Received: March 18, 2019

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Increasing Accuracy and Completeness of Intrusion Detection Model Using Fusion of Normalization, Feature Selection Method and Support Vector Machine

Bambang Setiawan[1]* **Supeno Djanali[1]** **Tohari Ahmad[1]**

[1] *Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
* Corresponding author's Email: setiawan@is.its.ac.id

Abstract: Detecting intrusion in network traffic has remained a problem for years. Development in the field of machine learning provides an opportunity for researchers to detect network intrusion without using a database signature. Accuracy and completeness are two critical aspects in determining the performance of an intrusion detection system. The amount of unbalanced training data on each type of attack causes the system to have high accuracy, but it is difficult to detect all kinds of attacks. So, it does not meet the completeness aspect. In this paper, we propose an intrusion detection model using a combination of the modified rank-based information gain feature selection method, log normalization, and Support Vector Machine with parameter optimization. Overall accuracy achieved using 17 features from NSLKDD dataset is 99.8%, while the false alarm rate is 0.2%. The completeness aspect can be achieved, and the detection accuracy of the minority class can be increased.

Keywords: Feature selection, Intrusion detection, Network security, Normalization, Support vector machine.

1. Introduction

Intrusion detection systems (IDS) dynamically monitor system activity in a certain environment and decide whether an activity will be considered an attack. Based on the detection method, the IDS are classified into two categories, named as a misuse based and an anomaly based [1]. The misuse based IDS uses the stored signatures of known attacks to identify traces of malicious behavior. An anomaly-based IDS is an approach to detect intrusions by first defining the training data of normal activities, and if the behavior of traffics is deviating from the training data, then the system will mark as malicious [2].

Accuracy and Completeness are two essential aspects measured in assessing the efficiency of an IDS [1, 3]. Inaccuracy happens when an IDS identifies an intrusion as a legitimate activity in the environment, and incompleteness occurs when the IDS fails to detect one or more types of attacks.

Class imbalance is one of the main problems for IDSs that use data mining and machine learning methods. Because classifiers generally designed to minimize the global error rates and have not considered the condition of class imbalance; this causes classifiers to perform poorly on unbalanced datasets [4, 5]. The amount of training data for each type of unbalanced attack can cause IDS to have high accuracy but the difficulty in identifying all types of attacks, so that completeness aspect is not fulfilled. IDS dataset likes KDD Cup 1999 and NSL-KDD, has a very unbalanced number of attack instances [6, 7].

Support Vector Machine (SVM) is one of the most popular tools used in classification. It has some advantages such as the absence of local minima, high generalization capability, being able to adapt to a small number of sample data and high dimensional sample data [8]. Liu et al. [9] mention that SVMs can work well in small and moderate imbalance ratio dataset.

Performance of SVM can also be improved by integrating it with dimensional reduction and optimization of parameter techniques. Thaseen and Kumar [10] show that SVM with dimensional reduction and parameter optimization can increase the classification rate and reduction detection time.

The use of normalization methods is very influential in the feature selection process and SVM implementation on IDS. Normalization can shorten the learning phase and improve the performance of classifier [11]. Although the normalization process is useful, it turns out that this process also has a negative impact if it is not used correctly. The normalization process has the potential to change the correlation value and mutual information between normalized features and the classification label. Mutual information changes will greatly affect the results of feature selection and have an impact on the process of finding support vectors in SVM. To best our knowledge, no one has explained the potential loss of information on the normalization process and its impact on IDS performance. Information loss can occur when we have to round off the normalization result to a particular decimal place. This condition can potentially happen in features with large value ranges because of the digit number of the back of the normalization results that have discriminative power. If that happens, the strength of the discrimination features will decrease. This condition can affect the results of feature selection and IDS's performance.

Based on this information, we propose an IDS model that combines normalization, feature reduction, and SVM with parameter optimization to get relatively high accuracy and to achieve the completeness aspect.

There are three uniquenesses of the proposed model compared to the existing IDS approach. First, this model uses log normalization methods to avoid changing mutual information from features. Second, the use of feature selection method to get a feature subset that better support detection of a minority class. Third, this model integrates normalization, feature selection method, and the self-adjusting capabilities of SVM in handling unbalance classes. These approaches are intended to achieve completeness aspect, improve detection accuracy in minority class and keep capable of detecting another with high accuracy.

The rest of this paper is organized as follows. Section 2 describes the related work. In Section 3 we discuss feature selection, attribute evaluation, and normalization methods. In the next section, we present the multi-class SVM model. Then in Section 5, we describe the methodology of the proposed feature selection. Implementation results are explained and discussed in Section 6. Finally, the conclusion is given in Section 7.

2. Related work

Many IDS models have been developed to overcome the limitations of the anomaly detection model. Following this, we will analyze some IDS literature in the last four years using the KDD Cup 99 and the NSL-KDD dataset which is a new version of KDD Cup 99.

Ashfaq et al. [12] use a fuzzy based semi-supervised learning approach and a single hidden layer feed-forward neural network (NN) to improve the performance of IDS. They use a fuzzy quantification process to categorize the unlabelled samples in the first stage. Then NN is re-trained by incorporating each category separately into the original training set. By using NSL-KDD dataset, the approach they use can achieve overall class classification accuracy above 80% while the classification accuracy for each class is not presented.

Jamali and Jafarzadeh [13] use a hierarchically structured learning automata in their IDS model. The learning automata is used to choose the optimal action. They mention that their approach is a flexible model that excels in detecting unknown attacks. The classification accuracy of this approach for the overall class is over 90%. Similar to [12], this study also does not present the classification accuracy for each class.

Bostani and Sheikhan [14] use a modified optimum path forest (OPF) algorithm for detecting intrusions. They improve the quality of training datasets by partitioning training data into homogeneous training subsets using the k-means clustering algorithm. This approach can increase the IDs performance in term scalability, execution time, detection rate, and false alarm. This model can detect all intrusion classes tested with classification accuracy for all classes exceeding 90%. Accuracy for minority classes (R2L and U2R) exceeds 77%, accuracy for Normal class and DoS class exceeds 90%. But the accuracy of Probe class which is also as majority class, together with Normal and DoS, is below 90%.

