ISSN 2347 - 3983
Volume 8. No. 10, October 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter1388102020.pdf
https://doi.org/10.30534/ijeter/2020/1388102020

Performance of Various SVM Kernels for Intrusion Detection of Cloud Environment

N. Nirmalajyothi[1], K. Gangadhara Rao[2], B. Basaveswara Rao[3], K. Swathi[4]
[1] Dept. of CSE, Acharya Nagarjuna University,India,nirmala.narisetty@gmail.com
[2] Dept. of CSE, Acharya Nagarjuna University, India,,kancherla123@gmail.com
[3]Computer Center, Acharya Nagarjuna University, India, bobbabrao@yahoo.co.in
[4]Dept.of CSE, NRI Institute of Technology, India, swathipvpsit@gmail.com

ABSTRACT

This paper investigates the performance among the various kernel based SVM classifiers for intrusion detection in cloud environment. Several researchers have presented the different kernel functions of SVM for Intrusion Detection. There is always an ambiguity in choosing which kernel function is to apply for better detection rate to identify classification accuracy factor. This paper explores to achieve this objective to identify the popular kernel functions linear, polynomial, radial basis function and Sigmoid. The CIDDS-001 dataset is adapted because of it is a recently available benchmark dataset and generated with new types of attacks of cloud environment. To evaluate the performance of different kernel functions computational time and accuracy taken as QoS metrics with ten-fold cross validation. The numerical results are calculated and conclusions are drawn.

Key words: Classification, Intrusion Detection System (IDS), Support Vector Machine, Kernel, Cloud Computing.

1. INTRODUCTION

Cloud computing is one of the burgeoning and contemporary technology which plays a vital role in IT industry. It is Internet based distributed computing model where virtual shared servers provide computing resources with different deployment models catering to the needs of varied types of customers and also several popular delivery models where majority of them work on pay-as you-use basis[1] . Due to cloud computing technological revolution, the users can utilize scalable resources without any huge investments on physical infrastructure as well as software procurements [2].

Since cloud uses the Internet to deliver the services, it has become highly vulnerable to the various types of attacks and therefore security remains a major problem that haunts the community of users [24]. Inorder to increase the resources utilization efficiently in better way and the tremendous rise in cyber attacks has caused the cloud network traffic to be distinguished as legitimate and malicious traffic. Network traffic analysis is therefore necessary for cloud-based Intrusion detection (ID) to monitor the cloud service provider's overall performance and to to prevent violations of the Service Level Agreement (SLA) [3].

One of the major threats faced by the cloud platform is DDoS attack like any other predecessor technologies had experienced. It is a special type of DoS attack, where malicious users generates volume of network traffic needed to exhaust processing and connectivity resources which reduces the availability of resources to legitimate users[4]. The victims are surprisingly government agencies, military departments, trade organizations and also some popular websites like Facebook, GitHub, and Amazon who have experienced interruption in normal operations leading to financial loss, service interruption and also lack of availability [5].

Distributed Denial of Service (DDoS) attacks can affect availability of the cloud services. Therefore, this area has been chosen to be the research focus. By studying the nature of DDoS attacks and cloud, it has been found that it is difficult for attackers to succeed in affecting the cloud service due to the huge resources that the cloud has in its data centers, which are distributed globally. However, there is another way that adversaries can use to affect the cloud by carrying out traditional DDoS attacks against cloud customers. This point is explained by Christopher Hoff in 2008, and he named it Economic Denial of Sustainability (EDoS). It is the phenomenon that exploits the elasticity and scalability of the cloud to increase the amount of payments and therefore hit the cloud payment model (pay-as-you-use) by generating DDoS attacks against customers networks by sending a huge number of fake requests, leading customers to ask the provider, according to Service Level Agreement (SLA), to allocate them more resources. The result of such a technique will be high bills for customers, forcing them to withdraw from cloud services.

Recently many researchers and scholars have done some significant work to detect and mitigate the DDoS attacks with statistical, OR and Machine Learning (ML) techniques through analytical, simulation and experimental studies. In compared to statistical methods, ML tools are apt and feasible to learning patterns with no previous knowledge of what those patterns may be. ML is a science of computer algorithms that improve automatically through experience without being explicitly programmed for a selected task. It gives computers ability to learn from input data called training data set and builds a prediction model for test set called test data. The larger the data, the more accurate are going to be the results of the study and helps to detect malicious activity faster and successively can stop the attacks before they get initiated [7].

