International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013)

Intrusion Detection System Using Data Mining Technique: Support Vector Machine

Yogita B. Bhavsar[1], Kalyani C.Waghmare[2]

[1]*Post Graduate Student,* [2]*Assistant Professor, Pune Institute of Computer Technology, Pune, Maharashtra, India*

***Abstract*— Security and privacy of a system is compromised, when an intrusion happens. Intrusion Detection System (IDS) plays vital role in network security as it detects various types of attacks in network. So here, we are going to propose Intrusion Detection System using data mining technique: SVM (Support Vector Machine). Here, Classification will be done by using SVM and verification regarding the effectiveness of the proposed system will be done by conducting some experiments using NSL-KDD Cup'99 dataset which is improved version of KDD Cup'99 data set. The SVM is one of the most prominent classification algorithms in the data mining area, but its drawback is its extensive training time. In this proposed system, we have carried out some experiments using NSL-KDD Cup'99 data set. The experimental results show that we can reduce extensive time required to build SVM model by performing proper data set pre-processing. Also when we do proper selection of SVM kernel function such as Gaussian Radial Basis Function, attack detection rate of SVM is increased and False Positive Rate (FPR) is decrease.**

***Keywords*— Classification, Intrusion Detection System (IDS), Kernel Function, NSL- KDD, Pre-processing, Support Vector Machine (SVM)**

I. INTRODUCTION

As network-based computer systems have important roles in modern society, they have become the targets of intruders. Therefore, we need to find the best possible ways to protect our systems. The security of a computer system is compromised when an intrusion takes place. An intrusion can be defined as any action done to hamper the integrity, confidentiality or availability of the system. There are some intrusion prevention techniques which can be used to protect computer systems as a first line of defense. But only intrusion prevention is not enough. As systems become more complex, there are always exploitable weaknesses in the systems due to design and programming errors, or various penetration techniques. Therefore Intrusion detection is required as another measure to protect our computer systems from such type of vulnerabilities.

II. RELATED WORK

In 1980, the concept of intrusion detection began with Anderson's seminal paper [1]; he introduced a threat classification model that develops a security monitoring surveillance system based on detecting anomalies in user behavior.

In 2003, Kaining Lu Zehua Chen Zhigang Jin Jichang Guo, [4] has presented one collaborate IDS module to make a real-time detection and block intrusions before occurrences based on HIDS using sequences of system call anomaly detection. In 2009, Chunhua Gu and Xueqin Zhang,[6] proposed a system using rough set for attribution reduction and support vector machine for intrusion detection classification .In 2009, Yong-Xiang Xia Zhi-Cai Shi and Zhi-Hua Hu ,[5] proposed a method of detecting intrusion using incremental SVM based on key feature selection. Again in the same year, Rung-Ching Chen, Kai-Fan Cheng and Chia-Fen Hsieh,[7] used RST (Rough Set Theory) and SVM (Support Vector Machine) to detect intrusions. First, RST is used to preprocess the data and reduce the dimensions. Next, the features were selected by RST will be sent to SVM model to learn and test respectively.

In 2010, Heba F. Eid [8] effectively introduced intrusion detection system by using Principal Component Analysis (PCA) with Support Vector Machines (SVMs) as an approach to select the optimum feature subset. In 2011, Shingo Mabu, Member, *IEEE,* [9] has described a novel fuzzy class-association rule mining method based on genetic network programming (GNP) for detecting network intrusions. Again in the same year, Carol J Fung and Jie Zhang, [10] have proposed Dirichlet-based trust management to measure the level of trust among IDSes according to their mutual experience.

Recently in 2012, [11] has described an adaptive network intrusion detection system which uses a two stage architecture. In the first stage a probabilistic classifier is used to detect potential anomalies in the traffic. In the second stage a HMM based traffic model is used to narrow down the potential attack IP addresses. Again in 2012, V. Jaiganesh, [15] proposed Kernelized Support Vector Machine with Levenberg-Marquardt (LM) Learning. Again In 2012, Gholam Reza Zargar, Tania Baghaie, [13] proposed a category-based selection of effective parameters for intrusion detection using Principal Components Analysis (PCA).

