International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue 4, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

On the KDD'99 Dataset: Support Vector Machine Based Intrusion Detection System (IDS) with Different Kernels

Md. Al Mehedi Hasan, Mohammed Nasser, Biprodip Pal

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. Support vector machine (SVM) has been employed to provide potential solutions for the IDS problem. However, the practicability of SVM is affected due to the difficulty of selecting appropriate kernel and its parameters. Thus, this paper is aimed to use different kernel on the KDD'99 Dataset and find out which is best for SVM based intrusion detection system. In this work, 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 which was an inherent problem of KDD'99 dataset, so the classifiers will not be biased towards more frequent records. The experimental results indicate that RBF kernel can achieve higher detection rate than others kernel like Linear and polynomial kernel in the same time. RBF kernel also shows lower false negative rate than polynomial kernel.**

Keywords – **Intrusion Detection, KDD'99, Support Vector Machine, Kernel, Kernel Selection.**

I. 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]. Intrusion detection is the process of monitoring events occurring in a computer system or network and analyzing them for signs of intrusions. IDSs are simply classified as host-based or network-based. The former operates on information collected from within an individual computer system and the latter collect raw network packets and analyze for signs of intrusions. There are two different detection techniques employed in IDS to search for attack patterns: Misuse and Anomaly. Misuse detection systems find known attack signatures in the monitored resources. Anomaly detection systems find attacks by detecting changes in the pattern of utilization or behavior of the system [2].

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 [3]. 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 [4]. Several machine learning (ML) algorithms, for instance Neural Network [5], Genetic Algorithm [5,6], Fuzzy Logic [4, 7, 8, 9], clustering algorithm [10] and more have been extensively employed to detect intrusion activities from large quantity of complex and dynamic datasets.

In recent times, Support Vector Machine (SVM) has been extensively applied to provide potential solutions for the IDS problem. But, the selection of an appropriate kernel and its parameters for a certain classification problem influence the performance of the SVM because different kernel function constructs different SVMs and affects the generalization ability and learning ability of SVM. There is no theoretical method for selecting kernel function and its parameters. Literature survey showed that, for all practical purposes, most of the researchers applied Gaussian kernel to build SVM based intrusion detection system [11, 12, 13, 14] and find its parameter value using different technique which is not unique and some research paper did not mention its value [13]and some uses the default value of the software package [15]. But there are many other kernel functions which are not yet applied in intrusion detection. Other kernels should also be used in comparison to find optimal results for applying SVM based approach depending upon the nature of classification problem [13]. This motivated us to apply different kernel functions of SVM apart from RBF for IDS classification purpose which may provide better accuracy and detection rate depending on different nonlinear separations. We have also tried to find out parameter value to the corresponding kernel. In this paper, we provide a review of the SVM and its kernel approaches in IDS for future research and implementation towards the development of optimal approach in intrusion detection system with maximum detection rate and minimized false alarms.

The remainder of the paper is organized as follows. Section II provides the description of the KDD'99 and NSL-KDD dataset. We outline mathematical overview of SVM in Section III. Experimental setup is presented in Section IV and Preprocessing, Evaluation Metrics and SVM model selection are drawn in Section V, VI and VII respectively. Finally, Section VIII reports the experimental result followed by conclusion in Section IX.

II. Dataset

A. KDDCUP'99

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 [16]. The KDD'99 dataset is a subset of the DARPA benchmark dataset prepared by Sal Stofo and Wenke Lee [17].

Table I: 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,

The KDD data set was acquired from raw tcp dump 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 [18]. The training set contains a total of 22 training attack types. Additionally the ''corrected KDD'' testing set includes an additional 15 attack types and 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, 18]: (a) Denial of Service Attack (DoS), (b) User to Root Attack (U2R), (c) Remote to Local Attack (R2L), (d) Probing Attack.

Table II: 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, Spy, Warezclient, Warezmaster, **Snmpgetattack, Named, Xlock, Xsnoop, Send-Mail, Http-Tunnel, Worm, Snmp-Guess.**

B. KDDCUP'99Problems & 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 [13]. The most important deficiency in the KDD data set is the huge number of redundant records. Analyzing KDD train and test sets, MohbodTavallaeefound that about 78% and 75% of the records are duplicated in the train and test set, respectively [19]. 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.

