Intrusion Detection Using Combination of Various Kernels Based Support Vector Machine

Md. Al Mehedi Hasan, Mohammed Nasser, Biprodip Pal, Shamim Ahmad

Abstract - The success of any Intrusion Detection System (IDS) is a complicated problem due to its nonlinearity and the quantitative or qualitative network traffic data stream with many features. To get rid of this problem, several types of intrusion detection methods have been proposed and shown different levels of accuracy. This is why, the choice of the effective and robust method for IDS is very important topic in information security. In this paper, a combining classification approach to network intrusion detection based on the fusion of multiple classifiers is proposed. This approach makes a combination of various kernel based Support Vector Machine (SVM) classifier for intrusion detection system using majority voting fusion strategy. The experimental results indicate that combined approach effectively generates a more accurate model compared to single kernel based SVM classifier for the problem of intrusion detection.

Index Terms: Intrusion Detection, Combined Classifier, Support Vector Machine, Kernel, KDD99 Dataset.

1. Introduction

Along with the benefits, the Internet also created numerous ways to compromise the stability and security of the systems connected to it. Although static defense mechanisms such as firewalls and software updates can provide a reasonable level of security, more dynamic mechanisms such as intrusion detection systems (IDSs) should also be utilized [1]. As network attacks have increased in number and severity over the past few years, Intrusion Detection Systems (IDSs) have become a necessary addition to the security infrastructure of most organizations [2]. Deploying highly effective IDS systems is extremely challenging and has emerged as a significant field of research, because it is not theoretically possible to set up a system with no vulnerabilities [3]. Several machine learning (ML) algorithms, for instance Neural Network [4], Genetic Algorithm [5], Support Vector Machine [6, 7], clustering algorithm [8] and more have been extensively employed to detect intrusion activities from large quantity of complex and dynamic datasets.

Because different classifiers often expose different pros and cons, it is very difficult to determine a single classifier that has perfect generalization ability to the detection of unforeseen attacks. Therefore, the use of ensemble systems avoids the risk of mistake in choosing a poor or inappropriate classifier as the target intrusion detection model [9]. There are many types of ensemble proposed in the machine learning literature. Many studies have applied the diversity of ensemble methods to the intrusion detection problem [10, 11, 12, 13, 14, 15, 16].

Literature survey showed that, most of the researchers used randomly generated records or a portion of record from the KDD'99 dataset to develop multiple Classifier based intrusion detection system [12, 13, 14] without using the whole train and test dataset. So, those finding will not find out the actual performance for classification on the KDD'99 dataset. Although some researcher used the whole dataset but did not remove redundant records which is also a problem because of having redundant record

- ***Md. Al Mehedi Hasan***
  is working as Assistant Professor of Computer Science and Engineering at Rajshahi University of Engineering & Technology(RUET),Bangladesh. Email:mehedi_ru@yahoo.com
- ***Mohammed Nasser***
  is working as Professor of Statistics Department at University of Rajshahi, Bangladesh. Email:mnasser.ru@gmail.com
- ***Biprodip Pal***
  is working as Lecturer of Computer Science and Engineering at Rajshahi University of Engineering and technology, Bangladesh. Email:biprodip.cse@gmail.com
- ***Shamim Ahmad***

is working as Professor of Computer Science and Engineering at University of Rajshahi, Bangladesh.Email:shamim_cst@yahoo.com

classification methods may be biased toward to the class that has redundant record [9, 15, 16]. These limitations motivated us to build a combined classifiers based on the whole train and test dataset of KDD'99 by removing redundant record.

The general formulation to the design problem of an ensemble system is to generate several individual classifiers, and then employ some fusion functions (e.g., majority voting) to combine classifier outputs to achieve high performance. Because each type of classifier can produce different results, ensemble method takes advantages of the strong points of each individual classifier to induce a better final outcome. In this paper we propose a new combining classifier approach to intrusion detection by considering a set of homogeneous classifiers. Three different kernel based SVM classifiers perform classification over an input pattern and results are then combined using majority voting methodology.

