ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

A machine learning framework of neural network combined with PLS regression and artificial immune system for intrusion detection

G. V. Pradeep Kumar[1], D. Krishna Reddy[2]

Assistant Professor, Department of ECE, C.B.I.T., Hyderabad, India [1]

Professor, Department of ECE, C.B.I.T., Hyderabad, India [2]

Abstract: An intrusion is essentially an attack on security layer in a TCP/IP protocol stack. Such attacks have definite signature for instance specific ports or RTT. Hence such attacks can be detected by cross verifying current communication signature with attack signatures. However attack signatures are evolutionary. Therefore using a string matching technique is neither robust nor fast. Hence several machine learning techniques are developed which are mainly based on classifier. These classifiers lack generalization capabilities which result in less performance leading to high false positives. However a specific attack can have wide range of signatures and a signature may of wide range of attacks. Therefore conventional classifiers like Neural Network needs frequent training when a new signature is discovered. Again discovery of such new signature also needs a regression with existing signature database. In order to avoid exploding the training nodes of Neural Network, it is important that a benchmark is set of introduce new nodes.

In this paper we use Artificial Immune System mark the signatures as genes. A packet or network level signature is verified for closeness with existing model. In case of significant diversification is detected, the signature is marked as new which is regressed with the existing signature model to automate the grouping of the signature. Signature similar to existing ones is subjected to regression using PLS method and is then classified by neural network. This paper evaluates the performance of technique using the publicly available KDD Cup dataset and compares the result with conventional Neural Network Based Classifier, Support Vector Machine based Classifier, pure regression based technique and conventional string matching technique. Further we investigate the real time applicability of the technique by using PLS Regression to detect anomaly in CIT college router dataset. We consider that the connections by peer clients should only be accessing internet. Other activities like using Bittorrent are considered as anomaly. First we take router log and extract the features. We then select a specific data row and classify it using auto regression.

Keywords: PLS Regression, Neural Network, KDD, DDoS.

I. INTRODUCTION

The objective of the system is to develop an Intrusion detection system based on learning technique. Firstly known classes of intrusion like DDOS, Perl attack, Neptune attack signature is formed from standard KDD dataset. This has several string values which are not understood by the classifier. Therefore these values are converted to suitable numbers based on their properties.

Database is partitioned into two parts: Training and Testing. Testing involves giving one row from the dataset as input. System classifies the row as Normal or Abnormal.The same concept is then adopted in a real time environment to detect anomaly in internet access from college data. Router log is used to extract the features. As these features are not preclassified, we use a regression technique rather than classification to find the similarity with any data of earlier dates. Baseon protocol is used; we then classify the data as normal or abnormal.

Multiple alert of the same signature leads to misleading inferences for intrusion database. Therefore system should be able to detect and store only those intrusions that are relevant for future detection and those that are significantly independent signature. Generally such a system work offline where firstly all the intrusions are marked as they appear and then the aggregation system aggregates the data.

An efficient intrusion detection system must be self learning. In order to validate the learning mechanism we partition the 9 attack modules of KDD cup data into three parts. One part is used as preliminary training, the second one being used to evaluate and validate the learning process and the third part is used to classify the attack signature.

Signature validation and learning is performed using Artificial Immune System. AIS is an agent based system which keeps monitoring network performances and then learns from attacks. We apply AIS validation to know if newly detected case of intrusion differs from existing database to a great deal. If so the new signature is put in the database and regressed. Regression process validates if the new signature can be grouped in any of existing classes of attack or should be treated as an entirely new class. In case the signature is found to be a new class, neural network is retrained with increased epochs to re-weight the hidden layer.

Problem Statement

IP packet/Network Properties/Router Log based intrusion detection is not proposed as a single model in any of the existing literature. Therefore classification, aggregation and intrusion detection is proposed by this paper using the

DOI 10.17148/IJARCCE

combination of Neural Network; PLS regression and AIS based technique. Detected patterns or signatures are used for future database for such a system. Intrusion detection systems are generally fast pattern or string matching algorithms that detect the intrusion. Hence a huge database size leads to significant delay in detection. Therefore detected signatures must be aggregated and sorted for precise interpretation of the detection. The work statement can be summarized as to detect packet and network level intrusion and summarize them for faster interpretation of the intrusion patterns and for future reuse of the patterns.

Kdd cup 99 data set description

Since 1999, KDD'99 [16] has been the most wildly used data set for the evaluation of anomaly detection methods. This data set is built based on the data captured in DARPA'98IDS evaluation program. DARPA'98[17] is about 4 gigabytes of compressed raw (binary) tcp dump data of 7 weeks of network traffic, which can be processed into about 5 million connection records, each with about 100 bytes. The two weeks of test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labelled as either normal or an attack, with exactly one specific attack type. The simulated attacks fall in one of the following four categories:

1) Denial of Service Attack (DoS): is an attack which makes the user to be denied to use the access of a machine by making the memory busy or full such that it cannot handle further requests.

2) User to Root Attack (U2R): is a class of attack that starts without the access provision to the system using a normal user account and it is also able to take advantage of some lapse in order to gain root access to the system.