Pajouh et al. [15] propose an IDS model based on two-tier classifier using k-Nearest Neighbor (k-NN) and Naïve Bayes classifier**.** They use Linear Discriminant Analysis for dimensionality reduction. This approach can also detect all classes of attacks. Its classification accuracy of the overall class is higher than [14]. But the accuracy for the Probe, R2L, and U2R classes is lower.

A multi-level hybrid IDS using SVM and extreme learning machine is proposed by Al-Yaseen et al. [16]. They use a modified k-mean to improve the quality of the training dataset. This approach can also detect all classes of attacks. Its classification accuracy of the overall class is higher than [15]. But the accuracy for the R2L and U2R classes is lower.

Thaseen and Kumar [10] propose a multi-class SVM to recognize the diverse attacks on a network. They use the z-score method in the normalization stage and use the chi-square feature selection method to choose appropriate attributes from the dataset. The experimental results show that their approach can detect all classes of attack with higher accuracy than those produced by the model in [16]. They use 31 features or greater than 50% of the total features of the NSL-KDD dataset.

A hybrid approach integrating NN with the evolutionary algorithm for detecting intrusions is proposed by Dash [17]. He uses particle swarm optimization (PSO) and gravitational search to training artificial NN. The performance of this hybrid approach can outperform the performance of the model in [10], in terms of classification accuracy of the overall class. Unfortunately, his study does not present the accuracy produced for each class.

Mahendiran and Appusamy [18] propose the other approach using CRF based classifier along with a feature selection method using One-R algorithm for detection intrusion. On the experiment, this approach can detect diverse attacks with high accuracy. The performance of this approach can outperform the performance of model in [10], in terms of classification accuracy of the overall class and U2R class but for other classes it is lower.

Kumar et al. [19] also propose a multi-class SVM to detect intrusion. They use a multi-linear dimensionality reduction (MLDR) method to reduce the dimensionality of the dataset. Their experimental results show that this approach also can improve performance SVM in classification accuracy. The performance of this approach can outperform the performance of model in [18], in terms of classification accuracy of the overall class but for each classes it is lower.

Lin et al. [20] propose another approach using centroid-based classifier, namely the cluster center and nearest neighbor (CANN). This approach generates the one-dimensional representative feature from the sum of two distances. The first distance is a distance between the data point to all centroids, while the second is a distance between the data point to its nearest neighbor in the same cluster. A k-NN classifier is used to process the one-dimensional representative feature. This approach can also detect all classes of attacks. The classification accuracy of this approach for overall class is over 99% but accuracy of the U2R class is under 5%.

In this study, we propose an intrusion detection model that can increase minority class detection and maintain classification accuracy in the overall class and the majority class remains above 90%. We integrate multi-class SVM with log normalization and the feature selection method to improve the accuracy of detecting the minority class. Optimization of the kernel parameter is done using the grid search techniques.

3. Attribute evaluation, normalization, and feature selection

In this section, we describe the attribute evaluation measure, the normalization methodology, and feature selection that we are adopting.

3.1 Attribute evaluation measure

The measurement based on information content widely used in machine learning. The amount of information from the outcome X_j is defined as a negative logarithmic of its probability as follows:

$$I(X_j) = -log_2 P(X_j) \qquad (1)$$

The average amount of information is called entropy of an outcome. If our experiment have m disjoint possible outcomes X_j where $j = 1..m$ and $\sum_j P(X_j) = 1$, the entropy of outcome is defined as:

$$H(X) = -\sum_j^m P(X_j)\, log_2 P(X_j) \qquad (2)$$

Information-gain is specified as the amount of information resulted from the attributes to determine the class as shown in Eq. (3). It is also known as mutual information due to its symmetry, as shown in Eq. (4).

$$InfoGain(A) = H_C - H_{C|A} \qquad (3)$$

$$\begin{aligned} H_C - H_{C|A} &= H_C + H_A - H_{CA} = I(A;C) \\ &= H_A - H_{A|C} = i(C;A) \qquad (4) \end{aligned}$$

Information-gain is a standard measurement of the quality of the attributes. In this research, we adopt information-gain as the attribute evaluation measurement to evaluate normalization methods which is applied to our proposed feature selection.

Table 1. The difference of information gain data in NSLKDD dataset before normalization and after rounding off the results of normalization with 2 to 10 of decimal places

Normalization scheme	Total Information Gain differences								
	2	3	4	5	6	7	8	9	10 *)
Log	0.00108	0.00004	-	-	-	-	-	-	-
Min-Max	0.05298	0.04767	0.03751	0.02953	0.01851	0.00753	0.00249	-	-
Z-Score	0.04136	0.03137	0.01521	0.00692	0.00227	-	-	-	-

*) number of the decimal place

3.2 Normalization and the problem of rounding off the normalization results

In this section, we describe the normalization methodology that we are adopting. We evaluate three schemes of attribute normalization for intrusion detection classifier, namely: z-score, min-max, and log normalization.

Min-max is the most straightforward normalization method that produces the standard numerical range of the scores from 0 to 1. The normalized score is determined in Eq. (5) where the normalized score of x is indicated by x', max(X) is the maximum value, and min(X) is the minimum value of the raw matching scores.

$$x' = (x - min(X))/(max(X) - min(X)) \quad (5)$$

Z-score normalization is also referred to as statistical normalization. The normalized scores s' are calculated by using Eq. (6) where σ is the standard deviation and μ is the mean of the set of scores.

$$s' = (s - \mu)/\sigma \quad (6)$$

Next, the Log normalization scores x' are calculated as follows:

$$x' = log(1 + x) \quad (7)$$

where x is the value of the feature before normalization and x' is its value after normalization.

There is a potential problem in rounding off the normalization result, which is risky losing information or changes to the original information of the dataset features. To examine the extent of these risks, we observe the information gain of continuous features in the NSL-KDD dataset before the normalization process and after the normalization round.

We calculate the difference in the amount of total information gain before normalization and after rounding off the results of normalization. The result of normalization is rounded by using nine variations of decimal numbers whose value is from 2 to 10. The summary of this observation is presented in Table 1.

Observation results show that the highest risk of rounding the normalization results is on the min-max method, followed by z-score and log normalization. The rounding of min-max normalization results using small decimal places (under nine) can a change in the information gain value. Furthermore, the log normalization scheme is better than the others tested normalization scheme because it has the three decimal place-safe threshold. Therefore, we propose to use the log normalization method in this study.