The remainder of the paper is organized as follows. Section 2 provides the description of the KDD'99 dataset. We outline mathematical overview of SVM and algorithm for Combining Classifiers in Section 3 and 4 respectively. Experimental setup is presented in Section 5 and Preprocessing, Evaluation Metrics and SVM model selection are drawn in Section 6, 7 and 8 respectively. Finally, Section 9 reports the experimental result followed by conclusion in Section 10.

2. KDDCUP'99 Dataset

Under the sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), MIT Lincoln Laboratory has collected and distributed the datasets for the evaluation of researches in computer network intrusion detection systems [17]. The KDD'99 dataset is a subset of the DARPA benchmark dataset prepared by Sal Stofo and Wenke Lee [18]. The KDD data set was acquired from raw tcpdump data for a length of nine weeks. It is made up of a large number of network traffic activities that include both normal and malicious connections. A connection in the KDD-99 dataset is represented by 41 features, each of which is in one of the continuous, discrete and symbolic form, with significantly varying ranges. The KDD99 data set includes three independent sets; ''whole KDD'', ''10% KDD'', and ''corrected KDD''. Most of researchers have used the ''10% KDD'' and the ''corrected KDD'' as training and testing set, respectively [19]. The training set contains a total of 22 training attack types. The ''corrected KDD'' testing set includes an additional 17 types of attack and excludes 2 types (spy, warezclient) of attack from training set, so therefore there are 37 attack types that are included in the testing set, as shown in Table I and Table II. The simulated attacks fall in one of the four categories [1, 19]: (a) Denial of Service Attack (DoS), (b) User to Root Attack (U2R), (c) Remote to Local Attack (R2L), (d) Probing Attack.

Table 1. Attacks in KDD'99 Training dataset

Classification of Attacks	Attack Name
Probing	Port-sweep, IP-sweep, Nmap, Satan
DoS	Neptune, Smurf, Pod, Teardrop, Land, Back
U2R	Buffer-overflow, Load-module, Perl, Rootkit
R2L	Guess-password, Ftp-write, Imap, Phf, Multihop, spy, warezclient, Warezmaster

Table 2. Attacks in KDD'99 Testing dataset

Classification of Attacks	Attack Name
Probing	Port-Sweep, Ip-Sweep, Nmap, Satan, **Saint, Mscan**
DoS	Neptune, Smurf, Pod, Teardrop, Land, Back, **Apache2,Udpstorm, Processtable,Mail-Bomb**
U2R	Buffer-Overflow, Load-Module, Perl, Rootkit, **Xterm, Ps, Sqlattack.**
R2L	Guess-Password, Ftp-Write, Imap, Phf, Multihop, Warezmaster, **Snmpgetattack, Named, Xlock, Xsnoop, Send-Mail, Http-Tunnel, Worm, Snmp-Guess.**

2.1 Inherent Problems of the Kdd'99 and our proposed solution

Statistical analysis on KDD'99 dataset found important issues which highly affects the performance of evaluated systems and results in a very poor evaluation of anomaly detection approaches [20]. The most important deficiency in the KDD data set is the huge number of redundant records. Analyzing KDD train and test sets, Mohbod Tavallaee found that about 78% and 75% of the records are duplicated in the train and test set, respectively [21]. This large amount of redundant records in the train set will cause learning algorithms to be biased towards the more frequent records, and thus prevent it from learning unfrequent records which are usually more harmful to networks such as U2R attacks. The existence of these repeated records in the test set, on the other hand, will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records.

To solve these issues, we have developed a new data set, KDD99Train+ and KDD99Test+, which does not include any redundant records in the train set as well as in the test set, so the classifiers will not be biased towards more frequent records. The numbers of records in the train and test sets are now reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion.