3) Remote to Local Attack (R2L): occurs when an attacker behaves like a user without having any account, tries to send the information over a network by exploiting the lapse of the system such that he can gain access as a user and can handle that machine.

4) Probing Attack: is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls. It is primary to make a note that the test and training data do not have the same probability distribution, which includes specific attack types not in the training data which make the task more real. Some of the intrusion experts accept as true that most novel attacks are nothing but the variations of known attacks and their signatures can be sufficient to catch novel variants.

The datasets contain 24 training attack types in total, with an additional of 14 types of test data only.

KDD'99 features can be classified into three groups:

1) Basic features: this category encapsulates all the attributes that can be extracted from a TCP/IP connection. Most of these features leading to an implicit delay in detection.

2) Traffic features: this category includes features that are computed with respect to a window interval and is divided into two groups:

a) "same host" features: These determine only the connections which are having the same destination host as that of the current connection in the past two seconds and thus estimate the statistics associated to protocol service, behaviour, etc.

b) "same service" features: inspect only the connections in the past two seconds that have the same service as that of the current connection.

The two aforementioned types of "traffic" features are called time-based. However, there are several slow probing attacks that scan the hosts (or ports) using a much larger time interval than two seconds, for example, one in every minute. As a result, these attacks do not produce intrusion patterns with a time window of two seconds. To solve this problem, the "same host" and "same service" features are re-calculated but based on the connection window of 100 connections rather than a time window of two seconds. These features are called connection-based traffic features.

3) Content features: unlike most of the Probing attacks and DoS, the R2L and U2R attacks don't have any intrusion frequent sequential patterns. This is because of the involvement of DoS and Probing attacks in many connections to some host(s) in a very short duration of time; on the other hand, the R2L and U2R attacks are entrenched in the data portions of the packets, and normally involves only a single connection. To detect these kinds of attacks, we need some features to be able to look for suspicious behaviour in the data portion, e.g., number of failed login attempts, are called content features.

II. Related Work

Intrusion Detection System is analysed by the use of system behaviour, assuming that some techniques work well with intrusion detection leads to a failure. This system evaluates the behaviour of the intrusion detection system based on system behaviour using the behaviour of the observed system, works at a low level of observing a system[1]. Intrusion Detection systems help in detection and prevention of denial of service attacks and also reduce power consumption in transmission, further pre processing of data plays a vital role as it provides a finite dataset for the IDS. Ganapathy et al. proposed an intrusion detection system based on such pre processing of data which helps in removing redundant features and further in decision making in intrusion detection[2]. Intrusion Detection is based on knowledge of security techniques applied over intrusion. In order to reduce the dependence on security systems different search techniques exists in literature. Tribak et al. proposed intrusion detection using NSL-KDD intrusion detection dataset. Various learning algorithms are applied for the recognition of attacks and further statistical analysis is applied to evaluate the performance of the system[3].

Hari Om et al. proposed a hybrid IDS based on feature extraction using entropy, which helps in selection of important features and reduction in irrelevant ones. Further these features are made into groups using K-means clustering[4]. Intrusion detection is used for handling intrusions that occur in a computer environment by the aid of triggering alerts that makes analysts to take appropriate action in order to block such intrusions. Signature based intrusion detection systems fail as they are not capable in the detection of unknown attacks[5]. Intrusion varies in wired and wireless networks. Hence wireless intrusions based on signatures are complicated. Conventional techniques which are used for intrusion detection fail in wireless networks. Hence, an agent based technique is proposed for gathering information from various nodes with an evolutionary artificial immune system for the detection of intrusion and preventing the same over the intrusive paths[6].

Anish Das et al. proposed divide and conquer feature reduction and also for feature selection over KDD dataset. These features help in the detection of intrusions as attacker will be using various methods to attack which are not identifiable by an intrusion detection system. KDD dataset contains huge number of features which need to be reduced for accurate classification of records [7-8]. KDD dataset uses most recently publicly available exploits and methods. The performance on KDD dataset does not represent its real performance on the attacks. More features do not prove that the classification results will be good enough[9].

Intrusion detection is a process of observing network traffic data for violation of security. Hence mining plays a major role in intrusion detection system. Himadri Chauhan et al. presented a comparison of classification algorithms based on their performance metrics under WEKA environment using KDD dataset[10]. Anamolies in the network traffic are classified into attacks. Gagandeep Kaur et al. proposed a statistical technique for the detection of distributed denial of service attacks based on the observation of anamolies behaviour in the network traffic. Such a technique proved to estimate the time of attack[11]. Quang Anh Tran et al. designed a prototype based on block based neural network combined with software enabled detection engine for intrusion detection. The weights of block based neural network are computed using genetic algorithm. This technique outperformed existing conventional methods[12].

Biological systems are used as inspiration for bio inspired computing. Artificial immune systems are such bio inspired computing which has contributed for complex problem solutions. Modeling is the way to understand immune system. Agent based modeling is used as a component of the Immune system and are agents which interact with each other such that a system behaviour can be setup[13]. Elham Hormozi et al. proposed negative selection algorithm for anamoly detection in artificial immune system on cloud. Experimental tests revealed that using parallel algorithm detection rate have increased to more than half of the use of serial algorithm[14]. AIS is capable of detecting attacks as it is trained to identify the difference between normal and abnormal behaviour during the training period[15].