3.3 Feature selection

We use the filter based feature selection approach to find the best subset of features from the original dataset. This selection using a ranking strategy; each feature is sorted independently based on the score function in descending order. The feature selection process takes places in three stages. In the first stage, it transforms nominal features to numeric then perform the normalization process. In the second stage, it builds a feature ranking based on the value of information-gain features. Finally, in the third stage, it chooses the candidate from the n-best rank feature subsets that are then passed to a classification algorithm.

4. Multi-class support vector machine classification model

The SVM is a supervised learning algorithm introduced by Boser, Guyon, and Vapnik [21]. The SVM classifies the data points by identifying a hyperplane. Consider N training data $\{x_1, y_1\}, \dots, \{x_N, y_N\}$, where $x_i \in R^m$ is a m-dimensional feature vector representing the i^{th} training data, and $y_i \in \{-1,1\}$ is the class label of x_i. A hyperplane in the feature space can be described as

$$g(x) = w^T x + b \quad (8)$$

where $w \in R^m$ and b is scalar.

When applying linear SVM does not produce satisfactory performance, it is recommended to use nonlinear SVM. The basic idea is using nonlinearly mapping $\emptyset(x)$ for mapping x to a much higher dimensional space in which the optimal hyperplane is found. The nonlinear mapping is done using kernel function $K(x_i, x_j)$ which computes the inner product of vectors $\emptyset(x_i)$ and $\emptyset(x_j)$. The commonly used kernel functions is the polynomial function

$$K(x_i, x_j) = (x_i^T x_j + 1)^d \tag{9}$$

and the radial basis function (RBF)

$$K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right). \tag{10}$$

In this research, we use the RBF kernel as a kernel function.

At the classification stage, the class label y_{SVM} of a sample x is determined by the sign of the following decision function

$$\begin{aligned} f(x) &= w^T \emptyset(x) + b \\ &= \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \end{aligned} \tag{11}$$

where α_i are the Lagrange multiplier coefficient for the i^{th} sample.

SVM was originally designed only for the classification of two classes, then developed for multi-class classification. There are two methods to implement SVM in multi-class, using the one-against-all (OAA) method or the one-against-one (OAO) method. The study conducted by Hsu and Lin [22] shows that OAO method has advantages in practical use. In this method, a multi-class SVM model with k classes dataset will be constructed from $k(k-1)/2$ SVM. Each one is trained on data from two classes.

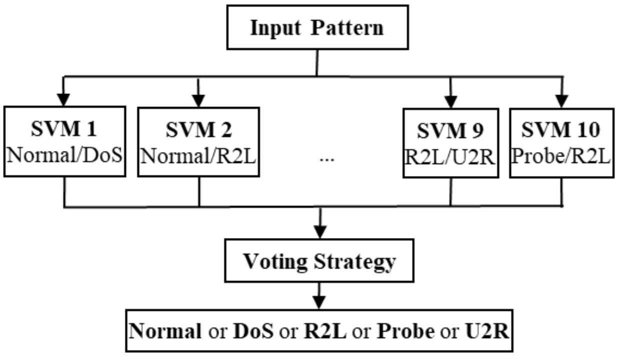

Figure. 1 Structure of multi-class SVM using OAO approach

In our research problem that differentiated five types of network traffics, we construct ten SVMs that structural diagram is as in Fig. 1. The five types of network traffic are Normal, Probe, Denial of Service (DoS), User to Root (U2R), and Remote to Local (R2L).

SVM-1 for class Normal and class DoS, SVM-2 for class Normal and class R2L, SVM-3 for class Normal and class Probe, and SVM-4 for class Normal and class U2R. SVM-5 for class DoS and class R2L, SVM-6 for class DoS and class Probe, and SVM-7 for class DoS and class U2R. SVM-8 for class R2L and class Probe, SVM-9 for class R2L and class Probe, and SVM-10 for class Probe and class U2R.

The SVM-1 processes training data composed of class Normal and class DoS, and it also classifies only class Normal and class DoS in testing data. When carrying out the classification process, all SVMs classify the test data and the results are identified as the class with the highest number of votes.

5. Proposed work

In this research, we propose a hybrid model for intrusion detection, which is the integration of multi-class SVM with an optimization parameter, log normalization, and modified rank based information-gain feature selection (modified-RIGFS) method. Block diagram of this proposed model is presented in Fig. 2. It has four stages of the process. In the initial stage, data preprocessing is done by transforming the nominal feature to numeric then performing the log normalization process on all features.

Feature selection is done using the modified-RIGFS method in the second stage. We modify the original rank based information-gain feature selection (original-RIGFS) method to get the feature subset which supports in detecting the minority classes. Feature selection is done on a temporary dataset that only consists of the Normal class and 50% of attack classes which considered as minority classes, namely the R2L class and U2R class. The best subset feature, 17 top ranking features, is chosen. Next, the subset of features is used to generate new datasets from the complete dataset consisting of all classes.

In the third stage, optimization kernel parameter gamma and C is done by using a grid search method. The parameter pair that produces the best accuracy is taken as the optimal parameter. In the fourth stage, the SVM classifier uses the optimal parameter to