3. SVM classification

Consider the problem of separating the set of training vectors belong to two separate classes, $(x_1, y_1), (x_2, y_2), \dots\dots, (x_n, y_n)$ where $x_i \in R^p and y_i \in \{-1, +1\}$ is the corresponding class label, $1 \le i \le n$. The main task is to find a classifier with a decision function $f(\mathbf{x}, \theta)$ such that $y = f(\mathbf{x}, \theta)$, where y is the class label for $\mathbf{x}$, θ is a vector of unknown parameters in the function.

3.1 SVM classification

The theory of Support Vector Machine (SVM) is from statistics and the basic principle of SVM is finding the optimal linear hyperplane in the feature space that maximally separates the two target classes [22]. Geometrically, the SVM modeling algorithm finds an optimal hyperplane with the maximal margin to separate two classes, which requires to solve the following constraint problem can be defined as follows

$$minimize_{w,b}\frac{1}{2}\|w\|^2$$

Subject to:

$$y_i(w^T x_i + b) \ge 1, i = 1,2,3, \dots\dots, n \qquad (1)$$

To allow errors, the optimization problem now becomes:

$$min_{w,b}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

Subject to:

$$y_i(w^T x_i + b) \ge 1 - \xi_i, i = 1,2,3, \dots\dots, n \qquad (2)$$
$$\xi_i \ge 0 \ , i = 1,2,3, \dots\dots, n$$

Using the method of Lagrange multipliers, we can obtain the dual formulation which is expressed in terms of variables α_i [6, 7, 22]:

$$maximize_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

Subject to: $\sum_{i=1}^{n} y_i\alpha_i = 0, 0 < \alpha_i < C$ for all $i = 1,2,3, \dots\dots, n$ (3)

Finally, the linear classifier based on a linear discriminant function takes the following form

$$f(x) = \sum_i^n \alpha_i x_i^T x + b \qquad (4)$$

In many applications a non-linear classifier provides better accuracy. The naive way of making a non-linear classifier out of a linear classifier is to map our data from the input space X to a feature space F using a non-linear function $\emptyset: X \to F$. In the space F, the optimization takes the following form using kernel function [23]:

$$maximize_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j k(x_i, x_j)$$

Subject to:

$$\sum_{i=1}^{n} y_i\alpha_i = 0 \ , 0 < \alpha_i < C$$
for all $i = 1,2,3, \dots\dots, n$ (5)

Finally, in terms of the kernel function the discriminant function takes the following form:

$$f(x) = \sum_{i}^{n} \alpha_i k(x, x_i) + b$$

Support vector machines are formulated for two class problems. But because support vector machines employ direct decision functions, an extension to multiclass problems is not straightforward [6, 7]. There are several types of support vector machines that handle multiclass problems. We used here only One-vs-All multiclass support vector machines for our research work.

3.2 Kernel and its parameters selection:

A kernel function and its parameter have to be chosen to build a SVM classifier [6, 7]. In this work, three main kernels have been used to build SVM classifier. Linear kernel has not considered due to its lower performance [6]. They are

1. Polynomial kernel:
   $K(x_i, x_j) = (< x_i, x_j > +1)^d$, d is the degree of polynomial.
2. Gaussian kernel:
   $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$, σ is the width of the function.
3. Laplace Kernel:
   $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|}{2\sigma^2})$,$\sigma$ is the width of the function.

Training an SVM finds the large margin hyperplane, i.e. sets the parameters α_i (c.f. Equation 5). The SVM has another set of parameters called hyperparameters: The soft margin constant, C, and any parameters the kernel function may depend on (width of a Gaussian or Laplace kernel or degree of a polynomial kernel)[24]. The soft margin constant C adds penalty term to the optimization problem. For a large value of C, a large penalty is assigned to errors/margin errors and creates force to consider points close to the boundary and decreases the margin. A smaller value of C (right) allows to ignore points close to the boundary, and increases the margin.

