Journal of Engineering Science and Technology
Vol. 16, No. 2 (2021) 1342 - 1355

CLUSTERING UNDER-SAMPLING DATA FOR IMPROVING THE PERFORMANCE OF INTRUSION DETECTION SYSTEM

MOHAMMAD NASRUL AZIZ[1], TOHARI AHMAD[2,*]

[1]Quality Assurance Board, Universitas Airlangga, Surabaya, Indonesia
[2]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
*Corresponding Author: tohari@if.its.ac.id

Abstract

The fast development of information technology has made information security and computer networks an essential factor. One possible method of protecting these security resources is the Intrusion Detection System (IDS), which recognizes abnormal packets among incoming data. In this study, we work on its detection capability by exploring a machine learning-based data mining approach. In this approach, proper training data are needed to obtain a useful detection model. Pre-processing is one way to increase the quality of the training data, which can be performed by removing noise. Our research attempts to cluster data for the majority class by using *k*-means that we can recognize the noise by taking an appropriate threshold. In this case, we identify the clusters with a value below the threshold as noise data. Thus, a new majority class of data should not contain noise anymore. This majority class is then combined with the minority class to form a new training data set. It is tested by implementing several classifiers: Naive Bayes (NB), *k*-Nearest Neighbor (*k*-NN), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Random Forest (RF) in the NSL-KDD and UNSW-NB15 dataset. The results we obtained from this proposed method show that it can improve the performance. It is depicted that the best improvement is achieved by using the NB classifier. In NSL-KDD, there is an increase from 88.60% to 88.85%, while in UNSW-NB15, it is from 76.04% to 92.57%.

Keywords: Classification, Computer security, Intrusion detection system, Machine learning, Network security, Undersampling.

1. Introduction

In this digital era, transmitting data between computer networks, such as through the internet, has been shared. This data transmission has made it easy for users to exchange information in any environment. Nevertheless, not all users utilize this technology for functional purposes; some may exploit it to send malicious packets. This activity has been a security issue for decades.

Some methods have been introduced to overcome that security problem; one of them is by detecting the bad incoming packets to the network [1]. It is often carried out by implementing an Intrusion Detection System (IDS), where an alarm is transmitted to the network administrator once a suspicious packet is detected. The IDS is a system for monitoring network traffic that recognizes intruders in the network. Its performance, however, may not be optimal. As a result, a false alarm may be sent just because regular access is detected as an attack, and vice versa.

On the other hand, the development of data mining and machine learning is increasingly widespread. It is useful for data analysis and knowledge-based systems. Based on this, many areas have adopted data mining techniques to solve the problems, including IDS implementation. In addition to the data mining approach, IDS also often takes a misuse-based approach, which works on the principle of matching packets that pass with rules that have been stored in the IDS database. A problem occurs when a pattern of attacks increases, which causes it not found in the database. The data mining implementation in IDS provides a new approach that is different from that of misuse based. The detection is primarily based on an analysis that supports uncertain data on the network [2].

Furthermore, the classification process is often used in this data mining-based IDS [3-5]. However, the classification method's improvement is considered to be less optimum if it is not supported by proper training data. Therefore, valid training data are needed to produce useful IDS. Some enhancements can be done to refine the quality of the training data, for example performing pre-processing data at the beginning. It is the main stage in data mining, aiming to make the algorithms run better and more useful [6]. In the previous studies, feature selection, imbalanced data, and data normalization have been investigated to improve IDS performance [7-9]. Next, the research in instance selector, instance generation, feature selection, imbalanced data, and discretization, has also been done [6, 10].

Imbalanced data occurs when the distribution of some clusters is much larger than the others. We call the classes as majority and minority to describe the class's state in the imbalanced dataset. This type of dataset can be a problem in classification algorithms because a classifier tends to predict the majority instead of minority classes that affect the performance of the classification [11]. Under-sampling and over-sampling techniques are widely implemented to overcome lazy imbalanced data. The over-sampling method is explored to balance data by raising the number of minority data to balance both classes. Another problem in the over-sampling method is over-fitting, mainly when data are generated by multiplying minority data directly [12]. To overcome this issue, an advanced method of over-sampling is developed, including Synthetic Minority Over-sampling Technique (SMOTE) [13], Borderline-SMOTE [14], and Resampling At Random (RAR) [15]. Besides, under-sampling is also implemented to drop the majority class; so that both classes' ratio is balance. However, this method is likely to lose important information from the dataset. Commonly, random under-sampling is designed by randomly reducing the sample of the majority

class. It is employed in the Ensemble of Under-sampling [16], which firstly divides the majority class into several small parts.