Therefore Network traffic classification is an essential step for Intrusion Detection (ID) in cloud Environment to utilize cloud resources proficiently. In general classification is the procedure of grouping similar entities with common features and then identifying to which of the categories a test sample belongs based on the training data containing whose categories are known.

A classification based IDS tries to classify all traffic as either normal or malicious. The major challenge in classification is to minimize the false positive rate (rate of normal traffic predicted as attacks), false negative rate (rate of malicious traffic predicted as normal), Mean Square Error (MSE) [6].

Classification can be accomplished in supervised learning or unsupervised learning. In supervised learning, label is associated with each data sample. It is supposed to be the answer to a question about the sample. If the label is categorical, then the task is referred to as classification else it’s termed as regression. In unsupervised learning one typically tries to discover hidden regularities or to detect anomalies with the unlabeled data based on similarities and differences [7].

Support Vector Machine (SVM) is one of the widely used Machine Learning algorithms for data analysis and pattern recognition classification. One of the applications of Support vector machine (SVM) in cloud environment is classification of network traffic efficiently due to its better generalization capabilities [9]. It can detect novel attacks and provides a standard mechanism to fit the surface of the hyper plane to the data by utilizing the kernel function to automatically avoid over-fit to the data and performs well in comparison with other classifiers [9]. For instance, finding how many neurons a task may require is another issue which determines whether optimality of that Neural Network is reached [10]. The complexity of classification does not rely on the dimensionality of the feature space, so they can potentially learn a larger set of patterns and can therefore scale better than neural networks [8].

Compared to artificial neural networks existing, it has relatively fast processing and good recognitio n performance, as shown in [8]. Feature selection or dimensionality reduction can help reduce the SVM classification time and saving memory space effectively [16].

The objective of this paper is to explore the performance of SVM classifier using different kernel models. For this purpose linear and non-linear kernel models are considered. Among the non-linear models the three kernel models polynomial, RBF and sigmoid are identified for performance evaluation. To evaluate the performance metrics accuracy and computational time are chosen. Further conclusions are drawn based on these metrics and suggest which model is suitable for mitigation

2. LITERATURE REVIEW

This section introduces contributions made by different authors in the areas of machine learning and how it is used in the context of intrusion detection in cloud environments

In [11], authors have successfully used k-nearest neighbour classification and k-means clustering algorithms on CIDDS-001 dataset to measure the complexity in terms of prominent metrics. They have successfully proved based on the results of evaluation that the chosen dataset is suitable for assessing intrusion detection based on anomalies.

Mohamed Idhammada [12] proposed IDS to capture the incoming network traffic to edge network routers of the physical layer which is an integral part of the Cloud setup. The network traffic is preprocessed and passed to machine learning classifiers such as Naïve Bayes and Random Forest to detect attacks in cloud. The system was evaluated using CIDDS-001 dataset and results were found to be satisfactory.

Same writers “suggested detection system of DDoS attacks in a cloud environment based on information theoretical entropy and random forest classifier. Time-based sliding window algorithm is employed to estimate the entropy of network header characteristics of incoming traffic. When estimated entropy exceeds its normal range then incoming traffic is preprocessed and then random forest classifier is applied. The significant improvement of the accuracy of 2.5% is noticed here compared to the accuracy of Random forest tested directly on the CIDDS-001 which is 97%” in [13].

In [14] paper “combination of k-cross validation and Grid Search method is used to look for optimal parameters for SVM, and compare the classification accuracy of various kernel function on two well-logging dataset. The experiment outcome shown that the type of kernel function affects classification rate most and Polynomial performs best”.

Raneel kumar, Lal and Sharma proposed [15] an Intrusion Detection system (IDS) to detect DoS attacks emanating from one or more Virtual machines to another in cloud environment which has got multiple VM's as multi-tenanted set up. The Intrusion Detection system composed of a packet sniffer, a function extractor, and one – class Support Vector Machine classifier. The proposed Intrusion Detection System showed promising results to detect seven different types of DoS attacks.