III. 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 [20, 21]. There are two types of data namely linearly separable and non-separable data. To handle these data, two types of classifier, linear and non-linear, are used in pattern recognition field.

A. Linear Classifier

Consider the problem of separating the set of training vectors belong to two linear separate classes, $(x_1, y_1), (x_2, y_2), \dots\dots, (x_n, y_n)$ where $x_i \in R^n, y_i \in \{-1, +1\}$ with a hyperplane $w^T x + b = 0$. Finding a separating hyperplane can be posed as a constraint satisfaction problem. For this problem, the constraint problem can be defined as follows find w and b such that

$$w^T x_i + b \geq 1 \; if \; y_i = +1$$
$$w^T x_i + b \leq -1 \; if \; y_i = -1$$
$$where \; i = 1,2,3, \dots\dots, n$$

Considering the maximum margin classifier, there is hard margin SVM, applicable to a linearly separable dataset, and then modifies it to handle non-separable data. This leads to the following constrained optimization problem:

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

Subject to:

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

The constraints in this formulation ensure that the maximum margin classifier classifies each example correctly, which is possible since we assumed that the data is linearly separable. In practice, data is often not linearly separable and in that case, a greater margin can be achieved by allowing the classifier to misclassify some points. 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:

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

The constant C > 0 sets the relative importance of maximizing the margin and minimizing the amount of slack. This formulation is called the soft-margin SVM [20, 21]. Using the method of Lagrange multipliers, we can obtain the dual formulation which is expressed in terms of variablesα_i [20, 21]:

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

Subject to:

$$\sum_{i=1}^{n} y_i\alpha_i = 0, 0 < \alpha_i < C \qquad (3)$$

$$\text{for all } i = 1,2,3, \dots\dots, n$$

The dual formulation leads to an expansion of the weight vector in terms of the input examples:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

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

$$f(x) = \sum_{i}^{n} \alpha_i x_i^T x + b \qquad (4)$$

B. Non-linear Classifier

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 $\emptyset : X \rightarrow F$. In the space F, the discriminant function is:

$$f(x) = w^T\emptyset(x) + b.$$

Now, examine what happens when the nonlinear mapping is introduced into equation (3). We have to optimize

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

Subject to: $\sum_{i=1}^{n} y_i\alpha_i = 0{,}0 < \alpha_i < C$

$$\text{for all } i = 1,2,3, \dots\dots, n \qquad (5)$$

Notice that the mapped data only occurs as an inner product in the objectives. Now, we can apply a little mathematically rigorous magic known as kernels. By Mercer's theorem, we know that for certain mapping $\emptyset(x)$ and any two points x_i andx_j , the inner product of the mapped points can be evaluated using the kernel function without ever explicitly knowing the mapping [22] . The kernel function can be defined as

$$k(x_i, x_j) = \emptyset(x_i)^T\emptyset(x_j)$$

Substituting the kernel inthe equation 5, the optimization takes the following form:

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

Subject to:

$\sum_{i=1}^{n} y_i\alpha_i = 0\ {,}0 < \alpha_i < C$

For all $i = 1,2,3, \dots\dots, n$ (6)

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

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

C. Kernel and its parameters selection

A kernel function and its parameter have to be chosen to build a SVM classifier [14]. In this work, three main kernels have been used to build SVM classifier. They are

1. Linear kernel: $K(x_i, x_j) =< x_i, x_j >$
2. Polynomial kernel: $K(x_i, x_j) = (< x_i, x_j > +1)^d$, d is the degree of polynomial.
3. Gaussian kernel:

   $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma}\right)$, σ is the width of the function.

Training an SVM finds the large margin hyperplane, i.e. sets the parameters α_i (c.f. Equation 6). 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 kernel or degree of a polynomial kernel)[23]. 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 [23]. The degree of the polynomial kernel and the width parameter σ of the Gaussian 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, small values of σ the decision boundary is nearly linear. As σ increases the flexibility of the decision boundary increases and large values of σ lead to over fitting [23].

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 linear kernel first, and then see if we can improve on its performance using a non-linear kernel [23].