Kernel parameters also have a significant effect on the decision boundary [24]. The degree of the polynomial kernel and the width parameter σ of the Gaussian kernel or Laplace Kernel control the flexibility of the resulting classifier. The lowest degree polynomial is the linear kernel, which is not sufficient when a non-linear relationship between features exists. Higher degree polynomial kernels are flexible enough to discriminate between the two classes with a sizable margin and greater curvature for a fixed value of the soft-margin constant. On the other hand in Gaussian Kernel, for a fixed value of the soft-margin constant, large values of σ the decision boundary is nearly linear. As σ decreases the flexibility of the decision boundary increases and small values of σ lead to over fitting [24].

A question frequently posed by practitioners is "which kernel should I use for my data?". There are several answers to this question. The first is that it is, like most practical questions in machine learning, data-dependent, so several kernels should be tried. That being said, we typically follow the following procedure: Try a kernel first, and then see if we can improve on its performance using other kernels [24].

4. Combining Classifiers

The main goal of combined classifiers is to determine the best achievable performance for attacks detection within the available classifiers. It has been observed that different classifier designs offer complementary information about the type of attacks to be detected, which could be combined to improve the performance of detecting different types of intrusion. A large number of combination methods have been proposed in the literature [13, 14, 15, 16]. In this paper, the focus on only the majority voting combination technique for combining multiple classifiers is addressed.

Voting algorithms take the outputs of some classifiers as input and select a class which has been selected by most of the classifiers as output. If most of the classifiers are agree on a class for a test pattern, the result of voting classifier is that class. But if each classifier has a different output, we select output of the classifier as output of voting classifier, which has a better accuracy rather than the other classifiers.

5. Dataset and Experimental setup

Investigating the existing papers on the anomaly detection which have used the KDD data set, we found that a subset of KDD'99 dataset has been used for training and testing instead of using the whole KDD'99 dataset [20, 21, 25, 26]. Existing papers on the anomaly detection mainly used two common approaches to apply KDD [21]. In the first, KDD'99 training portion is employed for sampling both the train and test sets. However, in the second approach, the training samples are randomly collected from the KDD train set, while the samples for testing are arbitrarily selected from the KDD test set. The basic characteristics of the original KDD'99 and our duplicate less (KDD99Train+ and KDD99Test+)intrusion detection datasets in terms of number of samples is given in Table III. Although the distribution of the number of samples of attack is different on different research papers, we have used the Table I and II to find out the distribution of attack [1, 2, 19]. In our experiment, whole train (KDD99Train+) dataset has been used to train our classifier and the test (KDD99Test+) set has been used to test the classifier. All experiments were performed using Intel core i5 2.27 GHz processor with 4GB RAM, running Windows 7.

To select the best model in model selection phase, we have drawn 10% samples from the training set (KDD99Train+) to tune the parameters of all kernel and another 10% samples from the training set (KDD99Train+) to validate those parameters, as shown in Table III. In our experiment, three different types of kernel have been used.

6. Pre-processing

SVM classification system is not able to process KDD99Train+ and KDD99Test+ dataset in its current format. SVM requires that each data instance is represented as a vector of real numbers. Hence preprocessing was required before SVM classification system could be built. Preprocessing contains the following processes: The features in columns 2, 3, and 4 in the KDD'99 dataset are the protocol type, the service type, and the flag, respectively. The value of the protocol type may be tcp, udp, or icmp; the service type could be one of the 66 different network services such as http and smtp; and the flag has 11 possible values such as SF or S2. Hence, the categorical features in the KDD dataset must be converted into a numeric representation. This is done by the usual binary encoding – each categorical variable having possible m values is replaced with m-1 dummy variables. Here a dummy variable have value one for a specific category and having zero for all category. After converting category to numeric, we got 115 variables for each samples of the dataset. Some researchers used only integer code to convert category features to numeric representation instead of using dummy variables which is not statistically meaningful way for this type of conversion [19, 20].The final step of pre-processing is scaling the training data, i.e. normalizing all features so that they have zero mean and a standard deviation of 1. This avoids numerical

