PERFORMANCE EVALUATION OF DIFFERENT KERNELS FOR SUPPORT VECTOR MACHINE USED IN INTRUSION DETECTION SYSTEM

Md. Al Mehedi Hasan[1], Shuxiang Xu[2], Mir Md. Jahangir Kabir[2] and Shamim Ahmad[1]

[1]Department of Computer Science and Engineering, University of Rajshahi, Bangladesh.
[2]School of Engineering and ICT, University of Tasmania, Australia.

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 numerous features. As a result, in order to get rid of this problem, several types of intrusion detection methods with different levels of accuracy have been proposed which leads the choice of an effective and robust method for IDS as a very important topic in information security. In this regard, the support vector machine (SVM) has been playing an important role to provide potential solutions for the IDS problem. However, the practicability of introducing SVM is affected by the difficulties in selecting appropriate kernel and its parameters. From this viewpoint, this paper presents the work to apply different kernels for SVM in ID Son the KDD'99 Dataset and NSL-KDD dataset as well as to find out which kernel is the best for SVM. The important deficiency in the KDD'99 data set is the huge number of redundant records as observed earlier. Therefore, we have derived a data set RRE-KDD by eliminating redundant record from KDD'99train and test dataset prior to apply different kernel for SVM. This RRE-KDD consists of both KDD99Train+ and KDD99 Test+ dataset for training and testing purposes, respectively. The way to derive RRE-KDD data set is different from that of NSL-KDD data set. The experimental results indicate that Laplace kernel can achieve higher detection rate and lower false positive rate with higher precision than other kernel son both RRE-KDD and NSL-KDD datasets. It is also found that the performances of other kernels are dependent on datasets.

KEYWORDS

Intrusion Detection, KDD'99, NSL-KDD, Support Vector Machine, Kernel, Kernel Selection

1. INTRODUCTION

In spite of having great advantages of Internet, still then it has compromised 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. The IDSs are simply classified as host-based or network-based. The former is operated 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. The anomaly detection systems find attacks by detecting changes in the pattern of utilization or behavior of the system [2].

As network attacks have been increased significantly 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 [6, 7], Fuzzy Logic [4, 8, 9], clustering algorithm [10] and more have been extensively employed to detect intrusion activities from large quantity of complex and dynamic data sets.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. The reason behind it is that different kernel functions construct different SVMs and affect the generalization ability and learning ability of SVM. However, 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 Radial Basis Function (RBF) kernel to build SVM based intrusion detection system [11, 12, 13, 14] and found the value of its parameter by using different technique and moreover some research paper did not mention value of the kernel parameter [13]and some others used the default value of the software package used [15].Surprisingly still there are many other kernel functions which are not yet applied in intrusion detection. But the nature of classification problem requires applying of different kernels for SVM to ensure optimal result [13]. This requirement motivated us to apply different kernel functions for SVM rather than just of using RBF in IDS, which, in turn, may provide better accuracy and detection rate. At the same time, we have also tried to find out parameter value to the corresponding kernel.

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

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 [16]. The KDD'99 dataset is a subset of the DARPA benchmark dataset prepared by Sal Stofo and Wenke Lee [17]. 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. The KDD99 data set includes three independent sets; ''whole KDD'', ''10%

KDD'', and ''corrected KDD''. Most of researchers used ''10% KDD'' and ''corrected KDD'' as training and testing set, respectively [18]. The training set contains a total of 22 training attack types. The ''corrected KDD'' testing set includes an additional 17 types of attacks and excludes 2 types (spy, warezclient) of attacks from training set. There are 37 attack types which are included in the testing set, as shown in Table 1 and Table 2. The simulated attacks fall in one of the four categories [1, 18, 19]: (a) Denial of Service Attack (DoS), (b) User to Root Attack (U2R), (c) Remote to Local Attack (R2L), (d) Probing Attack. 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 [20].

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 [13]. 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 [15]. 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 infrequent 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 derived a new data set RRE-KDD by eliminating redundant record from KDD'99 train and test dataset (10% KDD and corrected KDD), so the classifiers will not be biased towards more frequent records. This RRE-KDD dataset consists of KDD99Train+ and KDD99Test+ dataset for training and testing purposes, respectively. 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.

2.2. NSL-KDD DATASET

To overcome the problem of KDD'99 dataset, researchers have proposed a new data set, NSL-KDD, which consists of selected records of the complete KDD data set [15]. The development of NSL-KDD dataset was different than our approach. The NSL-KDD dataset also does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records. The numbers of records in the train and test sets are also reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

3. SVM CLASSIFICATION

The theory of support vector machine (SVM) is from statistics and the basic principle of SVM is finding the optimal linear hyper plane in the feature space that maximally separates the two target classes [21, 22, 23]. 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.