III. PROPOSED SYSTEM

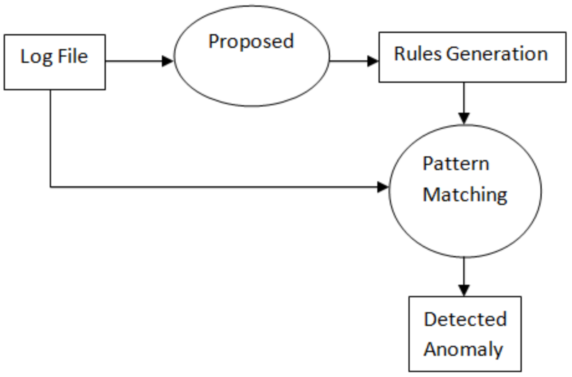


Fig.1:1stLevel Data Flow Diagram

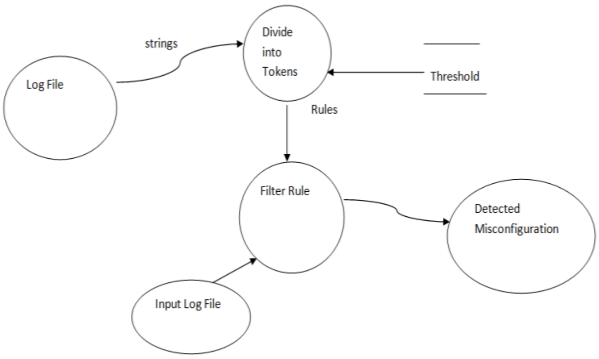


Fig. 2:2ndLevel Data Flow Diagram

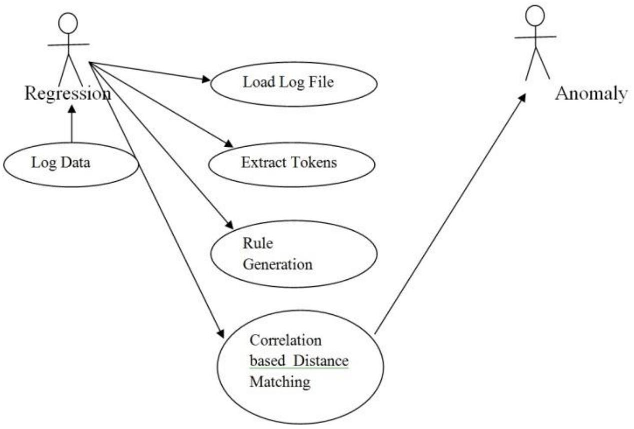
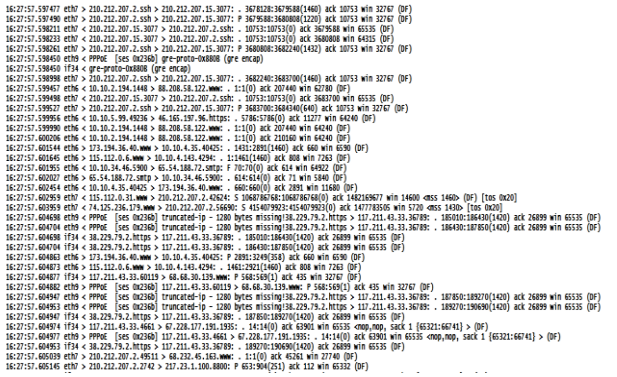


Fig. 3:Anamoly Detection using correlation based distance matching

Fig. 1, Fig. 2 and Fig.3 are the visual representations of the “flow” of data through a system for anamoly detection. The server log file of 90 days is obtained and is tokenized. The screenshot of the part of log file is shown in the Fig. 4 It is interesting to note that even though the log entries are unique and independent over time, there are several entries that creates a signature

Therefore it is important first step to extract independent signatures and group them appropriately. Conventionally such a task is carried out with the help of string matching and clustering techniques.