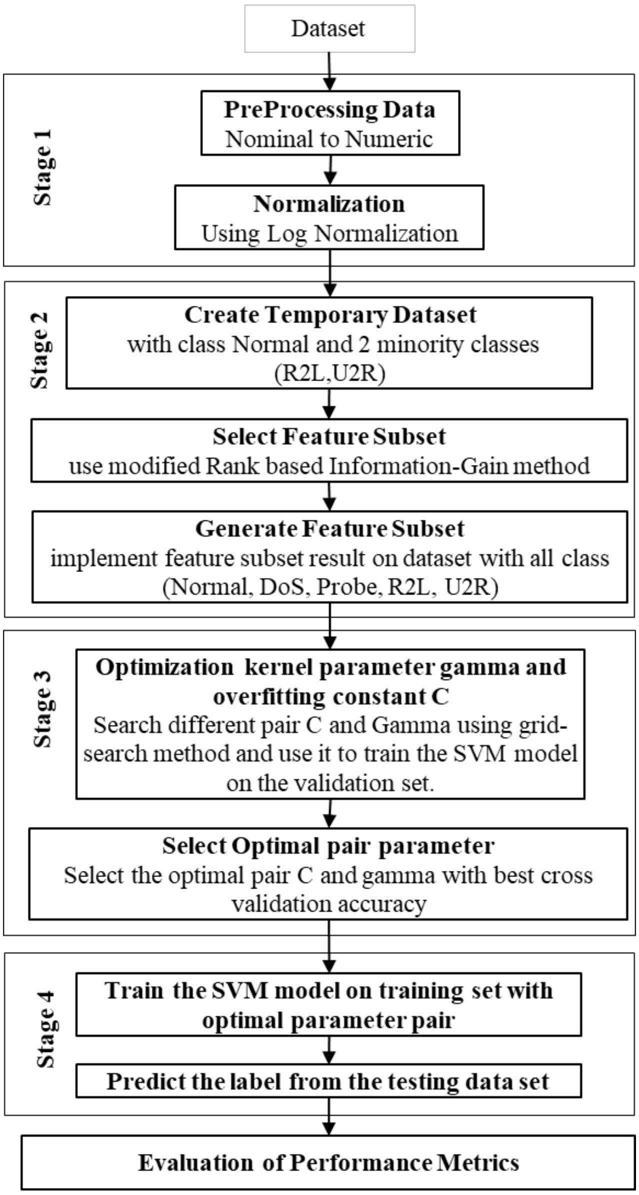
Dataset
Stage 1
PreProcessing Data
Nominal to Numeric
Normalization
Using Log Normalization
Stage 2
Create Temporary Dataset
with class Normal and 2 minority classes (R2L,U2R)
Select Feature Subset
use modified Rank based Information-Gain method
Generate Feature Subset
implement feature subset result on dataset with all class (Normal, DoS, Probe, R2L, U2R)
Stage 3
Optimization kernel parameter gamma and overfitting constant C
Search different pair C and Gamma using grid-search method and use it to train the SVM model on the validation set.
Select Optimal pair parameter
Select the optimal pair C and gamma with best cross validation accuracy
Stage 4
Train the SVM model on training set with optimal parameter pair
Predict the label from the testing data set
Evaluation of Performance Metrics

Figure.2 Proposed intrusion detection model

train and evaluate the model by using the 10-fold cross-validation method.

6. Implementation and results

We conduct the experiments by using Java programming and Weka 3.8.3 library [23]. OAO multi-class SVM with RBF kernel is implemented using the LibSVM [24] package that integrated with Weka. In addition, we use the grid-search method for optimizing RBF kernel parameters.

NSL-KDD dataset [25] is used in the experiment. This dataset has five classes, namely Normal, Probe, DoS, U2R, and R2L. The experiments use the entire NSLK-DD training dataset, which contains 125,973 records. The composition of the attack class as follows: Normal 67343 records, DoS 45927 records, Probe 11656 records, R2L 995 records, U2R 52 records. We test it using the 10-fold cross-validation method. Information about NSL-KDD dataset and its attacks can be found at [26].

Three non-numeric attributes in the dataset, namely flag, service, and protocol_type, are converted into numeric by categorizing it into the appropriate integers. Before starting the experiment, the log normalization process is firstly performed and proceed with the feature selection. The 17 attributes obtained from the feature selection process are shown in Table 2. We use the following metrics to measure the performance of models.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{12}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{13}$$

$$Specificity = \frac{TN}{TN+FP} \tag{14}$$

$$Gmean = \sqrt{Sensitivity \times Specivicity} \tag{15}$$

$$False\ Positive\ Rate = \frac{FP}{FP+TN} \tag{16}$$

$$False\ Negative\ Rate = \frac{FN}{FN+TP} \tag{17}$$

Accuracy is the fraction of predictions our model got right. It is a measure of the closeness of the experimental value to the actual amount of the substance in the confusion matrix. Sensitivity or recall is accuracy on the positives samples. Specificity is accuracy on the negative samples. G-mean indicates the balance between classification performances on the majority and minority class. This metric was recommended in [27] as the product of the prediction accuracies for both classes, i.e., sensitivity and specificity.

TP (True Positive) denotes as the number of positive samples which are correctly predicted as positives. FP (False Positive), often referred to as false alarm; defines as the number of negative samples incorrectly classified as positives. TN (True Negative) refers to the number of negative samples correctly classified as negatives. FN (False Negative) is determined as the number of positive samples incorrectly assigned as negatives.

Table 2. Top 17 ranking attributes from modified rank based information gain feature selection methods

Rank	Attributes	Description
1	src_bytes	The number of data bytes from source to destination
2	service	Network service on the destination, e.g., HTTP, telnet, etc.
3	dst_bytes	The number of data bytes from destination to source
4	dst_host_srv_count	Service count for destination host
5	hot	The number of “hot” indicators
6	dst_host_same_src_port_rate	Same source port rate for destination host
7	dst_host_srv_diff_host_rate	Different host rate for destination host
8	srv_count	The number of connections to the same service as the current connection in the past two seconds
9	count	The number of connections to the same host as the current connection in the past two seconds
10	duration	Length of the connection (seconds)
11	dst_host_count	Destination host count
12	is_guest_login	1 if the login is a “guest” login; 0 otherwise
13	srv_diff_host_rate	Percentage of connections to different hosts
14	dst_host_diff_srv_rate	Different service count for destination host
15	dst_host_rerror_rate	R-error rate for destination host
16	protocol_type	Type of the protocol, e.g., TCP, UDP, etc.
17	dst_host_srv_serror_rate	Srv-serror for destination host

6.1 Performance analysis of proposed IDS model

Firstly, we will show that modified-RIGFS method can improve the detection accuracy in minority class and keep capable of detecting another with high accuracy. We compare the accuracy of the SVM-based IDS which processes the subset of features produced by two feature selection method, original-RIGFS method and modified-RIGFS method. The experiment is conducted by using NSL-KDD dataset with a number of features from 2 to 32.