3.1. 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 hyper plane $w^T x + b = 0$. Finding a separating hyperplane can be posed as a constraint satisfaction problem. The constraint problem can be defined to determine *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:

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

$$\text{Subject to: } y_i(w^T x_i + b) \geq 1, \; i = 1,2,3, \dots\dots, n \qquad (1)$$

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:

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

$$\text{Subject to: } y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1,2,3,\dots\dots,n \qquad (2)$$
$$\xi_i \geq 0, i = 1,2,3,\dots\dots,n$$

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 [21, 22, 23]. Using the method of Lagrange multipliers, we can obtain the dual formulation which is expressed in terms of variables α_i [22, 23]:

$$maximize_\alpha \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i\, y_j x_i^T x_j$$
$$\text{Subject to: } \sum_{i=1}^{n} y_i\alpha_i = 0, 0 < \alpha_i < C \text{ for all } i = 1,2,3,\dots\dots,n \qquad (3)$$

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

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

3.2. 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 $\emptyset: X \to F$. 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

$$maximize_\alpha \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i\, y_j\emptyset(x_i)^T\emptyset(x_j)$$
$$\text{Subject to: } \sum_{i=1}^{n} y_i\alpha_i = 0, 0 < \alpha_i < C \text{ for all } i = 1,2,3,\dots\dots,n \qquad (5)$$

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 and x_j, the inner product of the mapped points can be evaluated using the kernel function without ever explicitly knowing the mapping [24] . The kernel function can be defined as

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

Substituting the kernel in the 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 $i = 1,2,3, \ldots \ldots, n$ (6)

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

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

3.3. KERNEL AND ITS PARAMETERS SELECTION

A kernel function and its parameter have to be chosen to build a SVM classifier [14]. In this work, four 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: $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma}\right)$, σ is the width of the function.
4. Laplace Kernel: $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{\sigma}\right)$, σ is the width of the function.

Training an SVM finds the large margin hyper plane, i.e. sets the parameters α_i. 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)[25]. 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* allows to ignore points close to the boundary, and increases the margin.

Kernel parameters also have a significant effect on the decision boundary [25]. 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 or Laplace 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 [25].

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 [21, 25].

3.4. 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, 21]. 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 hyper plane discriminating the class j and the combined class as

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

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

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

In this approach the index of the largest component of the discriminant vector $(g^1(x), g^2(x), \ldots\ldots, g^k(x))$ 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.

4. DATASETS AND EXPERIMENTS

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, 26, 27,28]. 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 RRE-KDD (KDD99Train+ and KDD99Test+) intrusion detection datasets in terms of number of samples is

given in Table 3. The distribution of the number of samples of each class of NSL-KDD dataset is also given in Table 3. Although the distribution of the number of samples of attack is different on different research papers, we have used the Table 1 and 2 to find out the distribution of attack [1, 3,18].In our experiment, whole train (KDD99Train+ and KDD Train+NSL-KDD) dataset has been used to train our classifier and the test (KDD99Test+ and KDD Test+NSL-KDD) 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 both of the training set (KDD99Train+ and KDD Train+NSL-KDD) to tune the parameters of all kernels and another 10% samples from the training set (KDD99Train+ and KDD Train+NSL-KDD) to validate those parameters, as shown in Table 3. In our experiment, four different types of kernel have been used.

Table 3: Number of Samples of Each Attack in Dataset

Dataset	**Various Independent Sets**	**Normal**	**DoS**	**Probing**	**R2L**	**U2R**	**Total**
Original KDD'99 Dataset	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
RRE-KDD Dataset	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
NSL-KDD Dataset	KDDTrain+NSL-KDD	67343	45927	11656	995	52	125973
	KDDTest+NSL-KDD	9711	7458	2421	2887	67	22544
	Train SetNSLKDD (For Model Selection)	6735	4593	1166	100	6	12600
	Validation Set NSLKDD (For Model Selection)	6735	4593	1166	100	6	12600

5. PRE-PROCESSING

SVM classification system is not able to process the train (KDD99Train+ and KDDTrain+NSL-KDD) and test (KDD99Test+ and KDDTest+NSL-KDD) 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 or NSL-KDD 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

(RRE-KDD Dataset) or 70 different network services (NSL-KDD dataset) 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 RRE-KDD dataset and 119 variables for each samples of the NSL-KDD 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).