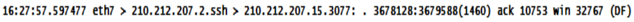
ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940



```
16:27:57.597477 eth7 > 210.212.207.2.ssh > 210.212.207.15.3077: . 3678128:3679588(1460) ack 10753 win 32767 (DF)
16:27:57.597490 eth7 > 210.212.207.2.ssh > 210.212.207.15.3077: P 3679588:3680808(1220) ack 10753 win 32767 (DF)
16:27:57.598211 eth7 < 210.212.207.15.3077 > 210.212.207.2.ssh: . 10753:10753(0) ack 3679588 win 65535 (DF)
16:27:57.598233 eth7 < 210.212.207.15.3077 > 210.212.207.2.ssh: . 10753:10753(0) ack 3680808 win 64315 (DF)
16:27:57.598261 eth7 > 210.212.207.2.ssh > 210.212.207.15.3077: P 3680808:3682240(1432) ack 10753 win 32767 (DF)
16:27:57.598450 eth9 < PPPoE  [ses 0x236b] gre-proto-0x880B (gre encap)
16:27:57.598450 if34 < gre-proto-0x880B (gre encap)
16:27:57.598998 eth7 > 210.212.207.2.ssh > 210.212.207.15.3077: . 3682240:3683700(1460) ack 10753 win 32767 (DF)
16:27:57.599457 eth6 < 10.10.2.194.1448 > 88.208.58.122.www: . 1:1(0) ack 207440 win 62780 (DF)
16:27:57.599498 eth7 < 210.212.207.15.3077 > 210.212.207.2.ssh: . 10753:10753(0) ack 3683700 win 65535 (DF)
16:27:57.599527 eth7 > 210.212.207.2.ssh > 210.212.207.15.3077: P 3683700:3684340(640) ack 10753 win 32767 (DF)
16:27:57.599956 eth6 < 10.10.5.99.49236 > 46.165.197.96.https: . 5786:5786(0) ack 11277 win 64240 (DF)
16:27:57.599990 eth6 < 10.10.2.194.1448 > 88.208.58.122.www: . 1:1(0) ack 207440 win 64240 (DF)
16:27:57.600206 eth6 < 10.10.2.194.1448 > 88.208.58.122.www: . 1:1(0) ack 210160 win 64240 (DF)
16:27:57.601544 eth6 > 173.194.36.40.www > 10.10.4.35.40425: . 1431:2891(1460) ack 660 win 6590 (DF)
16:27:57.601645 eth6 > 115.112.0.6.www > 10.10.4.143.4294: . 1:1461(1460) ack 808 win 7263 (DF)
16:27:57.601955 eth6 < 10.10.34.46.5900 > 65.54.188.72.smtp: F 70:70(0) ack 614 win 64922 (DF)
16:27:57.602027 eth6 > 65.54.188.72.smtp > 10.10.34.46.5900: . 614:614(0) ack 71 win 5840 (DF)
16:27:57.602454 eth6 < 10.10.4.35.40425 > 173.194.36.40.www: . 660:660(0) ack 2891 win 11680 (DF)
16:27:57.602959 eth7 < 115.112.0.31.www > 210.212.207.2.42624: S 1068786768:1068786768(0) ack 1482169677 win 14600 <mss 1460> (DF) [tos 0x20]
16:27:57.603959 eth7 < 74.125.236.179.www > 210.212.207.2.56690: S 4154079923:4154079923(0) ack 1477783505 win 5720 <mss 1430> [tos 0x20]
16:27:57.604698 eth9 < PPPoE  [ses 0x236b] truncated-ip - 1280 bytes missing!38.229.79.2.https > 117.211.43.33.36789: . 185010:186430(1420) ack 26899 win 65535 (DF)
16:27:57.604704 eth9 < PPPoE  [ses 0x236b] truncated-ip - 1280 bytes missing!38.229.79.2.https > 117.211.43.33.36789: . 186430:187850(1420) ack 26899 win 65535 (DF)
16:27:57.604698 if34 < 38.229.79.2.https > 117.211.43.33.36789: . 185010:186430(1420) ack 26899 win 65535 (DF)
16:27:57.604704 if34 < 38.229.79.2.https > 117.211.43.33.36789: . 186430:187850(1420) ack 26899 win 65535 (DF)
16:27:57.604863 eth6 > 173.194.36.40.www > 10.10.4.35.40425: P 2891:3249(358) ack 660 win 6590 (DF)
16:27:57.604873 eth6 > 115.112.0.6.www > 10.10.4.143.4294: . 1461:2921(1460) ack 808 win 7263 (DF)
16:27:57.604877 if34 > 117.211.43.33.60119 > 68.68.30.139.www: P 568:569(1) ack 435 win 32767 (DF)
16:27:57.604882 eth9 > PPPoE  [ses 0x236b] 117.211.43.33.60119 > 68.68.30.139.www: P 568:569(1) ack 435 win 32767 (DF)
16:27:57.604947 eth9 < PPPoE  [ses 0x236b] truncated-ip - 1280 bytes missing!38.229.79.2.https > 117.211.43.33.36789: . 187850:189270(1420) ack 26899 win 65535 (DF)
16:27:57.604953 eth9 < PPPoE  [ses 0x236b] truncated-ip - 1280 bytes missing!38.229.79.2.https > 117.211.43.33.36789: . 189270:190690(1420) ack 26899 win 65535 (DF)
16:27:57.604947 if34 < 38.229.79.2.https > 117.211.43.33.36789: . 187850:189270(1420) ack 26899 win 65535 (DF)
16:27:57.604974 if34 > 117.211.43.33.4661 > 67.228.177.191.1935: . 14:14(0) ack 63901 win 65535 <nop,nop, sack 1 {65321:66741} > (DF)
16:27:57.604977 eth9 > PPPoE  [ses 0x236b] 117.211.43.33.4661 > 67.228.177.191.1935: . 14:14(0) ack 63901 win 65535 <nop,nop, sack 1 {65321:66741} > (DF)
16:27:57.604953 if34 < 38.229.79.2.https > 117.211.43.33.36789: . 189270:190690(1420) ack 26899 win 65535 (DF)
16:27:57.605039 eth7 > 210.212.207.2.49511 > 68.232.45.163.www: . 1:1(0) ack 45261 win 27740 (DF)
16:27:57.605145 eth7 > 210.212.207.2.2742 > 217.23.1.100.8800: P 653:904(251) ack 112 win 65332 (DF)
```