In this paper, we work on an imbalanced data technique by combining over-sampling and under-sampling resulted from clustering. In this case, over-sampling is done to raise minority class data in a minority cluster, while the under-sampling technique is to bring down the majority class data in the minority cluster or outlier. Furthermore, we do the clustering first because the previous over-sampling method is possible to be over-fitting. Since the existence of outliers may affect the performance, under-sampling is implemented to remove them. Accessible imbalanced data methods include SMOTE, which can be improved by using clustering, such as the implementation of over-sampling based on the combination of *k*-means and SMOTE [17]. Furthermore, imbalanced class based on clustering is also applied in [18, 19].

This research paper is structured as follows. Section 1 describes the research background, and Section 2 depicts previous research. Section 3 explains the details of the proposed technique, while Section 4 provides the experiment results and the discussion. Finally, in the last section, we draw conclusions and suggestions about developing this method going forward.

2. Imbalanced Data in Machine Learning-based IDS

Rodda et al. [9] used the NSL-KDD data set, the evaluation of imbalance class shows that Random Forest (RF) works well for the Remote to User (R2L) class as a minority and others as the majority. This result also depicts that it is better than Naïve Bayes (NB) and J48. Furthermore, ensemble-based classifiers have been a possible solution [20, 21]. However, for classes that are very imbalanced like Remote to Local (U2R), RF has failed. Slightly different, Altwaijry and Algarny [22] investigated NB's effectivity in minimizing IDS problems by using the KDD data set. It is concluded that NB can detect intrusions well. Further development of this research can be performed by improving the quality of training data that can increase the classification's performance.

The SMOTE and feature reduction are implemented in the NSL-KDD data set with RF classification, whose results are mapped into an evaluation matrix [13, 23]. It is found that the method is capable of increasing the detection rate of R2L and U2R attacks specifically. Their performance rises to 0.3% and 0.6% for R2L and U2R, respectively. In this case, U2R can be detected more accurately than R2L, i.e., correspondingly 96.2% and 96%. This research shows that imbalanced data affect the detection rate.

Research specifically on the development of SMOTE is also carried out by Douzas et al. [17] by combining it with *k*-means clustering. It is assumed that SMOTE can generate a minority sample in the majority region that is becoming noise. The weakness arises in SMOTE that randomly selects minority areas to over-sample whose probability is uniform. So, that research proposes only over-sampling the densely populated minority areas using the k-means approach. The result of this design depicts that the combination of SMOTE and *k*-means method produces better results when tested on 12 imbalanced datasets, which are obtained from the UCI database. The proposed algorithm can outperform SMOTE, random over-sampling, and borderline-SMOTE in handling noise. Besides SMOTE, for handling imbalanced datasets, Mazini et al. [4] use boosting to overcome

imbalanced data on network attack datasets. Here, boosting is a data mining-based meta-algorithm that is implemented to drop variance and imbalances.

Specific ensemble classifier-based sampling approaches have also been investigated. A novel under-sampling technique known as cluster-based instance selection combines instance selection with clustering analysis [19]. The similar data samples are categorized by grouping analysis component taken from the majority class dataset into "subclass". Unrepresentative data samples are then filtered out from this "subclass" by the sample selection component. Using the Knowledge Extraction based on Evolutionary Learning (KEEL) dataset, the experimental results demonstrate that the clustering-based under-sampling technique can make MLP-based ensemble classifiers produce better performance.

Three approaches are often applied to solve the imbalanced data problem: cost-sensitive learning, method adaptation, and data resampling [24]. Among these three methods, resampling data, which is performed either by over-exemplary examples of minority classes or fewer samples in majority classes, is the most commonly implemented. However, in most cases, when this method is applied, there is a trade-off between the complexity and the performance. In this study, the under-sampling technique based on fast clustering is to overcome the imbalance of binary classes. It implies high predictive performance; furthermore, the time complexity relies on the number of instances in the minority groups. In the learning stage, the technique groups minority samples and chooses the same amount of majority ones from every class. Then, specific classifiers are implemented for each cluster. A member of each cluster that is not labelled is moved to the majority class if it is not appropriate to those clusters. Otherwise, a classifier specific to that cluster is taken to put the instance to the correct one. Next, the inverse-distance generated by the cluster is given to the previous results. The measurement includes some methods. By considering computational costs and predictive measures, the Pareto method is used. A broad set of experiments shows that only this proposed method is always found at the border.

