ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Conditional Feature Optimization for Intrusion Detection by Lib SVM

C.Bhanu Jyothi
M.Tech

Abstract:

In the area of feature reduction for anomaly based Intrusion Detection Systems, Computational Intelligence (CI) methods are increasingly being used for problem solving. This paper concerns using Computational intelligence based learning machines for intrusion detection in ordered order of attacking scenarios, which is a problem of general interest to transportation infrastructure protection since a necessary task thereof is to protect the computers responsible for the infrastructure's operational control, and an effective Intrusion Detection System (IDS) is essential for ensuring network security. We argue that the features opted to detect an attack scenario is not same for all kinds of attacks. Hence here in this paper a ordered feature optimization for Anomaly based Intrusion Detection System (HAB-IDS) is proposed. Two classes of learning machines for IDSs are Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). We consider the SVM in three critical respects of IDSs: SVMs train and run an order of magnitude faster; SVMs scale much better; and SVMs give higher classification accuracy. Hence we use SVM for our proposed Ordered Feature reduction for intrusion detection.

Key words:

IDS, DOS, R2l, U2R, Probe, support vector machine, PSO.

I.Introduction:

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusion [1]. Intrusions are defined as attempts to compromise the confidentiality, integrity or availability of computer or network. They are caused by attackers accessing a system from the internet, by authorized User of the systems who attempt to gain additional privileges for which they are not authorized and by authorized user who misuse the privileges given to them [9]. Anomaly detection and misuse detection [11] are two general approaches to computer intrusion detection system. Unlike misuse detection, which generates an alarm when a known attack signature is matched, anomaly detection identifies activities that deviate from the normal behavior of the monitored system and thus has the potential to detect novel attacks [14]. In this work our aim is to make anomaly-based intrusion feasible. In our experiment, we used DARPA data set. It has solved some of the inherent problems. It is considered as standard benchmark for intrusion detection evaluation [8]. The training dataset of DARPA 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 . Empirical studies indicate that feature reduction technique is capable of reducing the size of dataset. The time and space complexities of most classifiers used are exponential function of their input vector size [15]. Moreover, the demand for the number of samples for the training the classifier grows exponentially with the dimension of the feature space. This limitation is called the 'curse of dimensionality'.In the literature a number of work could be cited wherein several machine learning paradigms, fuzzy inference systems and expert systems, were used to develop IDS [4][5]. Authors of [8] have demonstrated that large number of features is unimportant and may be eliminated, without significantly lowering the performance of the IDS. Very little scientific efforts are diverted to model efficient IDS Feature optimization. IDS task is often modeled as a classification problem in a machine-learning context.The section II exploring the model proposed, section III introduces the methodologies and resources used, section IV explores ordered feature optimization procedure and section V discussing the proposed HFO-ANIDS that fallowed by results discussion, conclusion and references.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

II.Related Work:

A new method that could achieve more accuracy than the existing six classification patterns [Gaussian Mixture, Radial Basis Function, Binary Tree Classifier, SOM, ART and LAMASTAR],called Hierarchial Gaussian Mixture Model[HMM] for IDM was put forward by M.Bahrololum et al[1].Jiankun Hu and Xinghuo Yu et al [2] studied development of host-based anomaly intrusion detection, focusing on system call based HMM training. This was later enhanced with the inclusion of data pre-processing for recognizing and eliminating redundant sub-sequences of system calls, resulting in less number of HMM sub models. Experimental results on three public databases showed that training cost can be reduced by 50% without affecting the intrusion detection performance. False alarm rate is higher yet reasonable compared to the batch training method with a 58% data reduction.

R. Nakkeeran et al[3] proposed an anomaly detection system comprises of detection modules for detecting anomalies in each layer. The anomaly detection result of the neighbor node(s) is taken by the current node and its result in turn is sent to the neighbor node(s).Experimental results showed increased detection rate and reduced false alarm positives, compared to other methods.Jiong Zhang et al[4] proposed a new framework of unsupervised anomaly NIDS based on the outlier detection technique in random forests algorithm. The framework builds the patterns of network services over datasets labeled by the services. With the built patterns, the framework detects attacks in the datasets using the modified outlier detection algorithm, reducing the calculation complexity. This approach is independent of attack-free training datasets, but assumes that each network service has its own pattern for normal activities.