In 2009, Chunhua Gu and Xueqin Zhang,[16] proposed a system for classification of intrusion using rough set for reducing attributes and support vector machine. Again in the same year, Yong-Xiang et al. [17] "proposed Classification an intrusion detection using incremental SVM based on key feature selection".

3. BASICS

The original optimal hyper plane algorithm was proposed by Vapnik in 1963 was for a linear separable case. Consider the dataset containing a training samples $(x_1,y_1),(x_2,y_2),\ldots\ldots,(x_n,y_n)$ where $x_i \in R^n$, y_i is known as class labels, y_i is -1 or +1.These two labels can be applied to intrusion detection with +1 label representing normal and -1 label for representing malicious. However there may be many hyper planes that separate the data. The goal of SVM is to fine optimal hyper plane which separates two classes with maximum margin.

The classification line is defined as

$$f(x) = wx + b \qquad (1)$$

w=vector weight that are perpendicular to the hyper plane (Normal plane)
b=position of the field relative to the coordinate center

Then the decision function constraint solving is given by

$$wx + b = 0 \qquad (2)$$

The optimization problem of SVM can be summarized as:

$$\text{Minimize} \quad \frac{1}{2}||W||^2 \qquad (3)$$

By above equation the data points should satisfy the following equations in R^n such that

$$y_i(w^T x_i + b) \geq 1, \text{ for } i=1, 2, \ldots, n \qquad (4)$$

3.1 Kernel Types

There are four popular types of basic kernel functions which are: linear, polynomial, radial basic function (RBF), and sigmoid.

A. Linear Kernel function:

$$K(x, x_i) = x.x_i \qquad (5)$$

Linear kernel function is most frequently used to map information to a higher dimensional space when the numbers of features are more. It is faster in training than with another kernel for solving the optimization problems.

B. Non-Linear Kernels

Polynomial Kernel function:

$$K(x, x_i) = [(x.x_i) + 1]^d \qquad (6)$$

The Polynomial kernel is a dynamic kernel. Polynomial kernels are well appropriate for problems when training data is normalized. The parameter d is degree of kernel function. As d grows then dimensionality of mapping function grows and computational complexity grows, but it would be easier to classify the sample.

Radial Basis Function (RBF):

$K(x, x_i) = \exp(- * ||x - x_i||^2/2\sigma^2)$, σ is width of the function (7)

The Gaussian kernel also known as radial basis function. It is a widely used kernel function in SVM classification for learning. RBF has excellent performance on local points. In (7) $||x - x_i||^2$ is the Euclidian square distance between the two feature vectors.

Sigmoid Kernel function:

$$K(x, x_i) = \tanh(\gamma * (x.x_i) + c), \ \gamma > 0 \qquad (8)$$

The Hyperbolic Tangent Kernel is also known as the Sigmoid Kernel and as the Multilayer Perceptron (MLP) kernel. It is widely used in neural network field as an activation function for artificial neurons.

4. DATA SET DESCRIPTION

CIDDS-001(Coburg Intrusion Detection Dataset) [18] is a labeled unidirectional flow based dataset generated by emulating small business environment in cloud for the evaluation of Network Intrusion Detection System (NIDS). It consists of real traffic data from an internal server with open stack environment (Web, E-Mail servers etc.) and external server (file synchronization, web server). Python scripts emulate normal user behavior on the clients.

The numbers of attributes in dataset are 14, the first attributes 1 to 11 are default NetFlow attributes whereas the attributes 12 to 14 are additional attributes described the attacks. Table 1 provides the description of CIDDS-001 dataset attributes [20].

Table 1: Features and their Description of CIDDS-001 dataset

SL. No.	Feature Name	Feature Description
1	Date first seen	flow first seen at particular Start time
2	Duration	Duration of the flow
3	Proto_type	Transport Protocol (e.g. ICMP, TCP, or UDP)
4	Src_IP_Addr	IP Address of Source
5	Src_Pt	Source Port number
6	Dst_IP_Addr	IP Address of Destination
7	Dst_Pt	Destination Port number
8	Packets	Number of packets transmitted
9	Bytes	Number of bytes transmitted
10	Flows	
11	Flags	OR concatenation of all TCP Flags
12	Tos	Type of Service
13	Class	Classifying label (Normal, Attacker, Victim, Suspicious and Unknown)
14	Attack Type	Attacks Category (Port Scan, DoS, Brute force, Ping Scan)
15	Attacked	Unique Attack id. Allows attacks which belong to the same class carry the same attack id
16	Attack Description	It provides added information the set attack parameters Provided (e.g. the number of attempted password guesses for SSH-Brute-Force attacks)

5. PREPROCESSING

The collected raw data need to be preprocessed before it is used for learning to enable algorithms operates fast and work accurately. The data preprocessing stage consists of 3 steps transformation, normalization and sampling.