3. Clustering Under-sampling Data for IDS

This section explains the details of the proposed technique. The scheme in this research can generally be divided into two parts. The first section discusses the proposed over-sampling and under-sampling methods. The second part is designing the experiment of new data from pre-processed results with imbalanced data into classification and testing. For more details, the research flow can be depicted in Fig. 1.

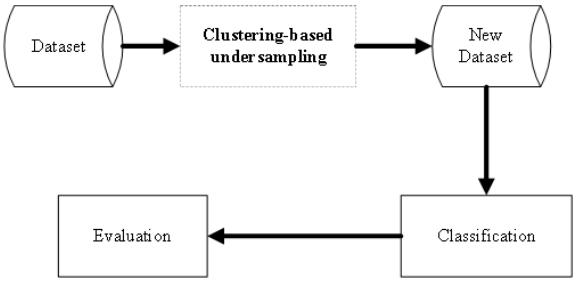

Fig. 1. Research flow.

3.1. Clustering-based under-sampling

The basic idea of this technique is to take out noise from the dataset. The initial step in removing this noise is by firstly clustering the data. In this step, the data are divided into the majority (*Ma*) and minority (*Mi*) classes. The primary of this method is under-sampling, so our focus is on the majority class. Furthermore, the *k*-means analysis is to categorize the majority of data into several clusters. The *k*-means performs clustering by determining the starting point of the cluster randomly. In general, grouping by employing *k*-means can be divided into two stages [1]. The first is to calculate each point according to the Euclidean distance between the considered point and center. The next step is to calculate the new center as a weighted average of points in each class. The algorithm stops when each center does not change. To calculate the distance of each point on the *k*-means, the Euclidean distance in Eq. (1) [2] is employed. Then, to update the center point C_k, the Eq. (2) [3] is applied, where x_i and y_i are two points in a dimensional Euclidean space.

$$d(x,y) = \sqrt{(x_i - y_i)^2 + (x_{i+1} - y_{i+1})^2} \quad (1)$$

$$c_k = \frac{\sum x_i \epsilon c_k \, x_i}{c_k} \quad (2)$$

After forming several clusters, there will be a majority cluster maximum (Ma_{max}) and the majority cluster minimum (Ma_{min}). To determine both values, we use the threshold (*T*) presented in Eq. (3), where the value can be obtained by dividing the total number of entire instances on Ma with the *k* value of that elbow method. In this design, Ma_{max} is a cluster that has some data above the threshold, and Ma_{min} is a cluster that has a value below the threshold.

$$T = \frac{Ma}{k};\ Ma_{max} \geq T,\ Ma_{min} < T \quad (3)$$

After that step, we get a cluster with total instances above and below the threshold. Those with a total instance lower than the threshold are then reduced, so we consider it noise. Otherwise, the clusters are combined with the minority class. Then, we call a new data set that has been selected to eliminate noise. This new data set formed from cluster-based under-sampling is employed for training by the classification engine. For more details, the under-sampling method that we propose can be seen in Fig. 2.

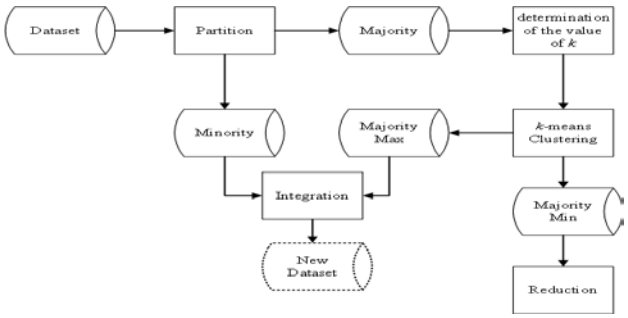

Fig. 2. Clustering based sampling.

3.2. Determination of the value of *k*

Choosing the optimal number of clusters is a fundamental problem in partition-based clusters, as in the *k*-means clustering that requires users to determine in advance the number of *k* that will be produced. Some studies have introduced the method to specify that value. The selection of popular *k* value is by Elbow and Silhouette [25]. This elbow method starts by varying the values of *k*, then calculating the average distance in clusters between objects and the center of clusters for each different *k*. The determination of the optimum *k* value is based on the sharp bend's location.