Table 3. Number of Samples of Each Attack in Dataset

Dataset	Normal	DoS	Probing	R2L	U2R	Total
WholeKDD (Original KDD)	972780	3883370	41102	1126	52	4898430
10% KDD (Original KDD)	97278	391458	4107	1126	52	494021
KDD corrected(Original KDD)	60593	229853	4166	16347	70	311029
KDD99Train+	87832	54572	2130	999	52	145585
KDD99Test+	47913	23568	2678	3058	70	77287
TrainSet(For Model Selection)	8784	5458	213	100	6	14561
ValidationSet(For Model Selection)	8784	5458	213	100	6	14561

instabilities during the SVM calculation. We then used the same scaling of the training data on the test set. Attack names were mapped to one of the five classes namely Normal, DoS (Denial of Service), U2R (user-to-root: unauthorized access to root privileges), R2L (remote-to-local: unauthorized access to local from a remote machine), and Probe (probing: information gathering attacks).

7. Evaluation Metrics

Apart from accuracy, developer of classification algorithms will also be concerned with the performance of their system as evaluated by False Negative Rate, False Positive Rate, Precision, Recall, etc. In our system, we have considered both the precision and false negative rate. To consider both the precision and false negative rate is very important in IDS as the normal data usually significantly outnumbers the intrusion data in practice. To only measure the precision of a system is misleading in such a situation [27]. The classifier should produce lower false negative rate because an intrusion action has occurred but the system considers it as a non-intrusive behavior is very cost effective.

8. SVM Model Selection

In order to generate highly performing SVM classifiers capable of dealing with real data an efficient model selection is required. In our experiment, Grid-search technique has been used to find the best model for SVM with different kernel. This method selects the best solution by evaluating several combinations of possible values. In our experiment, Sequential Minimization Optimization with the following options in Matlab, shown in Table IV, has been used. We have considered the range of the parameter in the grid search which converged within the maximum iteration using the train set (For Model Selection) and validation set (For Model selection) shown in Table III.

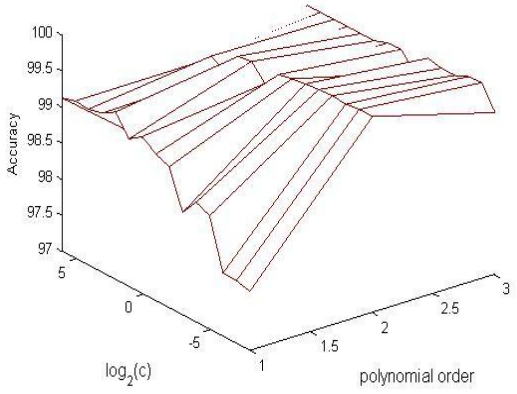

Fig. 1. Tuning Polynomial Kernel.