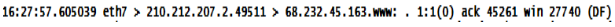
Fig. 4: CIT College Log File

For instance



```
16:27:57.597477 eth7 > 210.212.207.2.ssh > 210.212.207.15.3077: . 3678128:3679588(1460) ack 10753 win 32767 (DF)
```

is an entry of interface Ethernet 7 which is similar to entry



```
16:27:57.605039 eth7 > 210.212.207.2.49511 > 68.232.45.163.www: . 1:1(0) ack 45261 win 27740 (DF)
```

However the logs shows that the IP address accessed may vary from log to log and such IP addresses may also lead to significant information for instance accessing a particular site. Hence a self learning technique should be used as first step for extracting unique and independent signatures.

We perform this with the help of artificial immune system.

AIS Step

In general Artificial Immune System can be categorised into negative selection and clonal selection. The former finds its applications in pattern recognition and classification by attempting to model the entire process using T cells, while the latter is helpful in solving optimisation problems.

In this work we randomly select 6 patterns. We define patterns as sequence of tokens where each entry in a row the log file and dataset file is tokenized. The initial set is created in such a way that they are formed of independent entries. Each pattern is called a self-set

The self-set is a population six tokens (Incoming Port, Outgoing Port, Protocol, Header, Interface Number, IP address). The non-self-pattern set or negative set would describe abnormal behaviour and would have to be generated using the self-set by negative selection. This can be achieved by iterating through thousands of six pattern sequences from dataset or from log file and removing those which match the self-set and keeping those which do not. For example eth7, which does not match the if34 set, would be in the abnormal set.

In this problem, the search step would involve taking an unknown six token entry and matching it against the negative set given certain criteria (i.e., 4 character matches, or in case port number matching) to test if it is abnormal.

Once dataset is ready, it should be used for training. It is highly impossible to segregate and mark thousands of possible independent signatures (or self-set or entries). Hence conventional knowledge based classifiers are difficult to model. Therefore we need to design a self-learning classifier which is performed using PLS regression.

PLS Regression

Partial least squares regression (PLS regression) is a statistical method that bears some relation to principal components regression; instead of finding hyper planes of minimum variance between the response and independent variables, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space. Because both the *X* and *Y* data are projected to new spaces, the PLS family of methods are commonly referred as bilinear factor models.

PLS is used for finding the fundamental relations between the two matrices (*X* and *Y*), i.e., a latent variable approach to modeling the covariance structures in these two spaces. A PLS model tries to find the multi-dimensional direction in the *X* space that explains the maximum multi-dimensional variance direction in the *Y* space. PLS regression is predominantly suited when the matrix of predictors has more variables than observations, and when multi-collinearity exists among *X* values. By contrast, standard regression will fail in these cases.

However as PLS works on Matrices and not on string, it is important to obtain a numerical pattern from the string pattern extracted through AIS. This is performed in following ways.

1) Obtain all independent entries in protocol field. Create an enumerator by assigning unique number to each protocol.
2) Assign 0 for '>' (outgoing) and 1 for '<' (incoming)
3) See if the entry contains Acknowledgement. If yes, assign 1 else assign 0.
4) Divide both incoming and outgoing IP addresses into 4 fields each and separate address entries.
5) Remodel the extracted set of string self set into a Numeric set using the steps 1-4.

Form a Matrix from this numeric set and subject it to modelling

Modelling

The general underlying model of multivariate PLS is

$$X = TP^T + E$$

$$Y = UQ^T + F$$

where X is an $n \times m$ matrix of predictors, Y is an $n \times p$ matrix of responses: T and U are $n \times l$ matrices that are, respectively, projections of X (the *X* score, component or factor matrix) and projections of Y (the *Y* scores); P and Q are, respectively, $m \times l$ and $p \times l$ orthogonal loading matrices; and matrices E and F are the error terms, assumed to be independent and identically distributed (i.i.d) normal. The decompositions of X and Y are made so as to maximize the covariance of U and T.

A number of variants of PLS are present for estimating the factor and loading matrices T, P and Q. Most of them construct estimates of the linear regression between X and Y as $Y = X\tilde{B} + \tilde{B}_0$.

DOI 10.17148/IJARCCE

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 12, December 2014

Two patterns each from KDD dataset attack types are grouped along with two normal patterns. Closeness of the self set with dataset pattern is obtained using usual string matching technique defined in AIS step and an extra Matrix field in regressed matrix is added called Observation. It contains 1 for abnormal or attack entry and 0 for normal entry.

Any log entry is now classified using ANN (Artificial Neural Network) on the regressed matrix using numerical entry extracted from the pattern. If the classifier returns '1' then the string is matched extensively in KDD dataset to find the matching attack type.