This proposed method concentrates mainly on the proportion of variance as a function of the total of classes or clusters. According to the idea that an optimal value in the *k*-means algorithm must be found, the value of *k* is added sequentially to see variants of the within-cluster sum of squares (WCSS) by Eq. (4) [26] aimed at finding a minimum WCSS value with a small number of clusters in order to interpret the data.

$$wcss = \sum_{k-1}^{k} \sum_{x_i \in k} \|x_i - c_k\|_2^2 \quad (4)$$

Here, WCSS = 0 means that all data points are in a different cluster, while WCSS = 1, defines that those points are in one cluster.

4. Experiment and Analysis

This section explains the environment of the experiment and its results. It includes the description of the data sets being used for measuring the performance of the proposed method. A comparison with other existing methods is also provided to find a general view of the proposed method.

4.1. Experimental Setup

In this machine learning-based intrusion detection research, the evaluation is done by employing both NSL-KDD and UNSW-NB15 datasets, similar to [27, 28]. The NSL-KDD dataset is an extended version of the previous KDD'99, which was developed in 2009 [29]. In that data set, redundant instances are deleted to prevent the deviated classification results [30]. This data collection comprises several versions, of which 20% of the training data are KDDTrain + 20%, which contains 25192 examples. The test dataset is KDDTest +, which consists of 22544 instances. This dataset has 41 attributes and one target class. The 41 features can be grouped into four: Basic, Content, Traffic, and Host [22].

The UNSW-NB15 dataset is provided by The Australian Center for Cyber Security Lab, which employs the IXIA Perfect Storm tool for developing datasets comprising normal and anomaly traffics [31]. This data set has 48 features and one target class. Attack classes are again grouped into Backdoors, Analysis, Exploits, Fuzzers, Generic, Shellcode, Worms, Reconnaissance, and DoS. The data consist of 175,340 records of training and 82,000 records of testing data. These data, which were published by Moustafa and Slay [32] are an alternative for IDS research.

Naïve Bayes (NB), *k*-NN, SVM, MLP, and Random Forest (RF) algorithms are implemented for the classification, whose results are then put in the confusion matrix. Based on this table, some metrics are explored to measure: accuracy,

precision, and recall. To calculate accuracy, we use the number of anomaly measurements correctly detected by the total number of anomaly measurements [33]. Precision is the comparison between true positive predictions and the total positive predicted results. Recall is the number of true positives compared to the overall true data.

4.2. Experiment Results and Discussion

The first step in the experiment is to select the majority class from the NSL-KDD dataset, which can be found in Table 1 whose experimental result is in Table 2. Table 1 is about the composition of the NSL-KDD training data. So, we can get normal classes as the majority. The next is to determine the value of k for this specified majority class and get an elbow graph, as provided in Fig. 3. According to this figure, we can specify the value of k = 4; so that, in the next clustering process, we have 4 clusters. The next step is to determine T's value, from Table 1, Ma = 13449 (Normal class). So, T = 3362.25 depicts that the majority of max clusters have 13394 records based on Eq. (3). It can be seen in Table 2 that only one cluster which has a value above the threshold so that we can get Ma_{max} = 13394. The new training data comprise 25137 records in a total of Ma_{max} and minority class (Mi) The experiment is conducted by the classification methods to see the performance of the proposed pre-processing step against classification.

Table 1. Representation of NSL-KDD.

Class	Count	Status
Anomaly	11743	Minority
Normal	13449	Majority

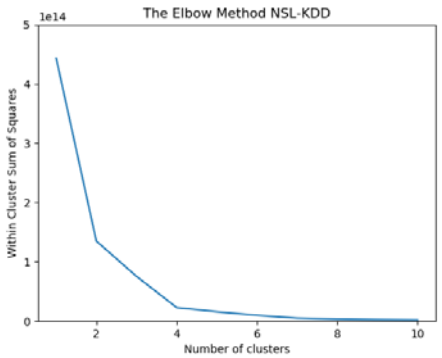

Fig. 3. Elbow Result of NSL-KDD.