6. 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. In our experiments, this method selects the values of parameters considering highest accuracy and then time if more than one position in search space has the same accuracy. In our experiment, Sequential Minimization Optimization with the following options in Matlab, shown in Table 4, 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 3. We have tuned SVM separately for each of the dataset (RRE-KDD and NSL-KDD dataset) using their corresponding train and validation set which has been derived for model selection purposes. In this section, we will give the procedure for SVM model selection for our derived RRE-KDD dataset in details with the obtained result in graphical presentation, however, result found for NSL-KDD dataset presented in tabular form.

Table 4: Sequential Minimization Optimization Options

Option	Value
Max Iter	1000000
Kernel Cache Limit	10000

6.1. MODEL SELECTION FOR RRE-KDD DATASET

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 1. We have taken the parameter value C=4 which provides highest accuracy 99.31% accuracy in the validation set 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 2. We have taken the parameters value d=2 and C=0.0039 which provides highest accuracy 99.70% accuracy in the validation set to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

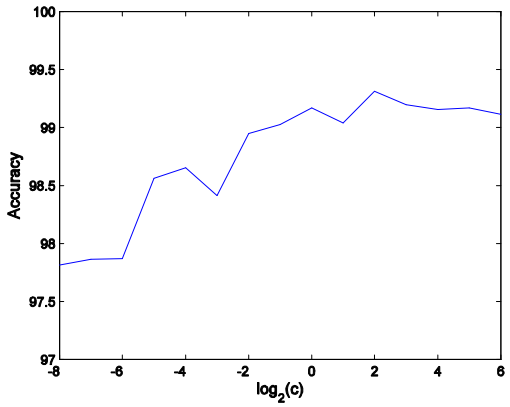


Figure 1: Parameter (C) tuning for Linear Kernel

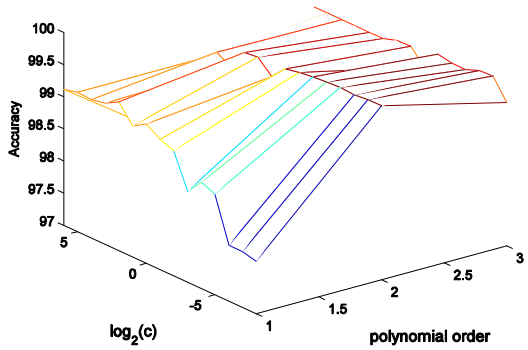


Figure 2: Parameters (C, d) tuning for Polynomial Kernel

For radial basis kernel, to find the parameter value C (penalty term for soft margin) and σ, 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 3. We have taken parameter value C=32 and σ=16 which provides highest accuracy 99.01% among the search space in the validation set to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

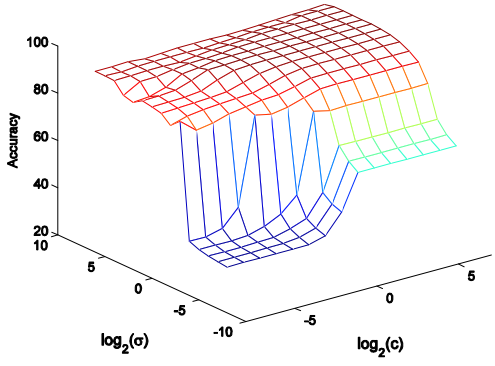


Figure 3: Parameter (C, σ) tuning for Radial Basis kernel

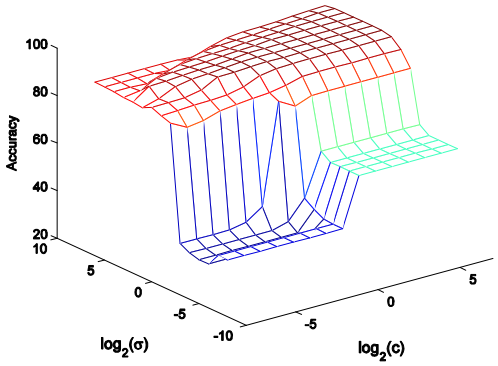


Figure 4: Parameters (C, σ) tuning for Laplace kernel

Again, for Laplace kernel, to find the parameter value C (penalty term for soft margin) and σ, 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 Laplace kernel is shown in Figure 4. We have taken parameter value C=64 and σ=8 which provides highest accuracy 99.70% accuracy in the validation set to train the whole train data (KDD99Train+) and test the test data (KDD99Test+).

6.2. MODEL SELECTION FOR NSL-KDD DATASET

We have followed the same procedure discussed in section 6.1 to select the model for SVM for NSL-KDD dataset. The optimal parameter value which we took for each of the kernel is shown in Table 5. This parameter value has been used to train the whole train data (KDDTrain+NSL-KDD) and test the test data (KDDTest+NSL-KDD).