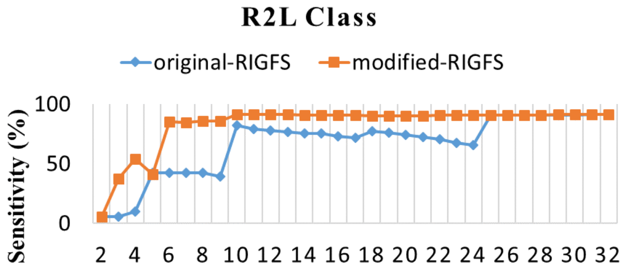

Figure.2 Sensitivity comparison in R2L class

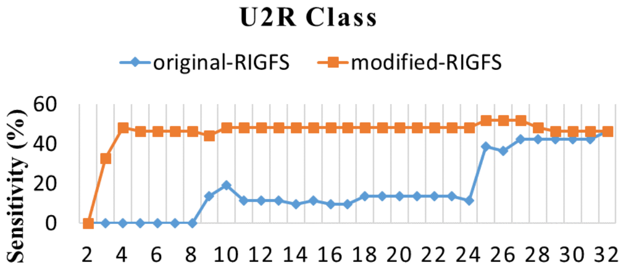

Figure.3 Sensitivity comparison in U2R class

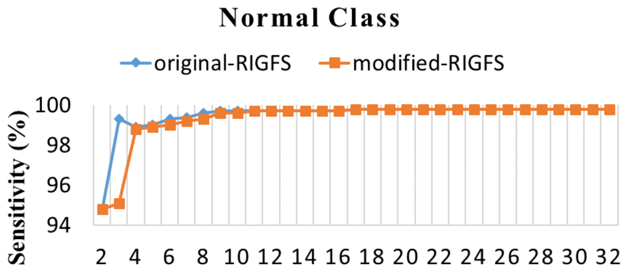

Figure.4 Sensitivity comparison in Normal class

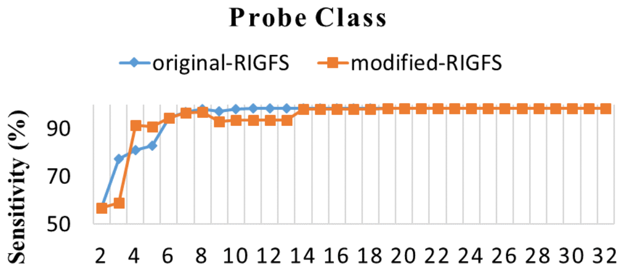

Figure.5 Sensitivity comparison in Probe class

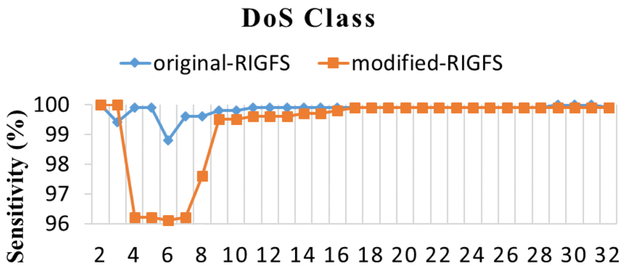

Figure.6 Sensitivity comparison in DoS class

Fig. 2-6 show the comparison of sensitivity in each class attack, namely R2L, U2R, Normal, Probe, and DoS. Figs. 2 and 3 show that the modified-RIGFS method causes a significant increase in the detection of minority classes. This method is superior in R2L and U2R classes, especially if the number of features is less than 25. This happens because the modified-RIGFS method can place the most influential features for R2L and U2R detection in the top 10 ranking features. Information about the ten features is in rows 2 to 11 in Table 2.

As shown in Fig. 2, the highest sensitivity in the R2L class can be achieved by using the top 10 features from the feature selection with modified-RIGFS method, whereas with the original feature selection method requires the top 25 features to reach that sensitivity level.

Fig. 3 also show that the highest sensitivity in the U2R class begins to be achieved by using the top 10 features of the selection of features with modified-RIGFS method, while the original-RIGFS method requires the top 32 features to achieve same sensitivity level.

The modified-RIGFS method can keep performance detection of majority classes high, especially if the number of features is more than 16. This condition is shown in Fig. 4 to Fig. 6.

Fig. 4 shows the decreasing in detection performance in the Normal class only occurs when using less than 10 top ranking features. Fig. 5 indicates the decreasing in detection performance in the Probe class that occurs in the use of datasets whose feature amount are 3, 9, 10, 11, 12, and 13. Whereas in the DoS class, in Fig. 6, the decreasing in detection performance occurs in the use of datasets whose feature amount is less than 17.

Observation of detection performance in the whole class is shown in Fig. 7 to Fig. 11. It sequentially shows a comparison of two feature selection methods in Accuracy, Sensitivity, Specificity, G-mean, and training time.

Accuracy graph at Fig. 7, Sensitivity graph at Fig. 8, Specificity graph at Fig. 9, and the G-mean graph at Fig. 10 show an increase when using modified-RIGFS method, from 13 until 17 top ranking features. When it uses more than 17 top ranking features, the graphics tend to be stable, the Accuracy is stable at a range of 99.6%, the Sensitivity is stable at a range of 99.7%, and the G-mean is stable at a range of 99.6%. Therefore, we chose a dataset with 17 features to be used in the intrusion detection model that we proposed.

