frosst, "Distilling a neural network into a soft decision tree," 2017, doi: https://doi.org/10.48550/arXiv.1711.09784.

[47] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8. pp. 32150–32162, 2020. doi: 10.1109/ACCESS.2020.2973219.

[48] A. Vijay, K. Patidar, M. Yadav, and R. Kushwah, "An efficient intrusion detection mechanism based on particle swarm optimization and KNN," *ACCENTS Transactions on Information Security*, vol. 5, no. 20, pp. 36–41, Oct. 2020, doi: 10.19101/TIS.2020.517003.

[49] S. Jain, S. C. Jain, and S. Vishwakarma, "Analysis and Prediction of Customers' Reviews with Amazon Dataset on Products," 2020, pp. 445–456. doi: 10.1007/978-981-15-0936-0_48.

[50] A. Sharma and P. K. Mishra, "Performance analysis of machine learning based optimized feature selection approaches for breast cancer diagnosis," *International Journal of Information Technology*, vol. 14, no. 4, pp. 1949–1960, Jun. 2022, doi: 10.1007/s41870-021-00671-5.

[51] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst Appl*, vol. 57, pp. 117–126, Sep. 2016, doi: 10.1016/j.eswa.2016.03.028.

[52] P. Lin, K. Ye, and C.-Z. Xu, "Dynamic Network Anomaly Detection System by Using Deep Learning Techniques," 2019, pp. 161–176. doi: 10.1007/978-3-030-23502-4_12.

[53] B. Roy and H. Cheung, "A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Nov. 2018, pp. 1–6. doi: 10.1109/ATNAC.2018.8615294.

[54] S. A. Ludwig, "Intrusion detection of multiple attack classes using a deep neural net ensemble," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Nov. 2017, pp. 1–7. doi: 10.1109/SSCI.2017.8280825.

[55] J. Brandt and E. Lanzén, "A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification," 2020.