Ahmed Awad E. Ahmed et al[5] proposed a biometrics-based intrusion detector model to provide a lightweight and self-contained module for detecting user identities misuse. System-calls and network traffic monitoring systems have to be combined to this detector to achieve the best solutions.Vijay Bhuse et al[6] proposed a technique to detect anomalies at all layers of a network stack in a sensor network, segregating the service at various levels. Physical layer intrusion is detected by using RSSI values of neighbors (dependant on background noise, weather conditions etc). Targeting MAC layer will work for schedule based and sleep/wake-up based MAC protocols while IASN protocol is aimed at the routing layer.

The experiments show that IASN can be used for source initiated routing protocols, table driven routing protocols and data dissemination mechanisms like directed diffusion and probability of detection increases linearly with the number of nodes running IASN. Nodes guard each other from masquerade at application layer. Depending on the resource availability, any combination of the above methods can be employed, as they are independent of one another. All techniques are energy efficient as they have very low false positive rates(except RSSI and round trip time) and low overhead. Using information theory measures [entropy, mutual information],a model was put forward by Hossein M. Shirazi, that ranked 41 connection features performing normalization on each attack class. The main features of this are, ranking (relevant features for each attack class are selected and computing complexity is decreased) and features-selection (detection rate preserved, yet detection time decreased). Noisy and irrelevant features can be eliminated by running some detection models like SF-5NN and SUS-5NN using only selected features. A combination of two detection engines(SF-KNN,SUS-KNN) based on best selected features and K-NN algorithm was proposed, that was much better(notably in detecting attacks like U2R, R2L) than approaches like traditional 5-NN, C4.5,C5. Experimentally, engines gave classification rates of 92.56%, 92.84% and false positive rates 2% and 4.52% respectively.

Dayu Yang et al [8] introduced a method to apply Auto Associative Kernel Regression (AAKR) empirical modeling and the SPRT for SCADA system intrusion detection. Quick in detecting anomalous behavior, this model is limited by two requirements - different indicators for different intrusion methods and managing a number of highly valuable variables. Identifying the optimal set of indicators for known and potential abnormalities is the future of this research.Combining multiple independent data sources and studying combined traditional intrusion attack and anomaly intrusion, the anomaly intrusion traffic detection work carried out by M.Thangavel et al[9], provided the statistical wavelet based detection mechanism. The examined backscatter data from a mostly unused NetCon server network along with flow anomaly based traffic data from a tier-1 ISP network. The properties such as attack duration, packet count, packet rate, and dominant protocol type match with the two data sets, as is indicated by attack structure.At lean and heavy traffic scenarios, the demand capacity of the server was observed to give better clarity of anomaly intrusion detection although server uptime.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Analysis of several traffic anomaly properties which is impossible using traditional intrusion measurements, can be performed by a new model that used anomaly intrusion attack measurements. Attack predictability (in terms of their originating server as well as interface used to enter a large ISP network). Small businesses seem to be the most common targets of attacks. Traditional measures in understanding and detecting of anomaly intrusion is no more reliable given the current trends of attacking using spoofed address sources. Miao Wang, et al [10], based on an assumption that Windows Native APIs are equal to UNIX system calls [some techniques in UNIX may not be effective in Windows], proposed a method suitable for Windows Host Anomaly Detection System, which is only used as a supplement one for other security mechanisms under windows, because it can only detect intrusions which invoke an anomaly sequence by programs. One of the general situations such as an unauthenticated use of normal programs cannot be detected.Constantine Manikopoulos et al[11], proposed a statistical anomaly detection technology called HIDE with ordered multi-tier multi-observation window system to monitor network traffic parameters simultaneously using a real-time PDF for each parameter, collected during the observation window.

The similarity measurements of measured PDF and reference PDF are comnined into an anomaly status vector classified by a neural network. This methodology detects attacks and soft faults with traffic anomaly intensity as low as 3–5 percent of typical background traffic intensity, thereby generating an early warning.Jeyanthi Hall et al[12], proposed an Anomaly based intrusion detection system for mobile networks, based on simulation results of mobility profiles for enhancing ABID in mobile wireless networks. If the mobility behavior of users has not been accurately found, the selection of specific values for key parameters, such as sequence length and cluster size is absurd. One possible strategy for enhancing the characterization of users and dealingwith concept drift (keeping UMP up to date), is to maintain a window of the newly observed sequences (analogous to the exponential weighted moving average) that can then be used to update the training patterns periodically and hence reduce the false positives.An intrusion detection algorithm and its architecture (two-layered, global central layer and a local layer, together performing data collection, analysis and response), based on data mining and useful in real time for network security, is proposed be HAZEM M. EL-BAKRY et al [13].