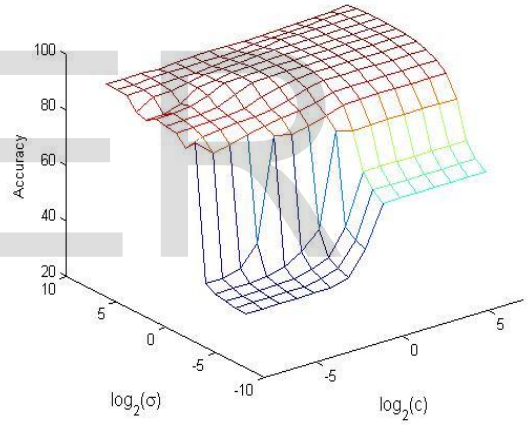

Fig.2. Tuning Radial Basis Kernel

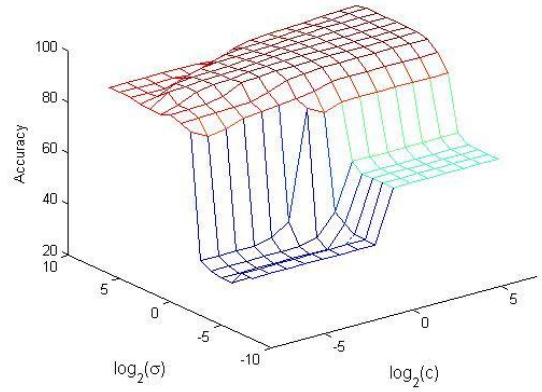

Fig.3. Tuning Laplace Kernel

Table 4. Sequential Minimization Optimization Options

Option	Value
MaxIter	1000000
KernelCacheLimit	10000

For polynomial kernel, to find the parameter value C (penalty term for soft margin) and d (poly order), we have considered the value from 2^{-8} to 2^{6} for C and from 1 to 3 for d as our searching space. The resulting search space for polynomial kernel is shown in Figure I. We took parameter value d=2 and C=0.0039 for giving us 99.70% accuracy in the validation set to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

For radial basis kernel, to find the parameter value C (penalty term for soft margin) and sigma, we have considered the value from 2^{-8} to 2^{6} for C and from 2^{-8} to 2^{6} for sigma as our searching space. The resulting search space for radial basis kernel is shown in Figure II. We took parameter value C=32 and sigma=16 for giving us 99.59% accuracy in the validation set to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

Again, for Laplace kernel, to find the parameter value C (penalty term for soft margin) and sigma, we have considered the value from 2^{-8} to 2^{6} for C and from 2^{-8} to 2^{6} for sigma as our searching space. The resulting search space for radial basis kernel is shown in Figure III. We took parameter value C=64 and sigma=8 for giving us 99.70% accuracy in the validation set to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

9. Obtained Result

The final training/test phase is concerned with the production and evaluation on a test set of the final SVM model created based on the optimal hyper-parameters set found so far in the model selection phase. After finding the parameter, we built the model using the whole train dataset (KDD99Train+) for each of the kernel tricks and tested the model using the test dataset (KDD99Test+). Finally, we have combined all of these kernels based SVM classifiers. The training and testing results are given in Table V according to the classification accuracy. From the results it is observed that the test accuracy for combined approach is better than single kernel based SVM classifier.

Table 5: Training and Testing Accuracy

Kernel	Training Accuracy	Testing Accuracy
Polynomial	99.73	91.27
Radial Basis	99.79	92.99
Laplace	99.97	93.19
Combining Approach	99.85	93.22

For the test case, the confusion matrix for each of the kernel and the combining approach are given in Table VI, VII, VIII and IX respectively. Going into more detail of the confusion matrix, it can be seen that Radial Basis kernel performs better on attack R2L detection and Laplace kernel performs well on Dos, probing, and U2R detection. Finally, combined approach produces moderately better result in all types of attack detection by taking the advantages of each individual classifier.

We also considered the false negative rate (%) and precision (%) for each of kernel and combined approach as shown in Table X and XI respectively. The Radial Basis kernel gives lower average false negative rate and Laplace gives higher precision. On the other hand, Combining Approach gives moderately lower false negative rate and moderately higher precision than other kernels.

Table 6: Confusion matrix for Polynomial Kernel

		Actual					
		Dos	Normal	Probing	R2L	U2R	%
Prediction	Dos	21317	115	380	5	16	97.64
	Normal	1988	47184	724	2424	28	90.14
	Probing	263	521	1524	105	6	63.00
	R2L	0	62	24	511	15	83.50
	U2R	0	31	26	13	5	6.67
	%	90.45	98.48	56.91	16.71	7.14	

Table 7: Confusion matrix for Radial Basis Kernel

		Actual					
		Dos	Normal	Probing	R2L	U2R	%
Prediction	Dos	22663	187	643	18	18	96.32
	Normal	824	46984	473	2224	23	92.99
	Probing	68	672	1536	131	0	63.81
	R2L	13	60	22	680	19	85.64
	U2R	0	10	4	5	10	34.48
	%	96.16	98.06	57.36	22.24	14.29	