IV. RESULTS

a) Experimentation
b)

We conducted our experiment in two datasets: KDDCup 99 dataset and CIT College's router's log. In the experiment with KDDCup data we randomly partitioned the table into two parts: One used for training and other for testing. We evaluated proposed method against

1) String Matching based Technique (Regx based)
2) ANN based technique
3) SVM based technique
4) PLS Regression based Technique and
5) AIS based technique.

Every technique has its pros and cons and offers some uniqueness over the other. It is important therefore to analyse the performance of independent technique alongside the proposed technique.

Methods were tested for False Positive, False Negative, Rate of Classification, Classification time against number of classes of attack and number of training samples.

We further carry out testing on the real dataset using proposed technique and only PLS regression based technique to evaluate the speed of detection. Experiment results are as presented below.

c) Results and Analysis

We first need a justification of using a numerical matrix based approach for classification over a regx based string matching technique for matching the signatures. The graph shown in the Fig. 5 is obtained for 10,000 training samples and 20,000 test samples. It can be clearly seen that the proposed technique fares better than string matching based technique and the performance difference is quite clearly visible for higher number of classes. Hence it can be safely said that the numerical model based classification would provide a better model for detecting attacks in pattern signature.

However there are several classifiers: Nearest Neighbour classifier, ANN, Fuzzy based Classifier, SVM and so on. Therefore a performance analysis is important among various classifiers to justify the proposed model.

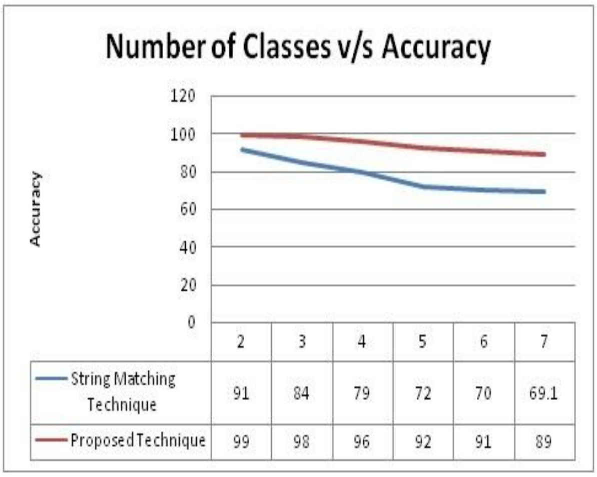


Fig. 5: Number of Classes versus Accuracy

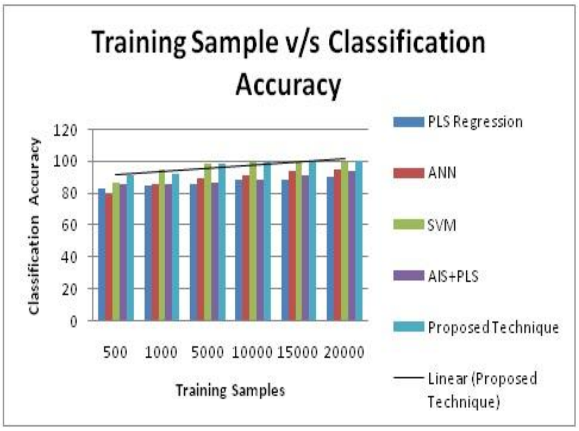


Fig. 6: Training Sample versus Classification Accuracy

The graph shown in the Fig.6 presents the classification accuracy performance of different techniques for 6 Attacks and one normal class. It can be seen that SVM based technique provides good performance even for lower number of training samples. However performance of all the techniques converges above 99% for 20000 samples. This proves that numerical modelling is an acceptable method for achieving high accuracy for attack classification. However this performance also clearly shows that the proposed technique fares better than other techniques in terms of accuracy.

Time complexity analysis presents an interesting aspect of the techniques. SVM and ANN which were found to be good classifiers are also found to take excessive time when number of training samples is increased. It is interesting to find that AIS combined with PLS classifies at a faster rate. As for 5000 samples, accuracy of the proposed technique is better than both ANN and SVM based method.

Another interesting fact emerges from the graph shown in the Fig. 7 is that the proposed technique can classify at a better accuracy without compromising on time complexity. As it can be seen, even for 20,000 training sample, proposed technique fares better than other two good classifiers.

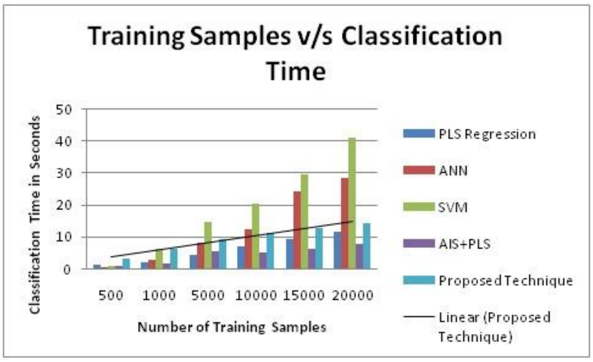


Fig. 7: Training Sample versus Classification Time

The number of classes v/s accuracy graph shown in the Fig. 8 is developed through experiment with 1000 training samples and 20,000 testing samples. It is very interesting to see that for 2 classes, SVM gives 100% result. However as the number of classes increase, performance of the classifiers decreases. For 7 classes performance of proposed system is almost similar to SVM based technique. For five classes, proposed technique is a stand out winner

.