A. Transformation:

In this step categorical features of CIDDS-001 dataset are transformed into continuous features. The features Proto_type (3), Src_IP_Addr (4), Dst_IP_Addr (6), Flags (11), Class Label (13) are categorical features in dataset and they are converted into numeric. Each categorical feature consists of numeric values in particular range that is the number of the categorical values in that feature after transformation, for example, the field "Class" with data normal, attacker, victim and suspicious will have only integer values of 0, 1, 2, 3 correspondingly[19].

In case of Source IP address and Destination IP address transformation, First three bytes of IP addresses are replaced with some label and fourth byte is just appended to it. So that all IP addresses in the same network will have common label to preserve the information about network structure. Similarly other categorical features are transformed [20].

As per the original CIDDS-001 dataset, three sample records are shown in Figure 1 and Figure 2 shows the transformation of the categorical values into nominal values

1) 01:17.7, 0, UDP, 192.168.220.16, 35549, DNS, 5, 1, 73, 1, ..., 0, normal
2) 01:22.4, 0.021, TCP, 192.168.220.15, 37039, EXT_SERVER, 8082, 2, 338, 1, .AP..., 0, normal
3) 42:09.3, 0.433, IGMP, 192.168.200.9, 0, 10008_22, 0, 2, 108, 1, ..., 0, normal

Figure 1: Original three sample records from CIDDS-001 dataset

1) 77007,0,1,220016,35549,100,53,1,73,1,3,0,0
2) 82004, 0.021, 0, 220015, 37039, 200, 8082, 2, 338, 1, 0, 0, 0
3) 2529003, 0.433,

Figure 2: Resulting three sample records after transformation

B. Normalization

Within a feature there may be a large difference between the minimum and maximum values, e.g. values for feature packet is 1 and 208768 correspondingly which may lead to increased dispersion error. By nature CIDDS-001 dataset features describe various characteristics of the data and the values with distinct ranges are quantitative. The advantage of normalization is to evade numerical difficulties during the computation. "Because kernel values normally depend upon the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel, large attribute values might cause numerical problems" [21].Therefore in addition to transformation now the features are need to be normalized to reduce these difficulties by scaling them so that they fall within a particular range [0,1] [22]. In this paper min-max normalization technique is applied for normalization.

$$X_i^{new} = \frac{x_i - \min(X)}{\max(X) \quad \min(X_i)} \tag{9}$$

Where X is a feature of the network traffic data to be normalized, xi is the current value of the feature, min and max are the minimum and maximum values of overall values of feature, and X_i^{new} is the normalized value. Figure 3 shows CIDDS-001 data samples after applying normalization

1) 77007, 0, 1, 220016, 35549, 100, 53, 0,0.0000000592, 1, 3, 0, 0
2) 82004, 0.000276, 0, 220015, 37039, 200, 8082, 0.00000479,0.000000566, 1, 0, 0, 0
3) 2529003, 0.005686, 2, 44009, 0, 1000822, 0,

Figure 3: Resulting three sample records after normalization

C. Stratified Sampling

Sampling is a statistical procedure of selecting smaller set of data from large population. Among the existing sampling methods stratified sampling is most commonly used by researchers which divides dataset into different subgroups and selects instance from each subgroup in a proportionate manner.

For this experimental study stratified sampling is applied on input dataset using ten-fold to draw 10% of instances which are 104867 that is one fold for training and another fold for testing the model.

6. METHODOLOGY

As shown in Figure 4, the proposed methodology consists two phases, they are i) preprocessing ii) classification using SVM. Preprocessing is done as explained in section 5 then classification model for intrusion detection is constructed using SVM kernels to classify cloud network traffic. The model is used to classify test data. Finally the results of various kernel based SVM methods were compared to evaluate the performance of the proposed approach.