III. DATA SET COLLECTION AND PRE-PROCESSING

A. Data Set Collection

To verify the effectiveness and the feasibility of the proposed IDS system, we have used NSL-KDD dataset.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013)

It is a new version of KDDcup99 dataset. NSL-KDD dataset has some advantages over KDDcup99 dataset. It has solved some of the inherent problems of the KDDcup99 [21], which is considered as standard benchmark for intrusion detection evaluation [17]. The training dataset of NSL-KDD similar to KDDcup99 consist of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or attack type ,with exactly one specific attack type.

Due to following reasons, NSL-KDD has become more popular dataset than KDD cup 99 dataset for intrusion detection purpose.

- Redundant records from the training set are eliminated.
- Duplicate records from the test set are removed to improve the intrusion detection performance.
- Use of NSL-KDD dataset for classification gives an accurate evaluation of different learning techniques.
- It is affordable to use NSL-KDD dataset for experiment purpose as it consists of reasonable numbers of instances both in the training set and testing set.

B. Data Set Pre-Processing

Pre-processing of original NSL-KDD dataset is necessary to make it as a suitable input for SVM. Data set pre-processing can be achieved by applying:

i. Data set transformation
ii. Data set normalization
iii. Data set discretization

i) *Data set transformation:* The training dataset of NSL-KDD consist of approximately 4,900,000 single connection instances. Each connection instance contains 42 features including attacks or normal. From these labeled connection instances, we need to transform the nominal features to numeric values so as to make it suitable input for classification using SVM. For this transformation, we will use table 2. Also, we have to assign numeric value to the last feature in the connection instance which is target class. For doing this, we have assigned a target class 'zero' for 'normal connection' and a 'one' for any deviation from that (i.e. if that is an attack) as per transformation table 2.In this step, some useless data will be filtered and modified. For example, some text items need to be converted into numeric values. Every instance in the dataset has 42 features or attributes including target class shown in Table 1.

TABLE I
FEATURES OF NSD- KDD CUP'99 DATASET

Sr. No	Feature Name
1	Duration
2	Protocol_type
3	Service
4	Flag
5	Src_bytes
6	Dst_bytes
7	Land
8	Wrong_fragment
9	Urgent
10	Hot
11	Num_failed_logins
12	Logged_in
13	Num_compromised
14	Root_shell
15	Su_attempted
16	Num_root
17	Num_file_creations
18	Num_shells
19	Num_access_files
20	Num_outbound_cmds
21	Is_host_login
22	Is_guest_login
23	Count
24	Srv_count
25	Serror_rate
26	Srv_serror_rate
27	Rerror_rate
28	Srv_rerror_rate
29	Same_srv_rate
30	Diff_srv_rate
31	Srv_diff_host_rate
32	Dst_host_count
33	Dst_host_srv_count
34	Dst_host_same_srv_rate
35	Dst_host_diff_srv_rate
36	Dst_host_same_src_port_rate
37	Dst_host_srv_diff_host_rate
38	Dst_host_serror_rate
39	Dst_host_srv_serror_rate
40	Dst_host_rerror_rate
41	Dst_host_srv_rerror_rate
42	Normal or Attack

An example of original NSL-KDD data set record is shown in figure 1.

0 tcp ftp_data SF491 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 1 0 0 150 25 0.17 0.03 0.17 0 0 0 0.05 0 normal
0 udp other SF 146 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 13 1 0 0 0 0 0.08 0.15 0 255 1 0 0.6 0.88 0 0 0 0 0 normal

Figure 1: The Original NSL KDD cup'99 dataset

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013)

TABLE II
TRANSFORMATION TABLE

Type	Feature Name	Numeric Value
Attack or Normal	Normal	0
Attack or Normal	Attack	1
Protocol_type	TCP	2
	UDP	3
	ICMP	4
Flag	OTH	5
	REJ	6
	RSTO	7
	RSTOS0	8
	RSTR	9
	S0	10
	S1	11
	S2	12
	S3	13
	SF	14
	SH	15
Service	All services	16 to 81

There are several nominal values like http, tcp, SF in the dataset. Hence we have to transform these nominal values to numeric values in advance. For example, the service type of "tcp" is mapped to 2,"udp" is mapped to 3,"icmp" is mapped to 4 and we will follow Table 2 to transform the nominal values of dataset features into the numeric values. After transformation, the original NSL-KDD cup'99 dataset will become as shown in Figure 2.