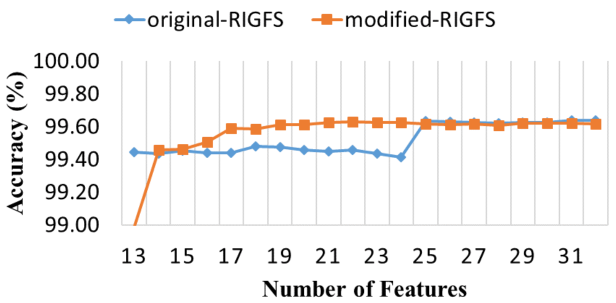

Figure.7 Accuracy comparison in the entire class

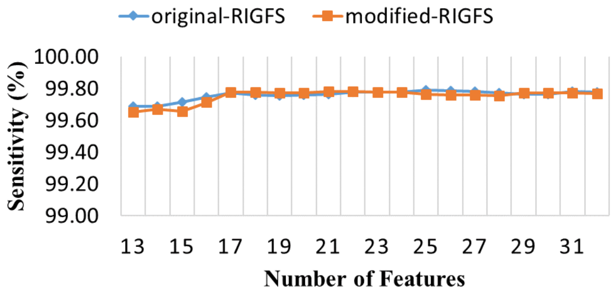

Figure.8 Sensitivity comparison in the entire class

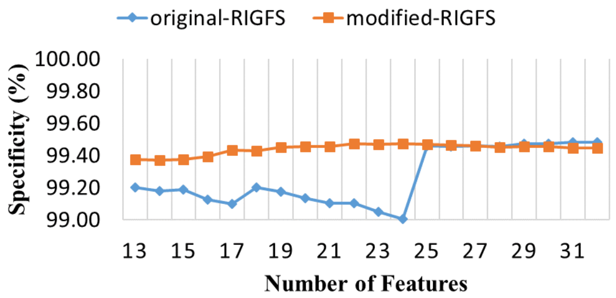

Figure.9 Specificity comparison in the entire class

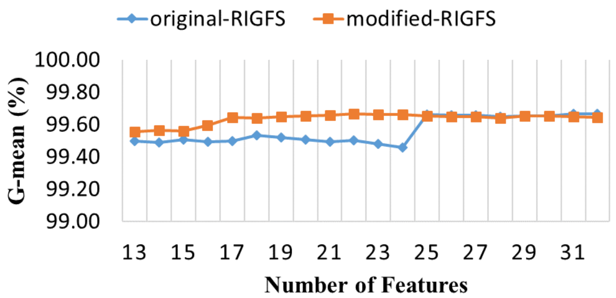

Figure.10 G-mean comparison in the entire class.

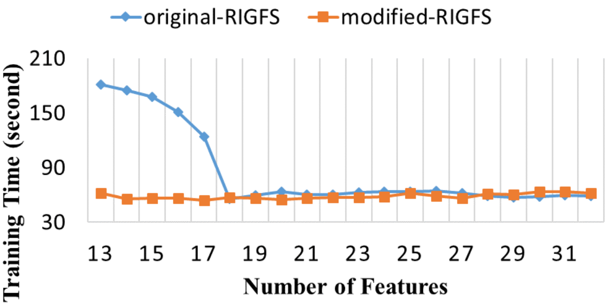

Figure.11 The comparison of training time

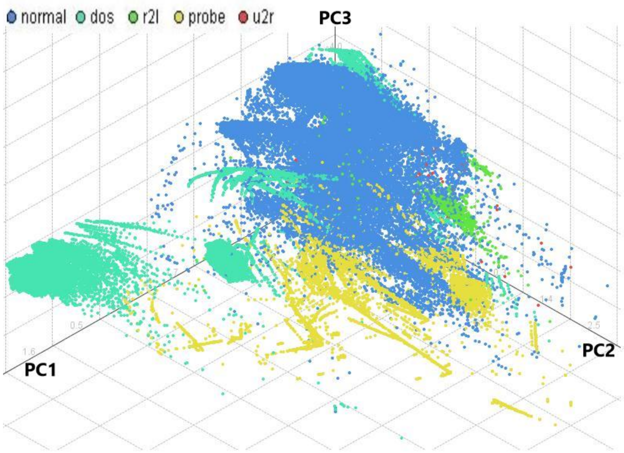

Figure.12 3D Scatter Plot of the dataset with 17 features that are produced by the original-RIGFS method

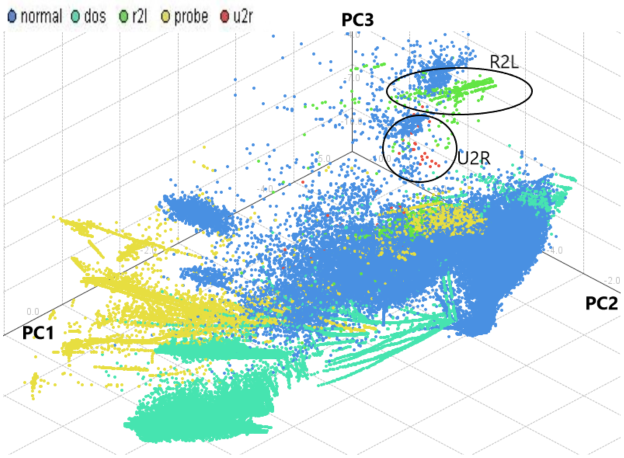

Figure.13 3D Scatter Plot of the dataset with 17 features that are produced by the modified-RIGFS method

Fig. 11 shows that the training time of SVM models that use our proposed feature selection method (modified-RIGFS) is faster than the SVM models with the original-RIGFS method, especially in a number of features below 18.

Those conditions show that our proposed feature selection method produces a subset with fewer features that can decrease training time and improve detection performance in all attack classes.

Next, we observe and compare 3D Scatter Plots from both datasets with these 17 features. By using Principal Component Analysis, we transform 17 features into 3 Principal Components (PC). Furthermore, the data points are visualized to 3D Scatter Plot using three PCs (PC1, PC2, and PC3) as x, y, and z coordinates.

Fig. 12 and 13 show 3D Scatter Plots from both datasets with 17 features obtained from both feature selection methods. The position of data points of the U2R (red) class and the R2L (green) class in Fig. 13 are more compact than in Fig.12.

Table 3. Comparison performance SVM with C=1.0 and gamma= 0.0

	All Features	original-RIGFS 17 features	modified-RIGFS 17 features
	(%)	(%)	(%)
Accuracy	99.621	99.441	99.590
Sensitivity	99.770	99.771	99.779
Specificity	99.461	99.098	99.434
FP Rate	0.539	0.902	0.566
FN Rate	0.230	0.229	0.221
G-mean	99.615	99.495	99.644
	(seconds)	(seconds)	(seconds)
Training time	54.645	109.220	54.228
Testing time	4.364	5.936	4.023

Table 4. Confusion matrix obtained from SVM with all features

		Prediction				
		Normal	DoS	R2L	Probe	U2R
Actual	Normal	67188	12	59	83	1
	DoS	20	45903	0	4	0
	R2L	86	0	908	0	1
	Probe	179	1	0	11476	0
	U2R	31	0	0	0	21

Table 5. Confusion matrix obtained from SVM with 17 features original-RIGFS

		Prediction				
		Normal	DoS	R2L	Probe	U2R
Actual	Normal	67189	13	68	73	0
	DoS	31	45891	0	5	0
	R2L	274	4	711	5	1
	Probe	179	2	2	11473	0
	U2R	45	0	0	2	5