D. Multiclass support vector machine

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 [12]. 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. The One-Vs-All technique is extended from the binary two-class problem to perform classification tasks with k > 2 classes. In this approach, the base classifier (in our case - SVM) is trained on K copies of the K-class original training set, with each copy having the K-th label as the positive label, and all other labels as the negative label (combined class). We denote the optimal separating hyperplane discriminating the class j and the combined class as

$$g^j = x^{T}\hat{w}^j + \hat{b}^j, \qquad j = 1,,2,3,....,k$$

where the superscript in $\hat{w}^j$ stands for the class which should be separated from the other observations.After finding the all k optimal separating hyperplanes, the final classifierhas been defined by

$$f_k(x) = argmax_j(g^j(x))$$

In this approach the index of the largest component of the discriminant vector $(g^1(x), g^2(x), \dots\dots, g^k(x))$ is assigned to the vector x. In other words, each input is classified by all K models, and the output is chosen by the model with the highest degree of confidence.

IV. 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 [13, 15, 24, 25]. Existing papers on the anomaly detection mainly used two common approaches to apply KDD [15]. 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, 3, 18].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 (KDDTrain+) to tune the parameters of all kernel and another 10% samples from the training set (KDDTrain+) to validate those parameters, as shown in Table III. In our experiment, three different types of kernel have been used.

Table III: Number of Samples of Each Attack in Dataset

Dataset	Normal	DoS	Probing	R2L	U2R	Total
Whole KDD (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
Train Set(For Model Selection)	8784	5458	213	100	6	14561
Validation Set (For Model Selection)	8784	5458	213	100	6	14561

V. Pre-Processing

SVM classification system is not able to process KDD99Train+ and KDD99Test+ dataset in its current format. Hence preprocessing was required before SVM classification system could be built. Preprocessing contains the following processes: SVM requires that each data instance is represented as a vector of real numbers. 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 70 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 119 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 [13, 18].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 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).

VI. Evaluation Metrics

Apart from accuracy, developer of classification algorithms will also be concerned with the performance of their system as evaluated by FalseNegative 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 [26]. 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.

VII. 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 (SMO) 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 trainset (For Model Selection) and validation set (For Model selection) shown in Table III.

Table IV: SMO Options

Option	Value
MaxIter	1000000
KernelCacheLimit	10000

For linear kernel, to find out the parameter value C, we have considered the value from 2^{-8} to 2^{6} as our searching space. The resulting search space for linear kernel is shown in Figure I. We took parameter value C=4 for giving us 99.31% accuracy in the validationset to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

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 II. 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 III. We took parameter value C=32 and sigma=16 for giving us 99.01% accuracy in the validation set to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