0,2,32,14,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,150,25,0.17,0.03,0.17,0,0,0,0.05,0,0
0,3,16,14,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0,0,0,0,0.08,0.15,0,255,1,0,0.6,0.88,0,0,0,0,0,0

Figure 2: Original NSL- KDD cup'99 dataset after transformation

ii) Data set normalization: Dataset normalization is essential to enhance the performance of intrusion detection system when datasets are too large. Here, we have used min-Max method of normalization.

iii) Data set discretization: Dataset discretization technique is used for continuous features selection of intrusion detection and to create some homogeneity between values, which have different data types. Here, we have used range discretization technique for this purpose.

IV. Attack Categories

Attacks are grouped into following four categories:

- Denial of Service(DoS) such as back, land, neptune etc.
- Remote-to-Local (R2L) such as imap, sendmail, phf etc.
- User to Root (U2R) such as buffer_overflow, sqlattack
- Probing such as mscan, saint, satan etc.

Simulated attacks are grouped into four categories as shown in Table 3.

TABLE III
ATTACK CATEGORIES

Attack Types	Category
Normal	Normal
apacha2	Dos
back	
land	
mailbomb	
netpune	
pod	
processtable	
smurf	
teardrop	
udpstorm	
buffer_overflow	U2R
httprunnel	
loadmodule	
perl	
ps	
rootkit	
sqlattack	
xterm	
ftp-write	R2L
guess_password	
imap	
multihop	
named	
phf	
sendmail	
snmpgetattack	
snmpguess	
spy	
warezclient	
warezmaster	
worm	
xlock	
xsnoop	
lpsweep	Probing
mscan	
namp	
portsweep	
saint	
satan	

V. Conversion Of Datasets To LIBSVM Format And Linear Scaling

Pre-processed datasets are converted to libSVM format. For this process, first categorical features from both training and testing datasets are converted to numeric value and then we have to determine target classes for classification phase.

IJETAE
EXPLORING RESEARCH AND INNOVATIONS
International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 3, March 2013)

Here, we have determined two target classes: class ‘zero’ for normal instance and class ‘one’ for attack or intrusion. Then we have to save target class and feature values of each instance in libSVM format. LibSVM format is:

[Label][index1]:[value1][index2]:[value 2]…….

Where,

‘Label’ is target classes of classification. Usually we put integers for the class value. We can have [0, 1] for target class or [-1, +1] for target class.

Here, we have used [0, 1] target class. Where class ‘0’ indicates ‘normal’ and class ‘1’ indicates ‘attack’.

‘Index’ is the ordered index. Usually continuous integer.

‘Value’ is the input data for training. Usually lots of real numbers.

Input dataset to the problem we are trying to solve involves 41 of ‘features’, so the input will be a set of these 41 features.

After this conversion, we have to perform linear scaling of libSVM format datasets and store these scaled datasets for further use. Linear scaling of datasets is done to improve the performance of classification using SVM.

VI. SUPPORT VERCTR MACHINE

The Support Vector Machine is one of the most successful classification algorithms in the data mining area.SVM uses a high dimension space to find a hyper-plane to perform binary classification.SVM approach is a classification technique based on Statistical Learning Theory (SLT).It is based on the idea of hyper plane classifier. The goal of SVM is to find a linear optimal hyper plane so that the margin of separation between the two classes is maximized.

The SVM uses a portion of the data to train the system. It finds several support vectors that represent the training data. These support vectors will form a SVM model. According to this model, the SVM will classify a given unknown dataset into target classes.

VII. INTRUSION DETECTION USING SVM

In the proposed system, we have constructed a SVM model for classification. While intrusion behaviors happen, SVM will detect the intrusion. A classification task involves training set and testing set which consist of instances. Each instance in the training set contains one “target value" (class labels: Normal or Attack) and several “attributes" (features).The goal of SVM is to produce a model which predicts target value of data instance in the testing set which is given only attributes.

To achieve this goal, we have used kernel functions available with SVM. There are 3 major SVM kernel functions:

(i) Gaussian Kernel (Radial Basis Function)
(ii) Polynomial kernel
(iii) Sigmoid kernel

(i) Gaussian Kernel Function: The Gaussian kernel is an example of radial basis function kernel.

$$K(Xi,Xj) = \exp\{-(\|Xi - Xj\|)/2\boldsymbol{\sigma}$$

Where, **σ** stands for window width.