Hence it can clearly be claimed that the proposed can be used with medium number of training samples, for optimum solution in terms of accuracy and speed acceptability. It can also be seen that using AIS in collaboration with PLS significantly improves the performance of PLS based technique.

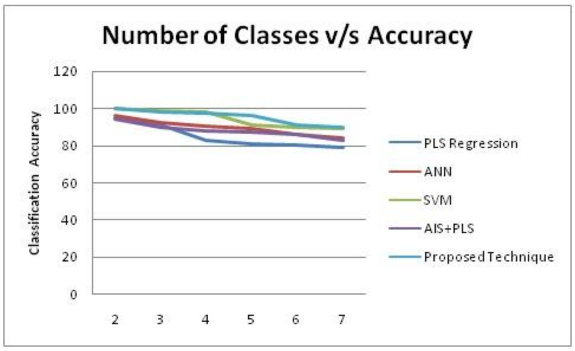
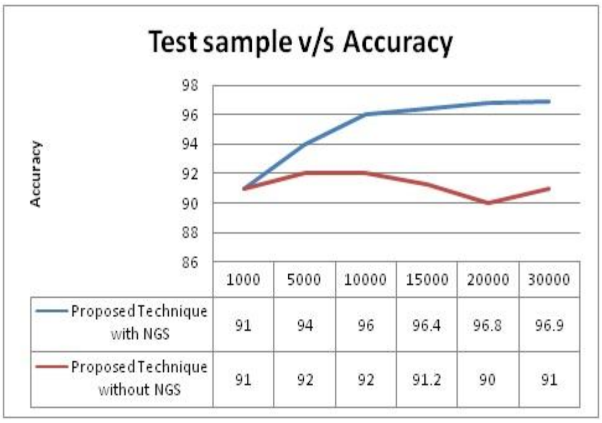


Fig. 8: Number of Classes versus Accuracy

In order to validate the role of Negative gene selection we introduced a feedback in the testing phase. Every percent count misclassification triggers readjustment of training process with negative gene selection. This experiment is conducted with 5000 training samples and 5 classes.

Results obtained in the Fig. 9 show that with negative gene selection, performance of the method can be significantly improved. As the testing triggers misdetection, if the training samples are reselected or dropped based on the result, the result can produce a good classifier. This is done by introducing the misdetected signature in the training set and eliminating the signature closest to it in terms of numerical manhatten distance.



	1000	5000	10000	15000	20000	30000
Proposed Technique with NGS	91	94	96	96.4	96.8	96.9
Proposed Technique without NGS	91	92	92	91.2	90	91

Fig. 9: Number of Test Samples versus Accuracy

It can be seen that both String Matching and nearest neighbour classifier which is similar to string matching in number domain has high false acceptance rate. False acceptance should be low for a system to acceptable in an attack detection scenario.

It is interesting to find that proposed technique has the best performance in terms of low False Acceptance Rate. The graph shown in the Fig. 10 is obtained with 5000 training and 20000 test samples. It is also interesting to see that PLS based method performs better than string matching technique but poorer than ANN and SVM based technique. This observation clearly implicates that when the problem is transformed to matrix domain, it improves the accuracy and FAR. Also the samples are to be selected through proper regression for optimum time performance. ANN on such an optimum set can classify better than SVM.

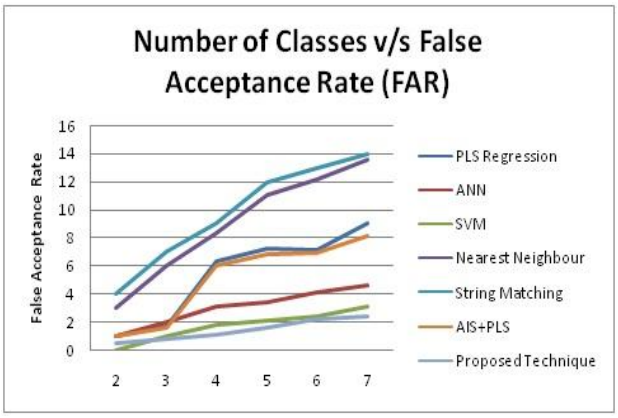


Fig. 10: Number of Classes versus FAR

An Intrusion detection system is said to be acceptable if False Acceptance Rate is lower than False Rejection Rate. Interestingly all the knowledge based classifiers other than nearest neighbour classifier achieves this. However even in the graph, shown in the Fig. 11, it can be clearly seen that the proposed technique fares better than the rest in terms of low FRR.