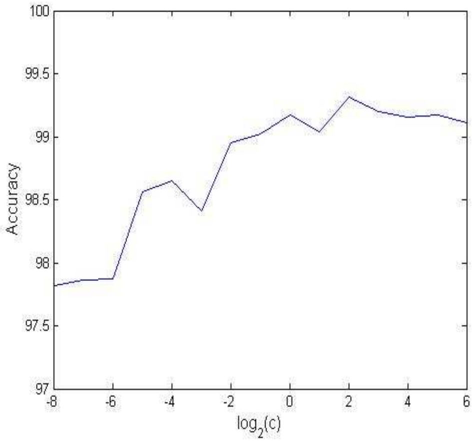

Fig.1. Tuning Linear Kernel

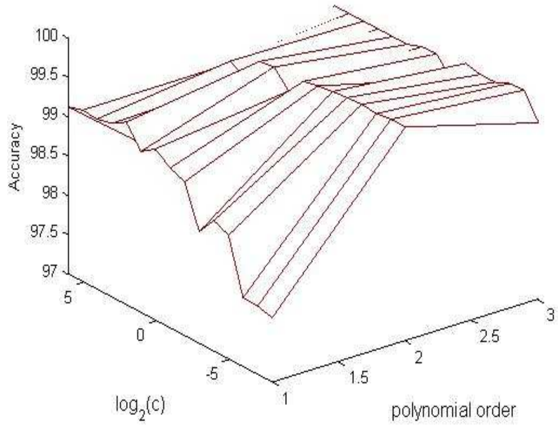

Fig.2. Tuning Polynomial Kernel

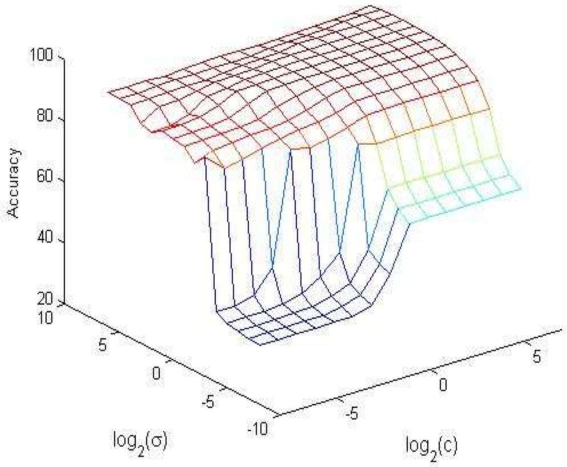

Fig.3. Tuning Radial Basis Kernel

VIII. 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 [19]. After finding the parameter, we built the model using the whole train dataset(KDD99Train+) for each of the kernel tricks and finally we have tested the model using the test dataset(KDD99Test+). 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 radial basis kernel is better than linear and polynomial kernel.

For the test case**,** the confusion matrix for each of the kernel is given in Table VI, VII and VIII respectively. Going into more detail of the confusion matrix, it can be seen that Linear kernel performs better on probing attack detection and RBF kernel performs well on Dos, R2L, and U2R detection. We also considered the false negative rate and precision for each of kernel and as shown in Table IX and X respectively. The linear kernel gives lower average false negative rate and on the other hand RBF kernel gives moderately lower false negative rate and high precision than other kernels.

Table V: Training and Testing Accuracy

Kernel	Training Accuracy	Testing Accuracy
Linear	77.82	36.93
Polynomial	99.73	91.27
Radial Basis	99.79	92.99

Table VI: Confusion matrix for Linear Kernel

	Actual						
		Dos	Normal	Probing	R2L	U2R	%
Prediction	Dos	21673	37036	310	954	34	36.18
	Normal	0	4047	0	0	0	100
	Probing	1498	6622	2353	1489	20	19.64
	R2L	397	175	10	468	12	44.07
	U2R	0	33	5	147	4	2.12
	%	91.96	8.45	87.86	15.30	5.71	

Table VII: 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 VIII: 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 IX: False Negative Rate of Different Kernels for each of the attack types.

Kernel	Dos	Probing	R2L	U2R	Average False Negative Rate
Linear	0	0	0	0	0
Polynomial	8.43	27.04	79.27	40	38.69
Radial Basis	3.50	17.66	72.73	32.86	31.69

Table X: Precision of Different Kernels for each of the attack types.

Kernel	Dos	Probing	R2L	U2R	Average Precision
Linear	0.36	0.20	0.44	0.02	0.26
Polynomial	0.98	0.63	0.83	0.07	0.63
Radial Basis	0.96	0.64	0.86	0.34	0.70

IX. 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 have been observed on the basis of their accuracy, false negative rate and precision. The results indicate that the ability of the SVM classification depends mainly on the kernel type and the setting of the parameters. Research in intrusion detection using SVM 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] H. G.Kayacik, A. N.Zincir-Heywood, M. I. Heywood, "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] A. A.Olusola., A.S.Oladele. And D.O.Abosede. "Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features", Proceedings of the World Congress on Engineering and Computer Science 2010 Vol I WCECS 2010, October 20-22, 2010.

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

[4] O. A. Adebayo, Z. Shi, Z. Shi, O. S. Adewale, "Network Anomalous Intrusion Detection using Fuzzy-Bayes", IFIP International Federation for Information Processing, Vol: 228, pp: 525-530, 2007.

[5] B. Pal., M. A. M. Hasan, "Neural Network & Genetic Algorithm Based Approach to Network Intrusion Detection & Comparative Analysis of Performance," Proceedings of the the 15th International Conference on Computer and Information Technology, Chittagong, Bangladesh, 2012.

[6] S. M. Bridges and R. B.Vaughn, "Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection", In Proceedings of the National Information Systems Security Conference (NISSC), Baltimore, MD, pp.16-19, October 2000.

[7] M.S.Abadeh, J.Habibi, "Computer Intrusion Detection Using an Iterative Fuzzy Rule Learning Approach", in Proceedings of the IEEE International Conference on Fuzzy Systems, pp: 1-6, London, 2007.

[8] B. Shanmugam, N. BashahIdris, "Improved Intrusion Detection System Using Fuzzy Logic for Detecting Anamoly and Misuse Type of Attacks", in Proceedings of the International Conference of Soft Computing and Pattern Recognition, pp: 212-217, 2009.