Table 8: Confusion matrix for Laplace Kernel

		Actual					
		Dos	Normal	Probing	R2L	U2R	%
Prediction	Dos	22715	96	627	19	8	96.80
	Normal	823	47301	456	2644	27	92.29
	Probing	30	499	1595	9	1	74.74
	R2L	0	13	0	386	11	94.15
	U2R	0	4	0	0	23	85.19
	%	96.38	98.73	59.56	12.62	32.86	

Table 9: Confusion matrix for Combining Approach

		Actual					
		Dos	Normal	Probing	R2L	U2R	%
Prediction	Dos	22652	102	616	10	14	96.82
	Normal	848	47304	481	2460	27	92.54
	Probing	68	481	1573	83	1	71.31
	R2L	0	13	4	500	13	94.34
	U2R	0	13	4	5	15	40.54
	%	96.11	98.73	58.74	16.35	21.43	

Table 10: False Negative Rate (%) of each Kernels and Combining Approach for each of the attack types

Kernel	Dos	Probing	R2L	U2R	Average False Negative Rate
Polynomial	8.43	27.04	79.27	40	38.69
Radial Basis	3.50	17.66	72.73	32.86	31.69
Laplace	3.5	17.03	86.46	38.57	36.39
Combining Approach	3.6	17.96	80.44	38.57	35.14

Table 11: Precision (%) of each Kernels and Combining Approach for each of the attack types

Kernel	Dos	Probing	R2L	U2R	Average Precision
Polynomial	98	63	83	7	63
Radial Basis	96	64	86	34	70
Laplace	96	74	94	85	87
Combining Approach	96	71	94	41	76

10. Conclusion

In this research work, we developed an intrusion detection system using support vector machines as classifier. The performances of the different kernel based approaches and a combining classifier approach have been observed on the basis of their accuracy, false negative rate and precision. The results show the effectiveness of classifier combination in providing more reliable results, as the final decision depends on the agreement among distinct classifiers. Research in intrusion detection using SVM and combined approach is still an ongoing area due to good performance. The findings of this paper will be very useful for future research and to use SVM more meaningful way in order to maximize the performance rate and minimize the false negative rate.

REFERENCES

[1] Kayacik H. G., Zincir-Heywood A. N., Heywood M. I., "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Benchmark", Proceedings of the PST 2005 – International Conference on Privacy, Security, and Trust, pp. 85-89, 2005.

[2] Hesham Altwaijry, Saeed Algarny, "Bayesian based intrusion detection system", Journal of King Saud University – Computer and Information Sciences, pp.1–6, 2012.

[3] O. Adetunmbi Adebayo, Zhiwei Shi, Zhongzhi Shi, Olumide S. Adewale, "Network Anomalous Intrusion Detection using Fuzzy-Bayes", IFIP International Federation for Information Processing, Vol: 228, pp: 525-530, 2007.

[4] Cannady J, "Artificial Neural Networks for Misuse Detection", in Proceedings of the '98 National Information System Security Conference (NISSC'98), pp. 443-456, 1998.

[5] Pal, B.; Hasan, M.A.M., "Neural network & genetic algorithm based approach to network intrusion detection & comparative analysis of performance," Computer and Information Technology (ICCIT), 2012 15th International Conference on, vol., no., pp.150,154, 22-24 Dec. 2012.

[6] Md. Al Mehedi Hasan, Mohammed Nasser and Biprodip Pal. "On The KDD'99 Dataset: Support Vector Machine Based Intrusion Detection System (IDS) with Different Kernels", International Journal of Electronics Communication and Computer Engineering, Volume 4.4, pp. 1164-1170, 2013.

[7] Md. Al Mehedi Hasan, Mohammed Nasser, "Intrusion Detection System (IDS) Using Support Vector Machine with Different Kernels",International Conference on Statistical Data Mining for Bioinformatics Health Agriculture and Environment, 21-24 December, 2012, pp. 628-635.