Table 2. The results of NSL-KDD clustering

	Cluster-1	Cluster-2	Cluster-3	Cluster-4
Total Record	13394	50	3	2
T = 3362.25	$> T$	$< T$	$< T$	$< T$

As previously described, the experiments use three different classifiers to evaluate the performance of both with and without the proposed method. Results from experiments on the NSL-KDD data set can be seen in Table 3. From this table,

we find that using the proposed method improves the performance of those all classifiers. The highest increase in accuracy occurs in the NB classification from 88.6% to 88.9%, with precision and recall of 88.9% and 88.8%. It proves that removing noise data can increase the level of accuracy. This experiment also shows that by reducing 0.02% of data noise, the accuracy rises around 0.1% - 0.3% in the NSL-KDD dataset.

Table 3. Classification result of NSL-KDD.

Classifier	Proposed method (with pre-processing)			Existing method (without pre-processing)		
	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)
SVM	95.4529	95.5	95.5	95.4192	95.4	95.4
Naïve Bayes	88.8491	88.9	88.8	88.6075	88.6	88.6
***k*-NN**	99.3993	99.4	99.4	99.3728	99.4	99.4
MLP	99.1805	99.2	99.2	99.0791	99.1	99.1
Random forest	99.7573	99.8	99.8	99.7568	99.8	99.8

In the next experiment, we compare the proposed method with other pre-processing algorithms. For this purpose, we apply HyperSMURF [34], SMOTE [13], Random under-sampling [35], and AdaBoost [36] whose results are provided in Table 4. To compare with those existing pre-processing methods, we use the RF classifier because, based on the experimental results depicted in Table 3, RF has the best performance. Specifically, by using the proposed pre-processing, its accuracy reaches 99.7573% with both precision and recall is 99.8%.

Table 4. Comparison between methods using the Random Forest classifier in the NSL-KDD dataset.

Method	Accuracy (%)	Precision (%)	Recall (%)
HyperSMURF [34]	99.4562	99.5	99.5
SMOTE [13]	99.8051	99.8	99.8
Random under-sampling [35]	99.7565	99.8	99.8
AdaBoost [36]	99.7301	99.7	99.7
Proposed method	99.7573	99.8	99.8

From Table 4, we see that the proposed method has a better position than HyperSMURF, Random under-sampling, and AdaBoost methods. However, it is slightly under SMOTE. It is because, in the NSL-KDD dataset, the distribution of the corresponding data is good enough, so the proposed method finds only a little noise. This is different from SMOTE, which can generate over-sampling data on better clusters; so that, SMOTE can cover imbalances that occur in NSL-KDD. Next, the resulting training data are more appropriate for forming a classification model in the RF. Besides, the proposed method only removes less noise in NSL-KDD. This condition significantly impacts the use of classifiers in Table 3 and also some existing methods. Generally, the proposed method outperforms the existing ones.

To further analyse the proposed method, we apply it to a different data set: UNSW-NB15. As in the previous experiment, we first looked at the data set's condition to determine the majority and minority classes that we summarize in Table 5. It is shown that more attack data dominate to be the majority of data (119340 records). After we get them, the value of k is determined using the elbow method, which results in $k = 2$, as in Fig. 4. From Eq. (3), the threshold, $T = 59670$.

Table 5. Representation of UNSW-NB15

Class	Count	Status
Attack	119340	Majority
Normal	56000	Minority

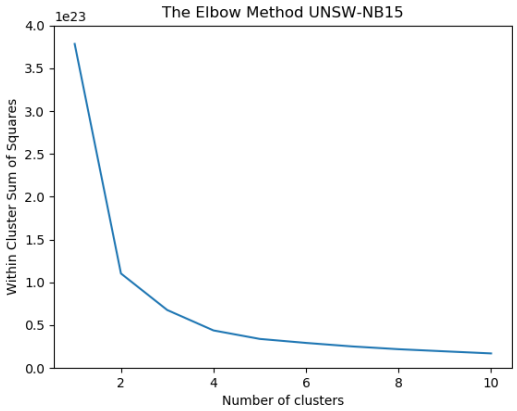

Fig. 4. Elbow result of UNSW-NB15.

The result of cluster generation with UNSW-NB15 is depicted in Table 6. It is found that the number of records in cluster-1 is less than the threshold. We can reduce 34201 records data on Cluster-1 (Ma_{min}) so that the total of majority data become 85140 records (Ma_{max}). After the noise is minimized, the new data set is combined with the minority (Mi) and Ma_{max} to have new training data with 141150 records. As in the previous experiments, the classification is performed by using: SVM, Naive Bayes, k-NN, MLP, and Random Forest, whose results are depicted in Table 7. We observe that the proposed method can refine the performance of classification on UNSW-NB15. It is shown that significant improvement can be achieved in NB, where the accuracy increases from 76.0410% to 92.5755%; their corresponding precision and recall are from 83.4% and 76.0% to 92.8% and 92.8%, respectively. Overall, the use of the proposed technique has raised the performance in all evaluated metrics.