By filtering out the known traffic behavior (intrusive and normal) this IDS focuses on analysis on unknown data thereby reducing false alarm rates.

III.Conditional Feature Optimization for Intrusion Detection:

Going into the details of the Ordered Feature Optimization for Anomaly Based Network Intrusion Detection (HFO-ANIDS), which performs consecutive branch based Ordered analysis, which depends availability, confidentiality, and integrity of data and (or) services over a network. Reducing computation and overall time required in detecting bizarre event is achieved by using proposed ordered model. Reduction in time to detect an intrusive event is possible by reducing the communication overhead among different ordered levels and this can be done by making the ordered levels autonomous and self-sufficient in blocking an attack without the aid of a central decision-maker. All the levels in the HAB-IDS framework are trained separately and then sequentially deployed. After defining the four ordered levels, Probe response level, Denial of Service level, REMOTE-TO-LOCAL level, and USER-TO-ROOT level corresponding to the four attack groups mentioned in the data set and then each level is separately trained with a small set of relevant features. Feature optimization is very important for Ordered approach we examine it in the next section.

For making the ordered levels independent, some of the features may be present in more than one of the levels. These ordered levels are vital as such they act as filters that block any bizarre connection, thus preventing the need of further processing at subsequent ordered levels making quick response to intrusion. The net result of that sequence of ordered levels is that the bizarre events are recognized and blocked at the very moment they are detected.Improving the speed of operation of the system is the next goal. For that, the HAB-IDS is put into effect and a small set of features are selected for every level rather than using all the 41 features. The result is that there is significant performance improvement of both the training and the testing of the system. In several situations, there is a trade-off between efficiency and accuracy of the system and various avenues are there to enhance system performance. Methods like naive Bayes, decision trees presume independence among the observed data and this surely increases system efficiency, but it may adversely affect the accuracy.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In order to balance this trade-off, we use the SVM's that are more precise, though expensive, but we implement the Ordered approach to enhance overall system performance. This performance of the proposed system, Ordered feature optimization for Anomaly based intrusion detection that backed by SVM learning, is on par with that of the decision trees and the naive Bayes, and this system has higher attack detection accuracy.

A.Extended QPSO (EQPSO):

We attempt to optimize the QPSO by replacing least good swarm particle with new swarm particle. An interpolate equation will be traced out by applying a quadratic polynomial model on existing best fit swarm particles. Based on emerged interpellant, new particle will be identified. If the new swarm particle emerged as better one when compared with least good swarm particle then replace occurs. This process iteratively invoked at end of each search lap.

The computational steps of optimized QPSO algorithm are given by:

Step 1: Initialize the swarm.
Step 2: Calculate mbest
Step 3: Update particles position
Step 4: Evaluate the fitness value of each particle
Step 5: If the current fitness value is better than the best fitness value (Pbest) in history Then Update Pbest by the current fitness value.
Step 6: Update Pgbest (global best)
Step 7: Find a new particle
Step 8: If the new particle is better than the worst particle in the swarm, then replace the worst particle by the new particle.
Step 9: Go to step 2 until maximum iterations reached.
The swarm particle can be found using the fallowing.

$t_i = \sum_{k=1}^{3} p_i^2 - q_i^2) * f(r)$	$p=a, q=b, r=c \text{ for } k=1;$ $p=b, q=c, r=a \text{ for } k=2;$ $p=c, q=a, r=b \text{ for } k=3$
$t1_i = \sum_{k=1}^{3} p_i - q_i) * f(r)$	$p=a, q=b, r=c \text{ for } k=1;$ $p=b, q=c, r=a \text{ for } k=2;$ $p=c, q=a, r=b \text{ for } k=3$

$$x_i = 0.5 * (\frac{t_i}{t1_i})$$

In the above math notations 'a' is best fit swarm particle, 'b' and 'c' are randomly selected swarm particles is new swarm particle.