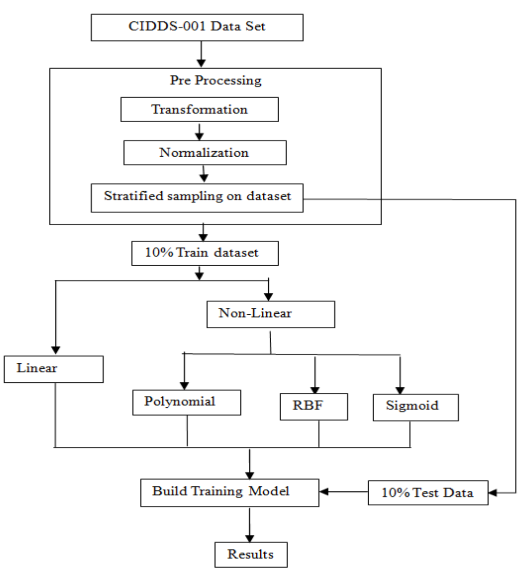

Figure 4: Methodology of Proposed Model

7. EXPERIMENTAL SETUP

In this paper data mining tool WEKA is used to perform experiments for Intrusion Detection [23] and applied ML kernel based support vector classification methods using LIBSVM to build classification model. Randomly selected set of 104867 points of the total data set (1048566) is used for testing various kernels with ten-fold cross validation. All experiments were performed using Intel core i5 with 1.80 GHz processor with 8GB RAM, running on windows 10.The statistical indices computational time and accuracy are used to analyze the performance of the SVM kernel based classifiers.

8. RESULTS & DISCUSSION

Support Vector Machine is one of the best learning algorithms [25]. The evaluation of SVM classifier was performed by a ten-fold cross validation for CIDDS-001 dataset inorder to avoid overfitting. Inorder to validate the performance of proposed model the results are compared with ten- fold cross validation with re-evaluation using supplied test set. The results are presented in below Table 2

Table 2: Computational time (seconds) for cross validation and reevaluation of various SVM kernels

	Various kernel Types			
	RBF	**Polynomial**	**Sigmoid**	**Linear**
Training Time	378.5	124.64	1.9	350.37
Testing Time	328.24	113.09	1.64	350.37

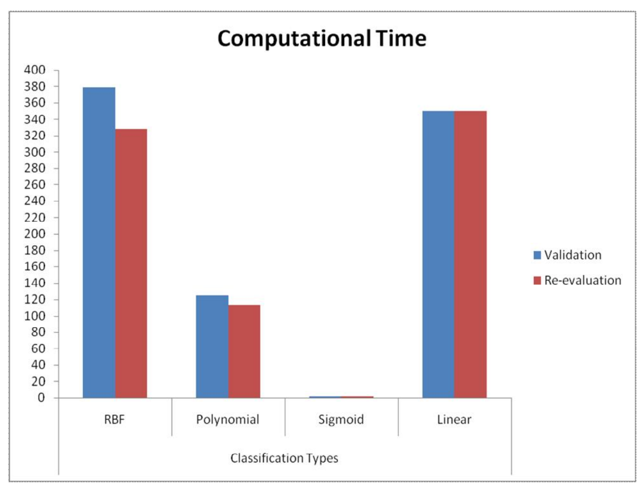

Figure 5: Classification Time for cross validation and re-evaluation of various SVM kernels

Figure 5 shows classification time of SVM kernels. Linear and RBF kernels are inline with each other. Polynomial takes reasonably less computational time where as sigmoid takes far less computational time compare to other kernels.