Table 5: Optimal Parameter Value for Each of the Kernel for NSL-KDD Dataset

Kernel	**C**	**d**	**σ**
Linear	16	-	-
Polynomial	0.0625	2	-
Radial Basis	32	-	2
Laplace	32	-	4

7. SIMULATION RESULTS

The final training/testing phase is concerned with the development and evaluation on a test set of the final SVM model created on the basis of optimal hyper-parameters found in the model selection phase [21]. After finding the parameters, we have built the model using the whole training dataset (KDD99Train+ and KDDTrain+NSL-KDD) for each of the kernel tricks. Finally we have tested the model using the test dataset (KDD99Test+ and KDDTest+NSL-KDD). The training and testing results are given in Table 6 according to the classification accuracy. From Table 6, it is observed that the Laplace kernel produces higher detection rate on both RRE-KDD dataset and NSL-KDD datasets than other kernels. It is also noticed that the linear and polynomial kernel performs better than RBF kernel on NSL-KDD dataset, whereas, RBF kernel performs better than linear and polynomial kernel on RRE-KDD dataset.

Table 6: Training and Testing Accuracy of Different Kernels on RRE-KDD and NSL-KDD Datasets

Kernel	**Training Accuracy**		**Testing Accuracy**	
	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset
Linear	77.82	86.56	36.93	63.60
Polynomial	99.73	99.31	91.27	73. 54
Radial Basis	99.79	99.80	92.99	56.88
Laplace	99.97	99.91	93.19	79.08

Table 7: False Positive Rate (%) of Different Kernels for Each of the Attack Types Including Normal.

Kernel	Dos		Normal		Probing		R2L		U2R		Average False Positive Rate	
	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset
Linear	8.04	58.5	91.55	3.99	12.14	21.11	84.7	99.92	94.29	98.87	58.14	56.48
Polynomial	9.55	27.46	1.52	5.34	43.09	37.42	83.29	81.58	92.86	97.37	46.06	49.83
Radial Basis	3.84	70.53	1.94	3.25	42.64	51.63	77.76	97.6	85.71	100	42.38	64.60
Laplace	3.62	12.2	1.27	4.45	40.44	27.51	87.38	86.53	67.14	97	39.97	45.74

The obtained false positive and precision rates for each of kernel are given in Table 7 and 8 respectively. The Laplace kernel gives lower false positive rate and higher precision than other kernels for both RRE-KDD dataset and NSL-KDD dataset. It is noted that Table7 and 8 provides detail performance of different kernels on different datasets as summarized in Table 6.

Table 8: Precision (%) of Different Kernels for Each of the Attack Types Including Normal.

Kernel	Dos		Normal		Probing		R2L		U2R		Average Precision	
	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset	RRE-KDD Dataset	NSL-KDD Dataset
Linear	36.12	90.31	100	67.31	19.63	36.7	44.07	9.52	2.12	15.38	40.39	43.84
Polynomial	97.64	89.69	90.14	70.08	63	52.44	83.49	94.29	6.67	41.18	68.19	69.54
Radial Basis	96.32	99.55	92.99	53.91	63.81	41.25	85.64	93.55	34.48	0	74.65	57.65
Laplace	96.8	94.03	92.29	70.92	74.74	77.04	94.14	96.74	85.18	88.89	88.63	85.52

8. CONCLUSION

In this research work, we evaluated the performance of different kernels for SVM used in IDS. The performances of the different kernels based approach has been observed on the basis of their accuracy in terms of false positive rate and precision. The results indicate that the performance of the SVM classification depends mainly on the types of kernels and their parameters. The obtained results justify the motivation of this work that only a single kernel cannot be considered for SVM used in IDS to achieve the optimal performance. Research in intrusion detection using SVM approach is still demanding due to its better performance. The research community working on SVM based classification will be benefited with the results of this study.

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] Adetunmbi A. Olusola., Adeola S. Oladele., Daramola 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] Hesham Altwaijry, Saeed Algarny, "Bayesian based intrusion detection system", Journal of King Saud University – Computer and Information Sciences, pp.1–6, 2012.

[4] 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.

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

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

[7] 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, pp.150-154, 22-24 Dec. 2012.

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

[9] Bharanidharan Shanmugam, Norbik BashahIdris, "Improved Intrusion Detection System Using Fuzzy Logic for Detecting Anomaly and Misuse Type of Attacks", Proceedings of the International Conference of Soft Computing and Pattern Recognition, pp: 212-217, 2009.