Next, similar to that of NSL-KDD, the proposed method is compared with existing ones: HyperSMURF, SMOTE, Random Under-sampling, and AdaBoost by implementing them in UNSW-NB15. Additionally, the Random Forest is also taken for the classification. Table 8, which represents this comparison result, shows that the proposed method has the best performance, where it achieves 97.1% for all evaluated metrics. Furthermore, the proposed method can recognize around 0.3% of data in the majority class as noise.

Table 6. The results of UNSW-NB15 clustering.

	Cluster-1	Cluster-2
Record	34201	85140
T **= 59670**	$< T$	$> T$

Table 7. Classification result of UNSW-NB15.

Classifier	Proposed method (with pre-processing)			Existing method (without pre-processing)		
	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)
SVM	96.9860	97.0	97.0	93.3649	93.4	93.3
Naïve Bayes	92.5755	92.8	92.8	76.0410	83.4	76.0
***k*-NN**	98.9634	99.0	99.0	97.0184	97.0	97.0
MLP	96.5750	96.6	96.6	93.4739	94.0	93.5
Random Forest	97.1482	97.1	97.1	88.9969	89.0	89.0

Table 8. Comparison between methods using the Random Forest classifier in the UNSW-NB15 dataset.

Method	Accuracy (%)	Precision (%)	Recall (%)
HyperSMURF [34]	95.8595	96.3	95.9
SMOTE [13]	90.5438	90.6	90.5
Random under-sampling [35]	95.5756	95.6	95.6
AdaBoost [36]	85.1723	85.7	85.2
Proposed method	97.1482	97.1	97.1

As of accuracy, the experimental results can be shown in Fig. 5. It can be inferred that the proposed technique is better than others in almost all evaluations. It is only slightly less than that of SMOTE when the NSL-KDD dataset is used. In the case the experiment is carried out in UNSW-NB15, the proposed method with pre-processing is significantly better than the others, including that without pre-processing. On the other hand, accuracy is sometimes not directly proportional to the other parameters: true positive and true negative. In addition to seeing the level of accuracy, this experiment also calculates the level of precision and recall in each experiment (Tables 3, 4, 7, 8). To make it easier to see the precision and recall level, we present the F1-Score in Fig. 6. It is shown that it has the same pattern as the accuracy, which means that the ability to detect true positives and true negatives is relatively good. As previously described, the quality of NSL-KDD data has been improved from KDD'99 [29]. Thus, without implementing pre-processing to this data, a relatively good accuracy value can be achieved. Nevertheless, its value can still be improved by applying the proposed method. Therefore, it can be inferred that this research, in general, can be implemented to obtain better performance.

The proposed method has been tested on two binary-class datasets for IDS. It is worth noting that the proposed method is not designed for the multiclass dataset. So, further development needs to do before it is evaluated in a multiclass

environment. Moreover, various multiclass datasets that are appropriate for IDS should be applied.

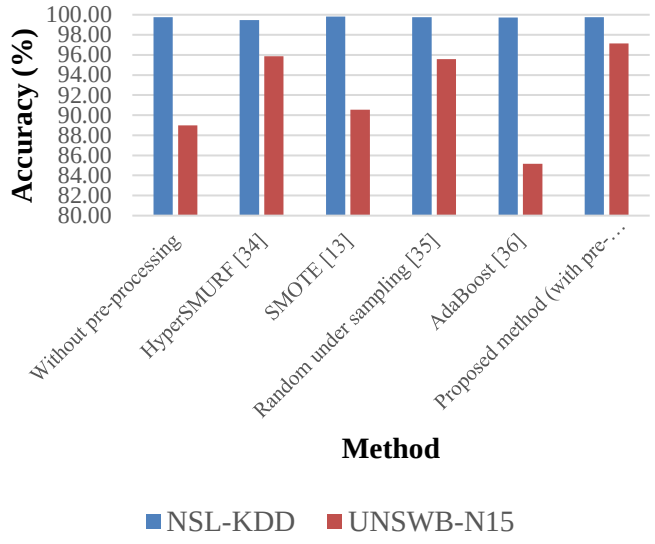

Fig. 5. Accuracy comparison.