B.Feature Selection Based on EQPSO for SVM classifier:

To surpass the usual L2 loss results in least-square SVR, we attempt to optimize hype parameter selection.There are two key factors to determine the optimized features using QPSO: one is how to represent the features as the particle's position, namely how to encode [10,11]. Another is how to define the fitness function, which evaluates the goodness of a particle. The following will give the two key factors.

i.Encoding Features:

The optimized features for SVM include kernel parameter and regularization parameter. To solve features selection by the proposed EQPSO (Optimized Least Significant Particle based QPSO), each particle is requested to represent a potential solution, namely features combination. A features combination of dimension m is represented in a vector of dimension m, such as .

ii.Fitness function:

The fitness function is the generalization performance measure. For the generation performance measure, there are some different descriptions. In this paper, the fitness function is defined as:

$$fitness = \frac{1}{RMSE(\sigma, \gamma)} \quad \ldots (12)$$

Where RMSE(σ ,γ) is the root-mean-square error of predicted results, which varies with the LS-SVM parameters (σ ,γ) . When the termination criterion is met, the individual with the biggest fitness corresponds to the optimal parameters of the LS-SVM.

There are two alternatives for stop criterion of the algorithm. One method is that the algorithm stops when the objective function value is less than a given threshold ε; the other is that it is terminated after executing a pre-specified number of iterations. The following steps describe the EQPSO-Trained LS-SVM algorithm:

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

(1) Initialize the population by randomly generating the position vector iX of each particle and set iP = iX;
(2) Structure SVM by treating the position vector of each particle as a group of features;
(3) Train SVM on the training set;
(4) Evaluate the fitness value of each particle by Eq.(12), update the personal best position iP and obtain the global best position gP across the population;
(5) If the stop criterion is met, go to step (7); or else go to step (6);
(6) Update the position vector of each particle according to Eq.(7), Go to step (3);
(7) Output the gP as a group of optimized parameters.

IV.Proposed Conditional Feature Optimization for Intrusion Detection model:

The verification of a network transaction represented by 41 features is done in a hierarchy. In traditional Anomaly based Intrusion detection systems each transaction analyzed to identify that transaction is normal or abnormal. Hence the feature selection process is selecting features those helpful to detect weather a network transaction is intrusion or not. As the taxonomy [27] of intrusions, each category is different in attack activity and influence of features. Hence the features selection in the context of considering all types of attacks such as DoS, Probe, R2L and U2R are similar is not optimal. The proposed model HFO-ANIDS optimizing the features in the order of hierarchy represented as DoS, Probe, R2L and U2R. The feature optimization might be redundant but leads to fast, accurate detection and nil or negligible false alarming.

V.Conditional Feature Optimization for Intrusion Detection procedure:

Apply EQPSO as described in section IV on given training dataset to optimize the features for DoS, Probe, R2L and U2R attack strategies.Train the SVM on given training dataset based on features optimized for DoS, Probe, R2L and U2R in sequence.Submit the network transaction to be tested to SVM, Hence the SVM verifies for the DoS activity, if found claims the transaction as DoS attack and terminates the verification process.If not SVM continues to analyze for Probe activity, if found claims the transaction as Probe attack and terminates the verification process.If not SVM continues to analyze for R2L activity, if found claims the transaction as R2L attack and terminates the verification process.
If not SVM continues to analyze for U2R activity, if found claims the transaction as U2R attack and terminates the verification process.

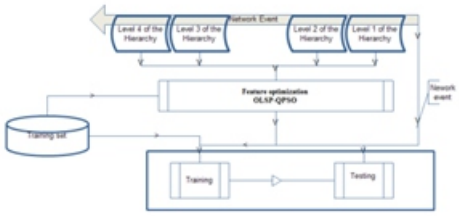

Fig 2: Block Diagram of HFO-ANIDS

VI. Experiments and results discussion:

Standard KDD '99 intrusion data set [16] is used in this experiments.1998 DARPA intrusion detection evaluation program, which is prepared and managed by the MIT Lincoln Laboratory, is the original version of this dataset. Five million connection records as the training data and about two million connection records as the test data are contained in this data set. 10 percent of the total training data and 10 percent of the test data (with corrected labels), which are provided separately are used in these experiments and this leads to about 494,020 training and 311,029 test instances. T the data set in each of these records indicates a connection between two IP addresses, one starting and the other ending at some well defined times along with a well-defined protocol. Also, every record is typified by 41 different features. A separate connection is represented by each record. Therefore each and every record is distinct.