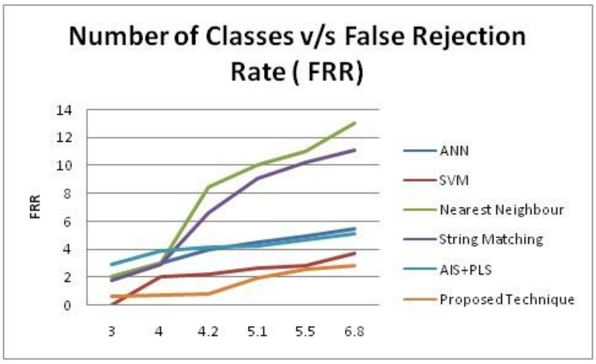


Fig. 11: Number of Classes versus FRR

V. CONCLUSION

The Intrusion detection in real time has been a challenging area in network security and has been evolving over time. As intrusion signatures are evolutionary and attackers often change the attack techniques like port or protocol header, the detection must also be evolutionary. In this work we have presented a unique Intrusion detection system that combines regression and classification. Further the training stage is made evolutionary using AIS. We have shown through extensive result analysis that proposed system can produce better classification accuracy over SVM with a fairly low and acceptable False Acceptance Rate and False Rejection Rate. We have also shown that the proposed technique can significantly improve the time complexity when the size of signature database grows exponentially. The work also is tested against real router log to justify the practical adoptability of the solution. This can be further investigated and improved using partition based classification method.

REFERENCES

[1] Martin Tomášek, Marek Cajkovsky and Branislav Mados, "Intrusion detection system based on system behavior", IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics, January, 2012, pp. 271-275.

[2] S. Ganapathy, K. Kulothungan, Dr. P. Yogesh and Dr. A. Kannan, "An intelligent Intrusion Detection System for ad hoc networks", Third International Conference on Sustainable Energy and Intelligent System, December, 2012, pp. 430-434.

[3] Hind Tribak, Blanca L. Delgado-Marquez, P.Rojas, O.Valenzuela, H. Pomares, I. Rojas, "Statistical analysis of different artificial intelligent techniques applied to Intrusion Detection System", 2012 International Conference on Multimedia Computing and Systems (ICMCS), May 2012, pp. 434-440.

[4] Hari Om, Aritra Kundu, "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system", 1st International Conference on Recent Advances in Information Technology, March 2012, pp. 131 - 136.

[5] Manish Kumar, M. Hanumanthappa, T. V. Suresh Kumar, "Intrusion Detection System using decision tree algorithm", 2012 IEEE 14th International Conference on Communication Technology (ICCT), November, 2012, pp. 629 - 634.

[6] G.V.Pradeep Kumar, D Krishna Reddy, "An Agent Based Intrusion Detection System for Wireless Network with Artificial Immune System (AIS) and Negative Clone Selection", International Conference on Electronic Systems, Signal Processing and Computing Technologies, 2014, pp. 429-433.

[7] Anish Das and Rashmi Bikash Nayak, "A divide and conquer feature reduction and feature selection algorithm in KDD intrusion detection dataset", Third International Conference on Sustainable Energy and Intelligent System, December, 2012, pp. 373-376.

[8] Anish Das and S. Siva Sathya, "A fuzzy approach to feature reduction in KDD intrusion detection dataset", 2012 Third International Conference on Computing Communication & Networking Technologies, July 2012, pp. 1-5.

[9] Gideon Creech, Jiankun Hu, "Generation of a new IDS test dataset: Time to retire the KDD collection", IEEE Wireless Communications and Networking Conference (WCNC): SERVICES & APPLICATIONS, 2013, pp. 4487-4492.

[10] Himadri Chauhan, Vipin Kumar, Sumit Pundir and Emmanuel S. Pilli, "A Comparative Study of Classification Techniques for Intrusion Detection", 2013 International Symposium on Computational and Business Intelligence, 2013, pp.40-43.

[11] Gagandeep Kaur, Suyash Varma, Arpit Jain, "A novel statistical technique for detection of DDoS attacks in KDD dataset", 2013 Sixth International Conference on Contemporary Computing (IC3), August 2013, pp. 393-398.

[12] Quang Anh Tran, Frank Jiang, Quang Minh Ha, "Evolving Block-Based Neural Network and Field Programmable Gate Arrays for Host-Based Intrusion Detection System", 2012 Fourth International Conference on Knowledge and Systems Engineering, 2012, pp. 86-92.

[13] AyiPurbasari, IpingSupriana S, Oerip S. Santoso, Rila Mandala, " Designing Artificial Immune System Based on Clonal Selection: Using Agent-Based Modeling Approach", 7th Asia Modelling Symposium, 2013, pp. 11-15.

[14] Elham Hormozi, Mohammad Kazem Akbari, Morteza Sargolzaei Javan, Hadi Hormozi, "Performance evaluation of a fraud detection system based artificial immune system on the cloud", The 8th International Conference on Computer Science & Education, April, 2013, pp. 819-823.

[15] Karim Ali, Issam Aib, Raouf Boutaba, "P2P-AIS: A P2P Artificial Immune Systems architecture for detecting DDoS flooding attacks", Global Information Infrastructure Symposium, June 2009, pp. 1-4.

[16] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, December 2014.

[17] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall,D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham,and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," discex, vol. 02,p. 1012, 2000.