(ii) Polynomial Kernel Function: This Polynomial kernel is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized. Adjustable parameters are the constant term **c** and the polynomial degree **d**.

$$K(Xp,Xj) = (Xp,Xj)^{\mathbf{d}} + \mathbf{c}$$

(iii) Sigmoid Kernel Function: Sigmoid Kernel is also called as Hyperbolic Tangent Kernel and the Multilayer Perceptron (MLP) kernel. The Sigmoid Kernel comes from the Neural Networks field, where the bipolar sigmoid function is often used as an activation function for artificial neurons

$$K(Xi,Xj) = \tanh(k(Xi,Xj) + r$$

In classification phase, SVM training model is build and SVM kernel function is selected to generate classification results. The system design for IDS using SVM is shown in figure 3.

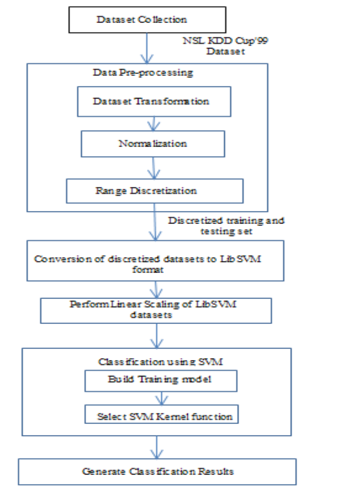

Figure 3: System Design for IDS using SVM

VIII. EXPERIMENTAL RESULTS

The proposed Intrusion Detection System is experimented using the Waikato Environment for Knowledge Analysis (WEKA 3.7) and LibSVM 1.5.

WEKA is a complete set of Java class libraries that execute several state-of-the-art machine learning and data mining approaches [20] and LibSVM is a library for support vector machines [19].Dataset used for experiment purpose is NSL-KDD dataset which is a new version of KDDcup99 dataset and has some advantages over KDDcup99 dataset**.**

Time required to build model using different SVM kernel functions and 10 fold cross validation as well as 10 fold cross validation with re-evaluation using supplied test set is shown in table 4 and respective graph is shown in figure 4.

Table 5 shows, accuracy achieved using different SVM kernel functions and classification using 10 fold cross validation as well as 10 fold cross validation with re-evaluation using supplied test set. The graph for accuracy using different SVM kernel functions at the time of classification is shown in figure 5.

TABLE IV
TIME TAKEN TO BUILD MODEL USING SVM

Kernel Type	Classification Type	Time taken to build model(seconds)
RBF Kernel	Cross Validation	**77.01**
	Cross Validation and Revaluate	**77.01**
Polynomial Kernel	Cross Validation	3859.57
	Cross Validation and Revaluate	3859.57
Sigmoid Kernel	Cross Validation	134.64
	Cross Validation and Revaluate	615.75

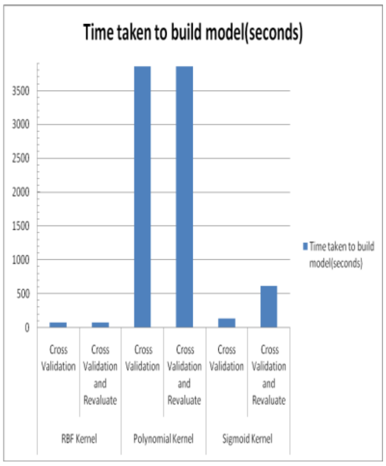

Figure 4: Model Building Time with different SVM kernel functions

TABLE V
ACCURACY OF DIFFERENT SVM KERNEL FUNCTIONS

Kernel Type	Classification Type	Accuracy
RBF Kernel	Cross Validation	94.1857 %
	Cross Validation and Revaluate	**98.5749 %**
Polynomial Kernel	Cross Validation	98.4281 %
	Cross Validation and Revaluate	98.4281 %
Sigmoid Kernel	Cross Validation	98.5749 %
	Cross Validation and Revaluate	73.0886 %

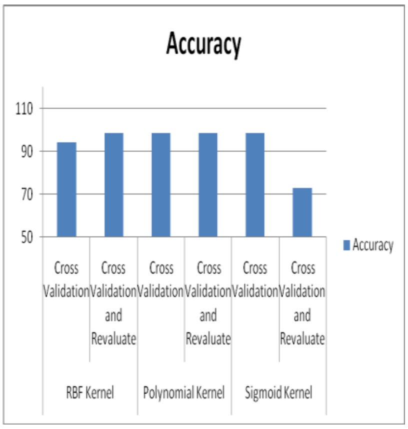

Figure 5: Accuracy of different SVM Kernel functions

IX. CONCLUSION

Now a days, intrusion which affects the security and privacy of the system, has become major concern for many organizations. Hence, there is a need of strong IDS which can detect novel attack with high attack detection accuracy. In this paper, we have proposed a method of intrusion detection using SVM which can reduce the time required to build model for classification and increase the intrusion detection accuracy when Gaussian RBF kernel is used.