Table 3: Accuracy for cross validation and re-evaluation of various SVM kernels

	Various kernel Types			
	RBF	**Polynomial**	**Sigmoid**	**Linear**
Valid ation	88.57 %	49.41%	88.20%	72.24%
Re-e valua tion	88.81 %	52.58%	88.20%	87.89%

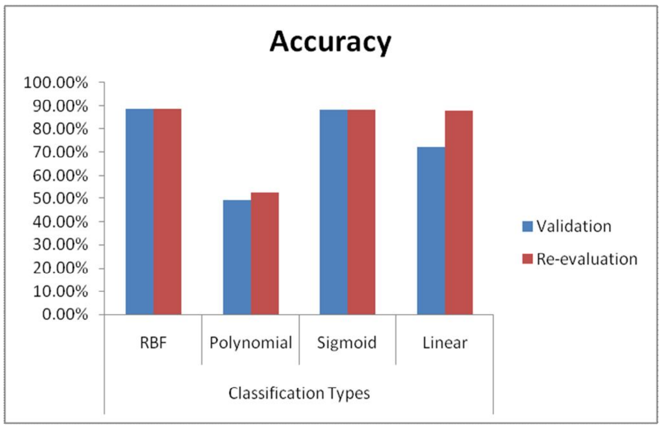

Figure 6: Accuracy for cross validation and re-evaluation of various SVM kernels

Table 3 and Figure 6 shows accuracy of SVM kernels, RBF and sigmoid exhibit more or less same level of accuracy. Linear is slightly on lower side where as polynomial struggles with 50% accuracy.

Table 4: Precision (%) of each kernel with training and test data

	Various kernel Types			
	RBF	**Polynomial**	**Sigmoid**	**Linear**
Validation	89.9%	82.2%	77.8%	82.8%
Re-evaluation	90.1%	82.0%	77.8%	83.7%

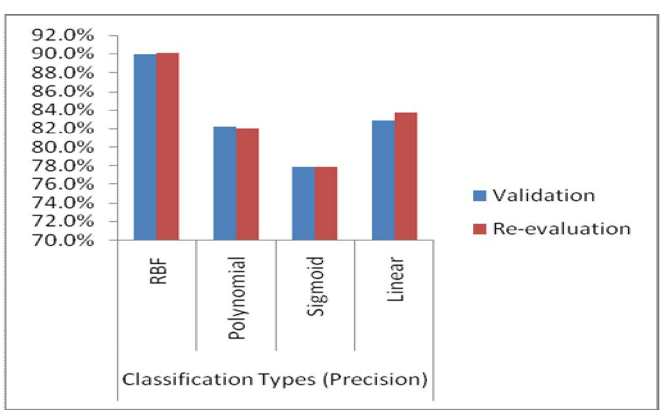

Figure 7: Precision (%) of cross validation and re-evaluation of various SVM kernels

Table 4 and Figure 7 shows precision of SVM kernels, in case of precision RBF kernel is with high degree of precision both in validation and re-evaluation. Linear is slightly lacks behind followed by polynomial and sigmoid with considerable gap.

Table 5- Recall (%) of each kernel with training and test data

	Various kernel Types -Recall			
	RBF	**Polynomial**	**Sigmoid**	**Linea r**
Validation	88.6%	49.4%	88.2%	72.2%
Re-evaluation	88.8%	52.6%	88.2%	87.9%

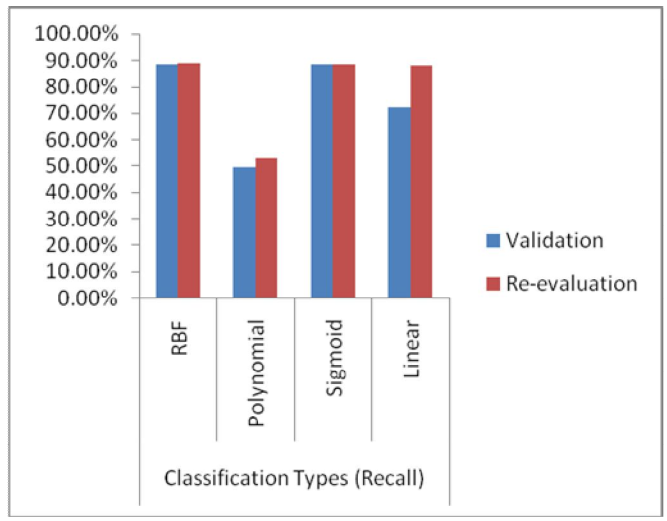

Figure 8: Recall (%) of cross validation and re-evaluation of various SVM kernels

Table 5 and Figure 8 shows recall of SVM kernels, RBF kernel exhibits good percentage of recall followed by sigmoid. Linear has moderate percentage of recall where as polynomial is only with 50%.