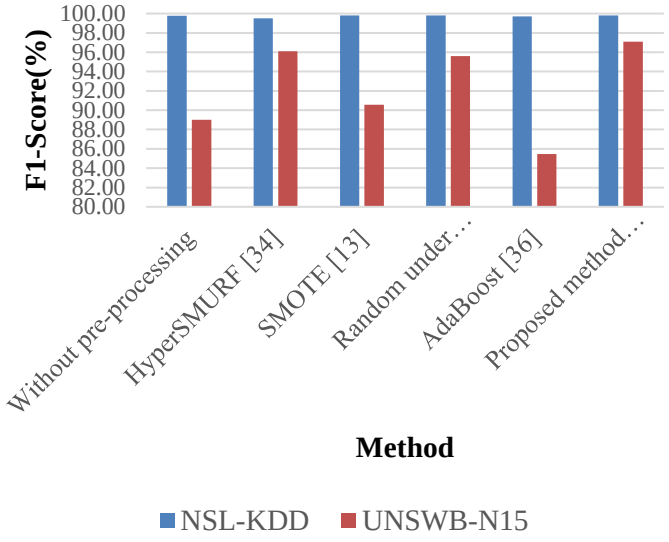

Fig. 6. F1-Score comparison.

5. Conclusions

In this research, we have proposed a pre-processing method for imbalanced data that can be applied to machine learning-based IDS. It is done by under-sampling the

majority class to remove noise. For this purpose, we use the *k*-means clustering approach. The clustered data are filtered by comparing them with our proposed threshold values. That is, data below the threshold are removed, and the remaining are combined with the minority to obtain new training data.

This proposed method, which has been evaluated by implementing it in both NSL-KDD and UNSW-NB15 datasets, is superior. In NSL-KDD, it outperforms most other evaluated pre-processing methods: HyperSMURF, Random under-sampling, AdaBoost; while in UNSW-NB15 it is the best. Furthermore, when UNSW-NB15 data set is used, the improvement is significant. Concerning the accuracy, the highest level is achieved in about 98%; it is much higher than the others. This superiority is also followed by other measurement metrics: precision and recall, where its values are also the highest. Therefore, the proposed method is more appropriate to use in imbalanced data.

In the future, removing the noise from the majority class is still the concern. It needs to find a more effective algorithm for this reduction. One of the possible ways is finding the more appropriate threshold value, which determines the status of the corresponding data.

References

1. Agarkar, A.A.; and Agrawal, H. (2019). LRSPPP: Lightweight R-LWE-based secure and privacy-preserving scheme for prosumer side network in smart grid. *Heliyon*, 5(3), 1-31.
2. Aamir, M.; and Zaidi, S.M.A. (2019). Clustering based semi-supervised machine learning for DDoS attack classification. *Journal of King Saud University - Computer and Information Sciences*, in press.
3. Hajisalem, V.; and Babaie, S. (2018). A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. *Computer Networks*, 136(1), 37-50.
4. Mazini, M.; Shirazi, B.; and Mahdavi, I. (2018). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *Journal of King Saud University - Computer and Information Sciences*, 31(4), 541-553.
5. Singaravelan, S.; Arun, R.; Arunshunmugam, D.; Joy, S.J.C.; and Murugan, D. (2020). Inner interruption discovery and defense system by using data mining. *Journal of King Saud University - Computer and Information Sciences*, 35(2), 592-598.
6. García, S.; Luengo, J.; and Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems*, 98, 1-29.
7. 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.
8. Aziz, M.N.; and Ahmad, T. (2019). Cluster analysis-based approach features selection on machine learning for detecting intrusion. *International Journal of Intelligent Engineering and Systems*, 12(4), 233-243.
9. Rodda, S.; and Erothi, U.S.R. (2016). Class imbalance problem in the network intrusion detection systems. *Proceedings of the First International Conference*

on Electrical, Electronics, and Optimization Techniques. Chennai, India, 2685-2688.