The experimental results show that, when data sets are properly processed and proper SVM kernel is selected i.e. Radial Basis Function (RBF), it can overcome the drawback of SVM i.e. extensive time required to build model.

When we have conducted experiment with 10 fold cross validation and Gaussian RBF kernel of SVM, the time required to build model was **77.07 seconds** and attack detection accuracy achieved was **94.1857 %** .This attack detection accuracy was increased to **98.5749 %,** when we have changed classification to 10 fold cross validation and re-evaluation using supplied test set with same RBF SVM kernel function.

REFERENCES

[1] James P. Anderson, "Computer Security Threat Monitoring and Surveillance," Technical report, James P. Anderson Co., Fort Washington, Pennsylvania. April 1980.

[2] Tomas Abraham, "IDDM: INTRUSION Detection using Data Mining Techniques", Technical report DTSO electronics and surveillance research laboratory, Salisbury.

[3] Wenke Lee and Salvatore J. Stolfo, "A Framework for constructing features and models for intrusion detection systems", ACM transactions on Information and system security (TISSEC), vol.3, Issue 4, Nov 2000.

[4] Kaining Lu Zehua Chen Zhigang Jin Jichang Guo." An Adaptive Real-Time Intrusion Detection System Using Sequences of System Call", CCECE 2003.

[5] Yong-Xiang Xia, Zhi-Cai Shi, Zhi-Hua Hu," An Incremental SVM for Intrusion Detection Based on Key Feature Selection" 2009 Third International Symposium on Intelligent Information Technology Application.

[6] Chunhua Gu and Xueqin Zhang," A Rough Set and SVM Based Intrusion Detection Classifier", Second International Workshop on Computer Science and Engineering, 2009.

[7] Rung-Ching Chen, Kai-Fan Cheng and Chia-Fen Hsieh ,"Using Rough Set And Support Vector Machine For Network Intrusion Detection" International Journal of Network Security & Its Applications (IJNSA), Vol 1, No 1, 2009.

[8] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham" Principle Components Analysis and Support Vector Machine" based Intrusion Detection System", IEEE 2010.

[9] Shingo Mabu, Ci Chen, Nannan Lu, Kaoru Shimada,and Kotaro Hirasawa," An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 41, No. 1, January 2011.

[10] Carol J Fung and *Jie* Zhang," Dirichlet-Based Trust Management for Effective Collaborative Intrusion Detection Networks" ,IEEE Transactions on Network And Service Management, Vol. 8, No 2, June 2011.

[11] R Rangadurai Karthick, Vipul P. Hattiwale, Balaraman Ravindran," Adaptive Network Intrusion Detection System using a Hybrid Approach ", IEEE 2012.

[12] Vincent F. Mancuso, Dev Minotra, Nicklaus Giacobe, MichaelMcNeese and Michael Tyworth " *idsNETS*: An Experimental Platform to Study Situation Awareness for Intrusion DetectionAnalysts",IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, New Orleans, LA, 2012.

[13] Gholam Reza Zargar, Tania Baghaie, "Category-Based IntrusionDetection Using PCA", Journal of Information Security, 2012.

[14] Neethu *B,* "Classification of Intrusion Detection Dataset using machine learning Approaches", International Journal of Electronics and Computer Science Engineering, 2012.

[15] V. Jaiganesh, "Intrusion Detection Using Kernelized Support Vector Machine With Levenbergmarquardt Learning", International Journal of Engineering Science and Technology, 2012.

[16] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali, A.Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", proceeding of IEEE symposium on computational Intelligence in security and defense application, 2009.

[17] KDD Cup 1999. Available on http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, October 2007.

[18] R.O. Duda, P.E. Hart, and D. G. Stork, "Pattern Classification", Vol. 1, Wiley, 2002.

[19] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[20] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009);the WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[21] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani A Detailed Analysis of the KDD CUP 99 Data Set proceeding of the 2009 IEEE symposium on computational Intelligence in security and defense application.