The training data is either labeled as normal or as one of the 24 different kinds of attack. Probing, Denial-Of-Service, REMOTE-TO-LOCAL, and USER-TO-ROOT are the 4 types into which these 24 groups can be grouped. In a similar fashion, the test data is also labeled as either normal or as one of the attacks pertain to the above mentioned four attack groups. It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not present in the training data. This makes the intrusion detection task more realistic [16]. LIBSVM [15],a java application, is utilized for the experiments with the SVM. For doing experiments with the decision trees and the naive Bayes classifier carried out using Weka [17]. For the process of data formatting and implementing the ordered approach we created a new java application.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In all the experiments detection is done, on both the normal as well as the anomalous connections for training the model. Experiments are done by using a PC processing with Intel(R) Core(TM) 2, CPU 2.4 GHz, and 2-GbyteRAM utilizing the similar conditions. Core interest lies, not in the time required for training of the model, but in the test time efficiency on which the real-time performance of the system is based upon. We identified that the system is highly proficient during the time of testing. When every ordered level features are taken into account, it was observed that around 60000 milliseconds was taken to test all around 26000 attacks and further it was decreased to around 20000 milliseconds when feature selection was done and the Ordered Approach was used. Further details will be provided after evaluating the results later.

We provide the Precision, Recall, and F-Value and not the accuracy alone because it is easy to attain very high accuracy by conscientiously choosing the size of the sample. As per the dataset opted, it is identified that the number of instances for the USER-TO-ROOT, Probes, and REMOTE-TO-LOCAL attacks is very low. So, the system can be prejudiced and can attain a precision of more than 99 percent for USER-TO-ROOT attacks [100],if at all accuracy is utilized to evaluate and test the perform the execution of the system. Whatsoever, Precision, Recall, and F-Value are independent of the size of the training and the test samples. They are defined as follows:

$tf_p = t_p + f_p$	$t_p f_n = t_p + f_n$	$RP^* = R * P$
$P = t_p * \frac{1}{tf_p}$	$R = t_p * \frac{1}{t_p f_n}$	$RP^+ = R + P$
		$F = \frac{(1+\psi^2) * RP^*}{(0+\psi^2) * RP^+}$

Here and indicate the number of True-Positives, False-Positives, and False-Negatives, respectively, and represents precision, represents recall and represents . The relative significance of vs and is generally set to 1. Normal, Probe, Denial-Of-Service, REMOTE-TO-LOCAL, and USER-TO-ROOT are the various groups into which we divide the trained data and also we divided the test data alike. 10 experiments for each attack class are done after haphazardly choosing data corresponding to that attack class and normal data only. It is explained here with a an example. In order to check the Probe attacks, training and testing of the system is done using only Probe attacks and normal data.

Even the Denial-Of-Service, REMOTE-TO-LOCAL, and USER-TO-ROOT are not used data in this process. Excluding them provides system to better acquire the knowledge of features for Probe attacks and normal events. Such a system when it is used online, attacks such as Denial-Of-Service can either be seen as normal or as Probes. Denial-Of-Service attacks, if at all they are tracked as normal, they are anticipated to be detected as attack at other ordered levels in the system. On the other hand if the Denial-Of-Service attacks are tracked as Probe, it is perceived as a gain as the attack is tracked at an early stage. Going by the same fashion, the probe attacks that are not tracked at the Probe level may be detected at further ordered levels. This is the reason for having independent models for all the 4 classes, which are then trained separately with different features to find attacks which confine to a specific group. The best, the average, and the worst cases are reported here.

A.Building phases for Ordered System:

Of the 2 experiments conducted, the aim of the first one is to observe the precision of SVM for intrusion detection and then compare with the other techniques which perform well. All the 41 features are utilized in training the systems and also feature selection is not taken into account. It was noticed that SVM accomplish USER-TO-ROOT attacks in a better way where as with the decision trees the same is the case with Probes and REMOTE-TO-LOCAL. The variation in attack detection precision for Denial-Of-Service is unimportant. It is noticed that cause for better execution of decision trees is that they do feature selection. This is the factor which induced us to bring out feature selection by choosing a minute set of features out of the 41, for all the attack groups, in the second experiment.