10. Ramírez-Gallego, S.; Krawczyk, B.; García, S.; Woźniak, M.; and Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, 239, 39-57.
11. Sain, H.; and Purnami, S.W. (2015). Combine sampling support vector machine for imbalanced data classification. *Procedia Computer Science*, 72, 59-66.
12. Lu, W.; Li, Z.; and Chu, J. (2017). Adaptive ensemble undersampling-boost: A novel learning framework for imbalanced data. *Journal of Systems and Software*, 132, 272-282.
13. 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, 321-357.
14. Han, H.; Wang, W.Y.; and Mao, B.H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Proceedings of the First International Conference on Intelligent Computing*. Hefei, China, 878-887.
15. Kim, Y.G.; Kwon, Y.; and Paik, M.C. (2019). Valid oversampling schemes to handle imbalance. *Pattern Recognition Letters*, 1251, 661-667.
16. Kang, P.; and Cho, S. (2006). EUS SVMs: Ensemble of under-sampled SVMs for data imbalance problems. *Proceedings of Neural Information Processing, 13th International Conference*. Hongkong, China, 837–846.
17. Douzas, G.; Bacao, F.; and Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, 465, 1-20.
18. Tsai, C.F.; Lin, W.C.; Hu, Y.H.; and Yao, G.T. (2019). Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences*, 477, 47-54.
19. Lin, W.C.; Tsai, C.F.; Hu, Y.H.; and Jhang, J.S. (2017). Clustering-based undersampling in class-imbalanced data. *Information Sciences*, 409-410, 17-26.
20. Nguyen, T.T.; Luong, A.V.; Dang, M.T.; Liew, A.W.C.; and McCall, J. (2020). Ensemble selection based on classifier prediction confidence. *Pattern Recognition*, 100, 1-15.
21. Salunkhe, U.R.; and Mali, S.N. (2016). Classifier ensemble design for imbalanced data classification: A hybrid approach. *Procedia Computer Science*, 85, 725-732.
22. Altwaijry, H.; and Algarny, S. (2012). Bayesian based intrusion detection system. *Journal of King Saud University - Computer and Information Sciences*, 24(1), 1-6.
23. Tesfahun, A.; and Bhaskari, D.L. (2013). Intrusion detection using random forests classifier with SMOTE and feature reduction. *Proceedings of the First International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*. Pune, India, 127-132
24. Ofek, N.; Rokach, L.; Stern, R.; and Shabtai, A. (2017). Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem. *Neurocomputing*, 243, 88-102.

25. Kuwil, F.H.; Shaar, F.; Topcu, A.E.; and Murtagh, F. (2019). A new data clustering algorithm based on critical distance methodology. *Expert Systems with Applications*, 129, 296-310.
26. Marutho, D.; Handaka, S.H.; Wijaya, E.; and Muljono. (2018). The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. *Proceedings of the Third International Seminar on Application for Technology of Information and Communication.* Semarang, Indonesia, 533-538.
27. 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 of the Third IEEE International Conference on Computing, Communication and Security*. Kathmandu, Nepal, 558-561.
28. Yang, Y.; Zheng, K.; Wu, C.; and Yang, Y. (2019). Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. *Sensors*, 19(11), 1-20.
29. Tavallaee, M.; Bagheri, E.; Lu, W.; and Ghorbani, A.A. (2009). A detailed analysis of the KDD CUP 99 data set. *Proceedings of the First IEEE Symposium on Computational Intelligence for Security and Defense Applications*. Ottawa,Canada, 1-6.
30. Aggarwal, P.; and Sharma, S. K. (2015). Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection. *Procedia Computer Science*, 57, 842-851.
31. Belouch, M.; Hadaj, S.E.; and Idhammad, M. (2018). Performance evaluation of intrusion detection based on machine learning using Apache Spark. *Procedia Computer Science*, 127, 1-6.
32. Moustafa, N.; and Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *Proceedings of the Fifth Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 1-6.
33. Soliman, H.H.; Hikal, N.A.; and Sakr, N.A.; (2012). A comparative performance evaluation of intrusion detection techniques for hierarchical wireless sensor networks. *Egyptian Informatics Journal*, 13(3), 225-238.
34. Schubach, M.; Re, M.; Robinson, P.N.; and Valentini, G. (2017). Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants. *Scientific Reports*, 7, 1-12.
35. Zhang, Y. (2016). An improved random sampling approach for large data set mining. *Proceedings of the First International Conference on Smart City and Systems Engineering (ICSCSE)*. Hunan, China, 558-561.
36. Zhang, Z.; and Xie, X. (2010). Research on AdaBoost.M1 with random forest. *Proceedings of the Second International Conference on Computer Engineering and Technology*. Chengdu, China, 647-652.