Table 6: Average F-Measure (%) of each kernel with training and test data

	Various kernel Types - F-measure			
	RBF	**Polynomial**	**Sigmoid**	**Linea r**
Validation	83.6%	57.8%	82.7%	76.3%
Re-eval uation	84.1%	60.8%	82.7%	85.7%

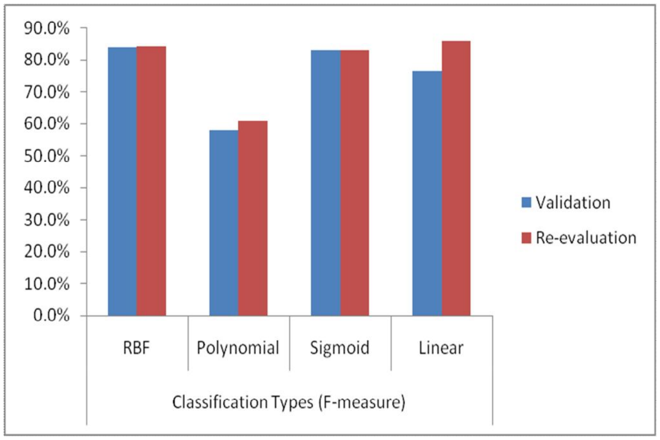

Figure 9: F-Measure (%) of cross validation and re-evaluation of various SVM kernels

Table 6 and Figure 9 shows F-Measure of SVM kernels RBF exhibits good percentage of F-Measure followed by sigmoid. Linear has moderate percentage of F-Measure where as polynomial is only with 50%.

9. CONCLUSION

This paper explored the performance evaluation of different kernel based SVM classifiers to detect intrusion in the cloud environment.SVM based kernel Classifiers are applied to the CIDDS-001 benchmark flow based dataset. This paper throws light to have a concrete judgment in this direction that is to identify the best kernel function among the popular ones like linear, polynomial, Gaussian radial basis function and sigmoid kernels are used to perform classification of cloud attack traffic using ten-fold cross validation.

Upon clear observation of the results and graphs one can conclude that radial basis function kernel provided the best performance of the training data with 88.57% accuracy and test data with 88.81% accuracy as compared to the other Kernel functions type. The experimental results shows that Radial basis function kernel is good at classification accuracy but in the case of computational time, sigmoid kernel provides best results with the average of 1.9 sec for training dataset and 1.64 sec for testing dataset which is much less that the time taken by other kernels. It is preferable to suggest RBF kernel only based on accuracy. In case of sigmoid kernel computational time is less dependent with more or less same degree of accuracy. Therefore sigmoid kernel is recommended. In future this study may be extended on real time experimental data and compare with other kernel methods.

REFERENCES

1. http://en.wikipedia.org/wiki/Cloud_computing.
2. Ms. Parag K. Shelke, Ms. Sneha Sontakke, Dr. A. D. Gawande. ***Intrusion Detection System for Cloud Computing.*** International Journal of Scientific & Technology Research Volume 1, Issue 4, May 2012.
3. Muhammad Shafiq, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, N abin Kumar Karn, F oudil Abdessamia. ***Network Traffic Classification Techniques and Comparative Analysis Using Machine Learning Algorithms***, IEEE Transactions on parallel and distributed systems.Vol.24, No.1, January 2013.
4. Ayman A.A. Ali, Prof. Saif Eldin Fattoh Osman. ***Investigation of Intrusion and Denial of Service Attacks in Cloud Computing***, International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 6, Issue 8, August 2017.
5. James Henderson, IDG Communications, ***https://www.arnnet.com.au/article/608976/major-aussie-websites-impacted-following-global-ddos-attacks/,*** [online].
6. Sandip Hingane, Dr. Umesh Kumar Lilhore. ***Intrusion Detection Techniques: A Review,*** International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2018 IJSRCSEIT,| Volume 3 , Issue 1,| ISSN : 2456-3307.
7. Pradeep Pundir, Dr.Virendra Gomanse, Narahari Krishnamacharya. ***Classification and Prediction techniques using Machine Learning for Anomaly Detection***, International Journal of Engineering Research and Applications (IJERA) , ISSN: 2248-9622
8. R.Ravinder Reddy, B.Kavya, Y Ramadevi. ***A Survey on SVM Classifiers for Intrusion Detection***, International Journal of Computer Applications (0975 – 8887) Volume 98– No.19, July 2014.
9. Fangjun Kuang, Weihong Xua, Siyang Zhang. ***A novel hybrid KPCA and SVM with GA model for intrusion detection***, 1568-4946/$ – see front matter © 2014 Elsevier B.V. All rights reserved, http://dx.doi.org/10.1016/j.asoc.2014.01.028.
10. Vikramaditya Jakkula, ***Tutorial on Support Vector Machine (SVM)***, School of EECS, Washington State University, Volume 37, 2006.
11. Abhishek verma, Virender Ranga. ***Statistical analysis of CIDDS-001 dataset for Network Intrusion Detection Systems using Distance-based Machine Learning***, in 6th International Conf. Smart Computing and Communications, ICSCC, Kurukshetra, India. 2017, pp. 709-716.
12. Mohamed Idhammada, Karim Afdel, Mustapha Belouch. ***Distributed Intrusion Detection System for Cloud Environments based on Data Mining technique***, The First International Conf. Intelligent Computing in Data