The same experiment is done with decision trees and naive Bayes and results are cross checked. We designated the integrated models as Ordered SVM, ordered decision trees, and ordered naive Bayes, respectively for better understanding, the outcomes of both the experiments together. Detecting Probe Attacks comprising all the features, we have haphazardly chosen in and around 10,000 normal records and all the Probe records from the training data. Then every normal and Probe recorded that is taken from the test data is engaged for testing. Thus, 15,000 training instances and 64,759 test instances are obtained.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

From the dataset opted for experiments it is found that around 65X1000 test instances are labeled during a time interval of around 15000 milliseconds. From this it can be inferred that the decision trees are more efficient when compared to that of the SVM and the naive Bayes. The obvious reason is that they have a small tree structure, often with very few decision nodes, which renders it highly efficient. Other advantage with the decision trees is that the attack detection accuracy is also higher, for that they choose the most significant features during tree construction. However, proceeding further we notice that, executing feature selection, the system gets conspicuous accuracy and efficiency improvement. Finding Probe Attacks by using Feature Selection the same set of instances are utilized for this experiment too. But now we have used feature selection for this experiment. Table 3 illustrates the outcomes of this experiment.Now the total around 65X1000 test instances are labeled in a time span of around 2000 milliseconds. This illustrated that, in the detection of probes, HFO-ANIDS with SVM classifier executed better and quicker than the previous one. Since the number of features that are utilized in normal decision trees and in the ordered decision trees is very nearly the same, the end result is the similar efficiency in both the cases and so it is concluded that there is no gain for the ordered decision trees when time is considered.

So it can be understood that the Recall and hence the F-Value for the ordered naive Bayes decreases extensively. This behavior is explained here. The classification accuracy regarding naive Bayes usually gets better as the number of features increases. But, the evaluation becomes capricious as the number of features increases to a very large scale. Hence naïve Bayes are rendered useful, when all the 41 features are utilized, their classification precision decline when the number of features is reduced to five. Henceforth we can summarize that in detecting Probe attacks HFO-ANIDS with SVM is preferred.In first phase choosing haphazardly 20,000 normal records and around 4,000 Denial-Of-Service records from the training data and then we have utilized every normal and Denial-Of-Service record from the test data for testing. Thus 24,000 training instances and 290,446 test instances are listed. 4,290,446 test instances were labeled in a testing time span of 64.42 seconds. From the observations it is observed decision trees have a slight edge when test time efficiency is considered, in contrast to the observations that all the 3 methods employed possess analogous attack detection accuracy.

Here the feature selection with EQPSO is performed using the data that is utilized in the previous experiment. It is noticed that 290,446 test instances were labeled in 15.17 seconds. Only a moderate betterment was observed when compared with the previous experiment. It is found that HFO-ANIDS is better option when testing time is considered. It can also be noticed that there is a tiny increase in the detection precision when feature selection is done, but this increase is imperceptible. The perceptible gain is seen in the reduced time for testing, which subsides by phases in hierarchy. In second phase experiment 1,000 normal records and all the REMOTE-TO-LOCAL records from the training data as the training data for detecting REMOTE-TO-LOCAL attacks are chosen. Then all the normal and REMOTE-TO-LOCAL records from the test data are utilized for testing purpose. In all, 2,000 training instances and 76,942 test instances are used. From the experiments, it can be noticed that more than a lakh test instances were labeled in less than 18000 milliseconds. Observations also reveal that though the decision trees have a higher F-Value, considering the count of the number of false alarms, it can be inferred that SVM execute better and have high Precision than that of the decision trees and the naive Bayes.