[8] Qiang Wang and Vasileios Megalooikonomou, "A clustering algorithm for intrusion detection", in Proceedings of the conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, vol. 5812, pp. 31-38, March 2005.

[9] Emna Bahri, Nouria Harbi and Hoa Nguyen Huu, "A Multiple Classifier System Using an Adaptive Strategy for Intrusion Detection" International Conference on Intelligent Computational Systems (ICICS'2012) Jan. 7-8, 2012 Dubai.

[10] Giacinto, Giorgio, Fabio Roli, and Luca Didaci. "Fusion of multiple classifiers for intrusion detection in computer networks."*Pattern recognition letters* 24.12 (2003): 1795-1803.

[11] Roberto Perdisci, Davide Ariu, Prahlad Fogla, Giorgio Giacinto, and Wenke Lee, "McPAD : A Multiple Classifier System for Accurate Payload-based Anomaly Detection", Preprint submitted to Elsevier Science, 18 November 2008

[12] Anazida Zainal1, Mohd Aizaini Maarof and Siti Mariyam Shamsuddin "Ensemble Classifiers for Network Intrusion Detection System" Journal of Information Assurance and Security 4 (2009) 217-225.

[13] Ali Borji, "Combining Heterogeneous Classifiers for Network Intrusion Detection", I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 254 – 260, 2007.

[14] Zainal, Anazida, et al. "Ensemble of one-class classifiers for network intrusion detection system." *Information Assurance and Security, 2008.ISIAS'08.Fourth International Conference on*.IEEE, 2008.

[15] Aljahdali, Sultan. "An effective intrusion detection method using optimal hybrid model of classifiers."

Journal of Computational Methods in Science and Engineering 10 (2010): 51-60.

[16] Natesan, P., P. Balasubramanie and G. Gowrison, "Improving the Attack Detection Rate in Network Intrusion Detection using Adaboost Algorithm"Journal of Computer Science 8 (7): 1041-1048, 2012, ISSN 1549-3636

[17] MIT Lincoln Laboratory, DARPA Intrusion Detection Evaluation, http://www.ll.mit.edu/CST.html,MA, USA. July, 2010.

[18] KDD'99 dataset, http://kdd.ics.uci.edu/databases, Irvine, CA, USA, July, 2010.

[19] M. Bahrololum, E. Salahi and M. Khaleghi, "Anomaly Intrusion Detection Design Using Hybrid Of Unsupervised And Supervised Neural Network", International Journal of Computer Networks & Communications (IJCNC), Vol.1, No.2, July 2009.

[20] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham, "Principle Components Analysis and Support Vector Machine based Intrusion Detection System", *10th International Conference on Intelligent Systems Design and Applications, 2010*.

[21] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[22] Vladimir N. Vapnik, The Nature of Statistical Learning Theory, Second Edition, Springer, New York, ISBN 0-387-98780-0, 1999.

[23] Bernhard Scholkopf, Alexander J. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond", The MIT Press Cambridge, Massachusetts London, England, 2001.

[24] A. Ben-Hur and J. Weston. "A User's guide to Support Vector Machines", In Biological Data Mining. Oliviero Carugo and Frank Eisenhaber (eds.) Springer Protocols, 2009.

[25] Fangjun KUANG, Weihong XU, Siyang ZHANG, Yanhua WANG, Ke LIU , "A Novel Approach of KPCA and SVM for Intrusion Detection", Journal of Computational Information Systems, pp. 3237-3244, 2012.

[26] Shilpalakhina, Sini Joseph and Bhupendraverma, "Feature Reduction using Principal Component Analysis for Effective Anomaly–Based Intrusion Detection on NSL-KDD", International Journal of Engineering Science and Technology Vol. 2(6), pp.1790-1799,2010.

[27] Jing Tao Yao, Songlun Zhao, and Lisa Fan, "An enhanced support vector machine model for intrusion detection", RSKT'06 Proceedings of the First international conference on Rough Sets and Knowledge Technology, Pages 538-543, 2006.