[9] J.T.Yao, S.L. Zhao, L.V.Saxton , "A study on Fuzzy Intrusion Detection", Proc. of Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, pp. 23-30, 2005.

[10] Q. Wang and V.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.

[11] V. Das, V. Pathak, S. Sharma, Sreevathsan, M.Srikanth, G. Kumar T, "Network Intrusion Detection System Based On Machine Learning Algorithms", International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, December 2010.

[12] A. Mewada, P. Gedam, S. Khan, M. U. Reddy,"Network Intrusion Detection Using Multiclass Support Vector Machine", Special Issue of IJCCT Vol.1 Issue 2, 3, 4; 2010 for International Conference [ACCTA-2010], August 2010.

[13] H. F. Eid, A.Darwish, A. E.Hassanien, and A.Abraham,"Principle Components Analysis and Support Vector Machinebased Intrusion Detection System", *10th International Conference on Intelligent Systems Design and Applications, 2010.*

[14] V.Jaiganesh, Dr. P. Sumathi, "Intrusion Detection Using Kernelized Support Vector Machine WithLevenbergmarquardt Learning", International Journal of Engineering Science and Technology (IJEST), Vol. 4 No.03 March 2012.

[15] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", in Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications, pp. 53-58, Ottawa, Ontario, Canada, 2009.

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

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

[18] 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.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. 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)

[20] V. N. Vapnik, **"**The Nature of Statistical Learning Theory**"**, Second Edition, Springer, New York, ISBN 0-387-98780-0, 1999.

[21] B.Scholkopf, A. J. Smola, "Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond", The MIT Press Cambridge, MassachusettsLondon, England, 2001.

[22] K. P. Bennett, C.Cambell, "Support Vector Machines: Hype or Hallelujah? ",SIGKDDExplorations, Volume 2, Issue 2 ,pp.1-13, 2000.

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

[24] F. KUANG, W. XU, S. ZHANG, Y. WANG, K. LIU , "A Novel Approach of KPCA and SVM for Intrusion Detection", Journal of Computational Information Systems,pp. 3237-3244, 2012.

[25] Shilpalakhina, S. 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.

[26] J.T.Yao, S.L. Zhao, L.V.Saxton , "A study on Fuzzy Intrusion Detection", Proc. of Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, pp. 23-30, 2005.

AUTHOR'S PROFILE

Md. Al Mehedi Hasan

is now working as an Assistant Professor of Computer Science and Engineering (CSE) Department, Rajshahi University of Engineering & Technology (RUET), Bangladesh. He got B.Sc. (Honours) and M.Sc degree in Computer Science and Engineering from University of Rajshahi, Bangladesh.He has more than twentypublications in international conferences and journals. His research interest is Artificial Intelligence, Pattern Recognition, Image Processing, Machine Learning, Computer Vision, Probabilistic and Statistical Inference, Operating System.
Emil: mehedi_ru@yahoo.com

Dr. Mohammed Nasser

is now Professor and Chairman at Department of Statistics, University of Rajshahi ,Rajshahi – 6205, Bangladesh. He got Honours and M.Sc degree in Statistics from Jahangirmagar University, Bangladesh and did his Ph.D. degree on "Continuity and Differentiability Of Statistical Functionals; Its Relation To Robustness In Boostrapping*"* at Research Centre for Mathematical and Physical Sciences, Chittagong Univeristy, Bangladesh. He was the founder chairman of Diagnostic Robust Resampling Statistics and Data Mining research group (now named as Statistical Learning Group) that has members working in five national-international universities and research organizations. He has already published more than forty articles in national and international journals in statistics, mathematics and sociology. His current research interest is in Mathematics of Kernel Methods, Bioinformatics, Robust Estimation and Globalization. He is a member of International Avisory Board of IFNA, a life member of both Bangladesh Statistical Society and Bangladesh Mathematical Society, and editorial board members of three national journals and one international journal (GJMS). Email:mnasser.ru@gmail.com

Biprodip Pal

is now working as a lecturer of Computer Science and Engineering(CSE)department of Rajshahi University of Engineering & Technology (RUET), Bangladesh. He got B.Sc. Engg.degree in CSE from Rajshahi University of Engineering & Technology (RUET), Bangladesh. He has publications in international conferences and journals. His research interest is Artificial Intelligence, Machine Learning, Probabilistic and Statistical inference, Data Mining,Network Security. Email:biprodip.cse@gmail.com