Table 6. Confusion matrix obtained from SVM with 17 features modified-RIGFS

		Prediction				
		Normal	DoS	R2L	Probe	U2R
Actual	Normal	67194	19	54	71	5
	DoS	48	45875	0	4	0
	R2L	92	0	900	1	2
	Probe	165	29	0	11462	0
	U2R	27	0	0	0	25

Table 7. Confusion matrix obtained from the proposed model with optimal parameter

		Prediction				
		Normal	DoS	R2L	Probe	U2R
Actual	Normal	67234	24	37	40	8
	DoS	16	45909	1	1	0
	R2L	58	0	933	1	3
	Probe	37	8	1	11610	0
	U2R	10	0	4	0	38

Table 8. The accuracy obtained from the optimizing RBF kernel parameters using a grid search

C	gamma	Accuracy (%)
0.001	0.00100	83.679
0.006	0.00208	89.933
0.040	0.00432	94.148
0.251	0.00897	98.150
1.585	0.01864	99.271
10.000	0.03873	99.611
63.096	0.08047	99.748
398.107	0.16721	99.743
2511.886	0.34743	99.729
15848.932	0.72191	99.698
100000.000	1.50000	99.608
398.107	**0.08047**	**99.802**

Table 3 shows that SVM with 17 features from modified-RIGFS is better than SVM with 17 features from original-RIGFS for all performance measures tested. While it is better than SVM with all features dataset in training time, testing time, sensitivity, and false negative rate, this condition can also be observed in the confusion matrix in Table 4, 5, and 6. The columns in the matrix show predicted values, the rows show actual values, and the diagonal entries present the correct prediction.

To improve the detection performance of the SVM based IDS on the use of datasets with 17 features, we perform optimizing RBF kernel parameters using the grid search method.

Table 8 shows some of the results in the C and gamma parameter optimization process. We test the model on various values of C and gamma. The range C values are from 0.001 to 100000, while the range value of the gamma-range starts from 0.001 to 1.5. Both parameters are set by adding ten steps using the logarithmic scale, so 121 combinations of pairs of C and gamma parameters are formed. The pairs of C and gamma values that produce the best accuracy is 398.107 and 0.08047.

After the parameter optimization process, the performance produced by SVM with the proposed feature selection method is superior compared to others. The following are the performance values using optimal parameters: Accuracy 99.802%, Sensitivity 99.838%, Specificity 99.794%, FPR 0.206%, FNR 0.162%, G-mean 99.829%, training time 56.603 seconds, and testing time 2.094 seconds. There is also an increase in accuracy in all classes. This condition can be observed in the confusion matrix in Table 6 and 7.

These results indicate that the proposed model can fulfill the accuracy aspect and completeness aspect, which are important criteria for any intrusion detection model. Next, we compare the performance of the proposed model with the other models. Table 9 shows the performance comparison between the proposed model and the previous IDS models. While Fig. 14 presents a graph that compares the accuracy of those IDS models.

The classification accuracy of the proposed model in overall class is observed to be higher compared to the others approach. Similarly, in majority classes (Normal, DoS, Probe), the proposed model accuracy is also found to be higher than the others. Whereas in minority classes (R2L and U2R), the proposed model accuracy is ranked third and fifth of the eight models observed. However, those models that rank higher in minority classes use more features and also use fewer data samples for training and testing.

6.2 Discussions

The proposed model is a combination of multi-class SVM optimized by tuning parameter techniques, log normalization, and modified-RIGFS method. This method is dissimilar with the approach commonly used to avoid high dimensional curses in large data sets.

The use of log normalization method is done to prevent the change in the value of mutual information from features caused by the use of decimal places that are not suitable when carrying out the process of rounding the normalization results. The modified-RIGFS method is created to produce a subset of features that support better detection in minority classes. Multi-class SVM in one-against-one mode is implemented to get high detection accuracy in all classes. Furthermore, the parameter optimization of the SVM model is carried out to produce better predictions.

The novelty of this approach is the use of log normalization to avoid loss of information at the normalization stage and efforts to obtain a better feature subset in supporting minority class detection carried out with multi-class SVM.

7. Conclusions

The intrusion detection model proposed in this study uses a log normalization method, a modified rank based information gain feature selection (modified-RIGFS), and multi-class SVM with a parameter optimization technique. The results of the investigation on the NSL-KDD dataset show that our proposed model can increase accuracy and fulfill Completeness aspects.