[10] 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.

[11] Vipin Das, Vijaya Pathak, Sattvik Sharma, Sreevathsan, MVVNS. Srikanth, Gireesh 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] Arvind Mewada, Prafful Gedam, Shamaila Khan, M. Udayapal 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] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, Ajith Abraham, "Principle Components Analysis and Support Vector Machine based 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 With Levenbergmarquardt Learning", International Journal of Engineering Science and Technology (IJEST), Vol. 4 No. 03 March 2012.

[15] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, 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), 2009.

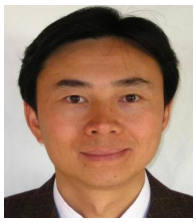
[16] MIT Lincoln Laboratory, DARPA Intrusion Detection Evaluation, http://www.ll.mit.edu/CST.html, MA, USA. July, 2010.

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

[18] M. Bahrololum, E. Salahi, 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] Nadiammai, G. V., M. Hemalatha. "Effective approach toward Intrusion Detection System using data mining techniques." Egyptian Informatics Journal 15(1), pp. 37-50, 2014.
[20] Aggarwal, Preeti, and Sudhir Kumar Sharma. "Analysis of KDD Dataset Attributes-Class wise for Intrusion Detection." Procedia Computer Science, 57, pp. 842-851, 2015.
[21] Md. Al Mehedi Hasan, Mohammed Nasser, Biprodip Pal, Shamim Ahmad,"Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS)" Journal of Intelligent Learning Systems and Applications, Vol.6 No.1, PP. 45-52, February 2014.
[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] Kristin P. Bennett, Colin Cambell, "Support Vector Machines: Hype or Hallelujah?",SIGKDDExplorations, Volume 2, Issue 2, pp.1-13, 2000.
[25] A. Ben-Hur, J. Weston. "A User's guide to Support Vector Machines", In Biological Data Mining. OlivieroCarugo and Frank Eisenhaber (eds.) Springer Protocols, 2009.
[26] 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.
[27] Shilpalakhina, Sini Joseph, 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.
[28] Vasan, K. Keerthi, B. Surendiran., "Dimensionality reduction using Principal Component Analysis for network intrusion detection." Perspectives in Science, 2016.

AUTHORS

Md. Al Mehedi Hasan is currently a PhD Fellow in the Department of Computer Science and Engineering (CSE) of Rajshahi University (RU), Bangladesh. He got B.Sc. (Hons) and M.Sc degree in Computer Science and Engineering from Rajshahi University, Bangladesh. After working as a lecturer (from 2007), he is an assistant professor (from 2010) in the Dept. of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Bangladesh. His interested areas of research are Artificial Intelligence, Pattern Recognition, Image Processing, Machine Learning, Computer Vision, Probabilistic and Statistical Inference, Operating System, Bioinformatics, and Computational Biology.



Shuxiang Xu is currently a lecturer of School of Engineering and ICT, University of Tasmania, Australia. He received a Bachelor of Applied Mathematics from University of Electronic Science and Technology of China (1986), China, a Master of Applied Mathematics from Sichuan Normal University (1989), China, and a PhD in Computing from University of Western Sydney (2000), Australia. He received an Overseas Postgraduate Research Award from the Australian government in 1996 to research his Computing PhD. His current interests include the theory and applications of Artificial Neural Networks, Genetic Algorithms, and Data Mining.

Mir Md Jahangir Kabir is currently an Assistant Professor of the Department of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Bangladesh. He received B.Sc. in Computer Science and Engineering from Rajshahi University of Engineering and Technology, Bangladesh (2004), a M.Sc. in Information Technology from University of Stuttgart, Germany (2009) and a P.hD. in University of Tasmania, Australia (2016). After working as a lecturer (from 2004), he is an assistant professor (from 2010) in the Dept. of Computer Science and Engineering, Rajshahi University of Engineering and Technology, Bangladesh. He received an Overseas Postgraduate Research Award from the Australian government in 2013 to research in PhD. His research interests include the theory and applications of Data Mining, Genetic Algorithm, Machine Learning and Artificial Intelligence.

Dr. Shamim Ahmad: Received his Doctor of Engineering in Electrical Engineering from Chubu university, Japan. He got his B.Sc (Hons) and MSc degree in Applied Physics and Electronic Engineering from Rajshahi University, Bangladesh. Following that he worked as research student in the department of Computer Engineering, Inha University, South Korea. Currently he is working as Professor in the department of Computer Engineering of Rajshahi University. He was the former head of that department. His interested areas of research are Embedded System, Data Miming and Digital Image Processing.