Sciences, science direct, Procedia Computer Science 127 2018,pp. 35–41.

13. Mohamed Idhammada, Karim Afdel, and Mustapha Belouch. ***Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest***, Security and Communication Networks Volume 2018, Article ID 1263123, 13 pages , https://doi.org/10.1155/2018/1263123 .
14. Yili Ren, Fuxiang Hu, Hongping Miao. ***The optimization of kernel function and its parameters for SVM in well-logging***, 2016 13th International Conference on Service Systems and Service Management (ICSSSM), china, 2016.
15. Raneel kumar, Sunil Pranit Lal & Alok Sharma. . ***Detecting Denial of Service Attacks in the Cloud***, 978-1-5090-4065-0/16 $31.00 © 2016 IEEE DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.70
16. Chunhua Gu and Xueqin Zhang. ***A Rough Set and SVM Based Intrusion Detection Classifier***, Second International Workshop on Computer Science and Engineering, 2009.
17. Yong-Xiang Xia, Zhi-Cai Shi, Zhi-Hua Hu. ***An Incremental SVM for Intrusion Detection Based on Key Feature Selection***, Third International Symposium on Intelligent Information Technology Application, 2009.
18. CIDDS-001 dataset. (2017, Aug.) [Online] Available: ***https://www.hs-coburg.de/forschung kooperation/forschungsprojekteoeentlich/ingenieur wissenschaften/cidds-coburg-intrusion-detection-data-sets.html***
19. B.Basaweswara Rao, K.Vamsi Krishna, K.Swathi. ***A Fast KNN Based Intrusion Detection System For Cloud Environment***, Journal of Advanced Research Dynamical & Control Systems, Vol . 10, 07-Special Issues, 2018.
20. M Ring, S Wunderlich, D Grüdl, D Landes and A Hotho. ***Creation of Flow-Based Data Sets for Intrusion Detection***, Journal of Information Warfare Vol. 16, No. 4 (Fall 2017), pp. 41-54.
21. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. ***A Practical Guide to Support Vector Classification***, National Taiwan University, Taipei 106, Taiwan http://www.csie.ntu.edu.tw/~cjlin Initial version: 2003 Last updated: April 15, 2010.
22. S. B. Kotsiantis, D. Kanellopoulos and P. E. Pintelas. ***Data Preprocessing for Supervised Leaning*** , World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, Vol:1, No:12, 2007.
23. T. Nathiya, G. Suseendran. ***An effective way of cloud intrusion detection system using decision tree , support vector machine and naïve bayes algorithm***, International Journal of Recent Technology and Engineering (IJRTE), Volume-7 Issue-4S2, December 2018.
24. Pavithra G , Abirami P , Bhuvaneshwari S , Dharani S , Haridharani B." **A Survey on Intrusion Detection Mechanism using Machine Learning Algorithms", IJETER**, April 2020,pp. 945 – 949
25. Cherry D. Casuat1, Enrique D. Festijo2, Alvin Sarraga Alon3," **Predicting Students' Employability using Support Vector Machine: A SMOTE-Optimized Machine Learning System**", IJETER, Volume 8. No. 5, May 2020, pp. 2101 - 2106