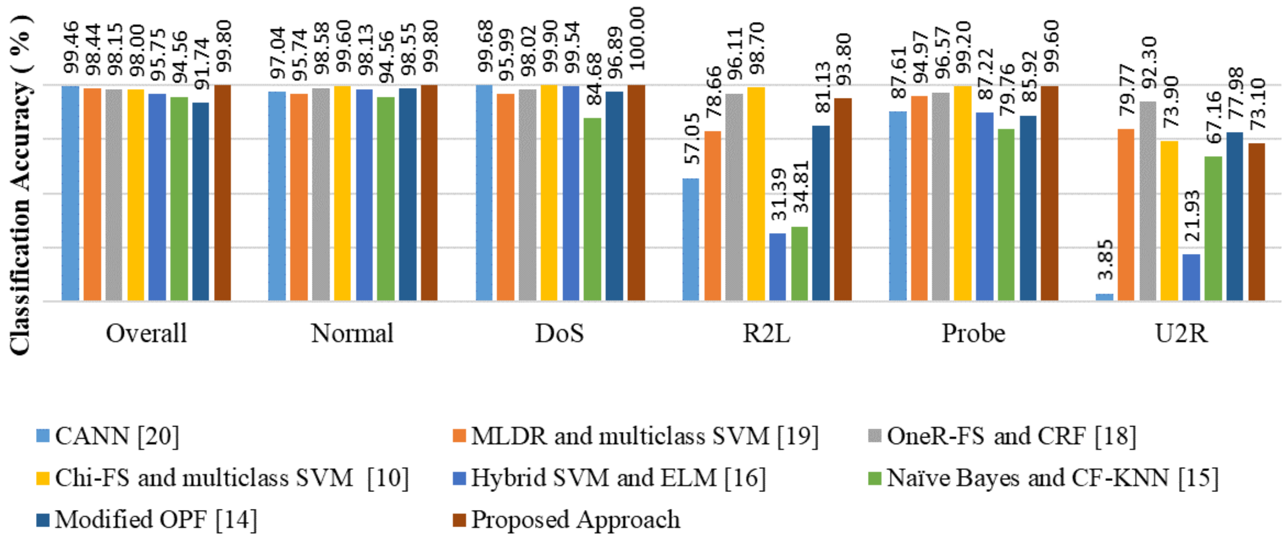

Figure.14 Comparison with other models based on accuracy

Table 9. Performance comparison of the various IDSs

Methods	∑	**Accuracy**					
	features	**Overall**	**Normal**	**DoS**	**R2L**	**Probe**	**U2R**
Proposed Approach	17	**99.80**	**99.80**	**100.00**	93.80	**99.60**	73.10
CANN [20]	19	99.46	97.04	99.68	57.05	87.61	3.85
MLDR and multi-class SVM [19]	---	98.44	95.74	95.99	78.66	94.97	79.77
OneR-FS and CRF [18]	24	98.15	98.58	98.02	96.11	96.57	**92.30**
GSPSO and ANN [17]	---	98.13	---	---	---	---	---
Chi-FS and multi-class SVM [10]	31	98.00	99.60	99.90	**98.70**	99.20	73.90
Hybrid SVM and ELM [16]	---	95.75	98.13	99.54	31.39	87.22	21.93
Naïve Bayes and CF-KNN [15]	---	94.56	94.56	84.68	34.81	79.76	67.16
Modified OPF [14]	---	91.74	98.55	96.89	81.13	85.92	77.98
LA-IDS [13]	7	90.40	---	---	---	---	---
Fuzziness semi-supervised [12]	55	84.12	---	---	---	---	---

Our proposed model can increase the detection of minority classes (R2L and U2R) and be able to keep another class accuracy high. The accuracy of R2L and U2R classes are 93.8% and 73.1% while the accuracy of the majority class (Normal, DoS, Probe) can be maintained above of 99.0%.

For future enhancements, we want to implement this approach to others IDS dataset and develop a hybrid SVM with other techniques for parameter optimization.

References

[1] G. M. Nazer, "Current Intrusion Detection Techniques in Information Technology - A Detailed Analysis", *European Journal of Scientific Research.*, Vol. 65, No. 4, pp. 611–624, 2011.

[2] S. V. Yeruru and T. R. Rangaswamy, "An anomaly-based intrusion detection system with multi-dimensional trust parameters for Mobile Ad Hoc Network", *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 4, pp. 81–90, 2017.

[3] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems", *Computer Networks*, Vol. 31, No. 8, pp. 805–822, 1999.

[4] H. He and A. Ghodsi, "Rare Class Classification by Support Vector Machine", In: *Proc. of the 20th International Conference on Pattern Recognition*, pp. 548–551, 2010.

[5] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets", *Knowledge and information systems*, Vol. 25, No. 1, pp. 1–20, 2010.

[6] A. Ozgur and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015", *PeerJ Prepr.*, Vol. 4, pp. 0–21, 2016.

[7] A. Tesfahun and D. L. Bhaskari, "Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction", In: *Proc. of the 2013 International Conference on Cloud*

and Ubiquitous Computing and Emerging Technologies, pp. 127–132, 2013.

[8] M. J. Abdi, S. M. Hosseini, and M. Rezghi, "A Novel Weighted Support Vector Machine Based on Particle Swarm Optimization for Gene Selection and Tumor Classification", *Computational and Mathematical Methods in Medicine*, Vol. 2012, pp. 1–7, 2012.

[9] Y. Liu, X. Yu, J. Xiangji, and A. An, "Combining integrated sampling with SVM ensembles for learning from imbalanced datasets", *Information Processing and Management*, Vol. 47, No. 4, pp. 617–631, 2011.

[10] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM", *Journal of King Saud University-Computer and Information Sciences*, Vol. 29, No. 4, pp. 462–472, 2017.

[11] W. Li and Z. Liu, "A method of SVM with normalization in intrusion detection", *Procedia Environmental Sciences*, Vol. 11, No. PART A, pp. 256–262, 2011.

[12] R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, and Y. L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system", *Inf. Sci. (Ny).*, Vol. 378, pp. 484–497, 2017.

[13] S. Jamali and P. Jafarzadeh, "An intelligent intrusion detection system by using hierarchically structured learning automata", *Neural Comput. Appl.*, Vol. 28, No. 5, pp. 1001–1008, 2017.

[14] H. Bostani and M. Sheikhan, "Modification of supervised OPF-based intrusion detection systems using unsupervised learning and social network concept", *Pattern Recognit.*, Vol. 62, pp. 56–72, 2017.

[15] H. H. Pajouh, G. H. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: a machine learning approach", *Journal of Intelligent Information Systems*, Vol. 48, No. 1, pp. 61–74, 2017.

[16] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system", *Expert Systems with Applications*, Vol. 67, pp. 296–303, 2017.

[17] T. Dash, "A study on intrusion detection using neural networks trained with evolutionary algorithms", *Soft Computing*, Vol. 21, No. 10, pp. 2687–2700, 2017.

[18] A. Mahendiran and R. Appusamy, "An intrusion detection system for network security situational awareness using conditional random fields", *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 3, pp. 196–204, 2018.

[19] B. N. Kumar, M. S. V. S. B. Raju, and B. V. Vardhan, "Enhancing the performance of an intrusion detection system through multi- linear dimensionality reduction and Multi-class SVM", *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 1, pp. 181–192, 2018.

[20] W. C. Lin, S. W. Ke, and C. F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors", *Knowledge-Based Systems*, Vol. 78, No. 1, pp. 13–21, 2015.

[21] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", In: *Proc. of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[22] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines", *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, pp. 415–425, 2002.

[23] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Fourth Edi. Morgan Kaufmann, 2016.

[24] C. Chang and C. Lin, "LIBSVM : A Library for Support Vector Machines", *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, p. 27, 2011.

[25] Canadian-Institute, "NSL-KDD dataset," 2009. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.

[26] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", In *Proc. of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.

[27] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection", *ICML*, Vol. 97, pp. 179–186, 1997.

[28] I. Mierswa, M. Wurst, and R. Klinkenberg, "YALE : Rapid Prototyping for Complex Data Mining Tasks", In: *Proc. of the 12th ACM SIGKDD Int. Conf. Knowl. Discov. data Min.*, pp. 935–940, 2006.