In next phase particular experiment feature selection is done in order to detect REMOTE-TO-LOCAL attacks. Considering the observations, it can be noticed that 76,942 instances are tested in a time span of 5.96 seconds. It can also be noticed that, the Ordered SVM rendered better performance than the SVM (left over only 34%), ordered decision trees (increase is approximately more than double), decision trees (1/4 times), ordered naive Bayes (increase is approximately 2.5 times), and naive Bayes (increase is approximately 2.5 times) and therefore they are the pick of the choices for detecting the REMOTE-TO-LOCAL attacks. Though the Ordered SVM takes a little more time, it doesn't matter as higher detection accuracy is obtained. In this experiment 1,000 normal records which are chosen haphazardly and every USER-TO-ROOT record from the training data is taken as the training data to detect the User to Root attacks. Along with them all the normal and USER-TO-ROOT records from the test data are also utilized for testing purpose. In all there are 1,000 training instances and 60,661 test instances. From the experiments, it is noticed that 60,661 test instances are labeled in a time span of 13.45 seconds. It is observed that the SVM have advantage over the other two methods.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The of SVM 1.5 times more than that of decision trees and in the case of naive Bayes it is more than 6 times. The advantage of SVM is that they can be utilized to reliably detect The USER-TO-ROOT attacks, which in general, are very perplexing to detect and most of the contemporary intrusion detection systems fail to detect these with acceptable reliability.In the present experiment, the instances that were used in the earlier experiment were only used. The difference is that feature selection is executed here. From the experiments, it can be noticed that around 70000 test instances were labeled in less than 3000 milliseconds. The inference is that Ordered SVM is the most valuable choice when it comes to the detecting of the USER-TO-ROOT attacks and are far better than SVM (an increase of about 8% percent), ordered decision trees (approximately 1/30 times increase), decision trees (approximately 184% increase), ordered naive Bayes (approximately 1/3 times increase), and naive Bayes (approximately 6.8 times increase).Further it can also be observed that the attack detection capability also increases for both the decision trees and the naive Bayes.

It is also clear from the results that the correctness of Ordered SVM is notably higher for the USER-TO-ROOT, REMOTE-TO-LOCAL, and the Probe attacks, but he difference in accuracy is, however, inconspicuous for the Denial-Of-Service attacks. It is also pellucid that irrespective of the method employed and especially for the SVM, by executing feature selection, the time required for training and testing the system is reduced to a great extent. Further it can be noticed that increase in detection accuracy is not perceptible in the case of the ordered decision trees and the ordered naive Bayes for the Denial-Of-Service group of attack. Their accuracy of detection also decreases in the case of the Probe and REMOTE-TO-LOCAL attacks while it increases for the USER-TO-ROOT attacks. Whatsoever, it was found that in all the cases, when a small set of specific features for training are utilized, the Ordered SVM has executed strikingly better than any other one.

B.HFO-ANIDS in practice:

In general, the category of an attack is not known to us. We are engaged in knowing the attack category only when the system detects an event as anomalous. As every layer in a Ordered Approach is trained to detect only a specific category of attack, this approach helps to enhance the attack and also to recognize the type of the attack.

So, if an attack is recognized at the USER-TO-ROOT layer, it is very plausible that the attack is of "USER-TO-ROOT" type, thus facilitating to execute quick recovery and take precautions to prevent such attacks. Fig. 4 shows the real-time system indication. In this experiment, we combine the4 models (along with feature selection) of section 6.1 in order to evolve the final system and the data utilized is the same that was used for training the individual models the earlier experiments, except for the fact the data in the test set is relabeled either as normal or as attack and all the data from the test set is passed though the system starting from the first layer. A connection which is identified as an attack by the layer 1 is blocked and labeled as "Probe." The events which are labeled as "Normal" are only allowed to go to the next layer. Similar process is redone at the next ordered levels where an attack is blocked and labeled as "Denial-Of-Service," "REMOTE-TO-LOCAL," or "USER-TO-ROOT" at hierarchy second phase, third phase, and last phase , respectively .All the experiments are carried out 10 times and their average is disclosed.

It is noticed that FPO-ANIDS with SVM classifier is able to detect most of the "Probe" (almost 100%), "Denial-Of-Service" (left only 0.90%), and "USER-TO-ROOT" (left only 10.03%) and at each level, a very few false alarms are given by them. It can also be noticed that the system is also able to detect "REMOTE-TO-LOCAL" attacks (reliability observed around 30%), which is a good improvement when compared with the previously reported. By observing confusion matrix, it is revealed that only 30% of Denial-Of-Service attacks left over from labeling as Denial-Of-Service during testing. But, here it is very crucial to understand that the precision for detecting Denial-Of-Service attacks is almost 100% because the left over is only 2.6%. The reason is that 25.50 percent of the Denial-Of-Service attacks are already traced at the first phase, though our system identifies them as possible probe attacks. This is an interesting act to observe that the attacks are identified in first phase itself. It is also significant to note that most of the "USER-TO-ROOT" attacks are reported as REMOTE-TO_LOCAL in third level itself, because the features reflecting the nearness with REMOTE_TO_LOCAL. The rest suspicious activities traced as USER-TO-ROOT at last phase. As of the experiments carried out on dataset provided by DARPA, we can conclude that the ordered order can be in the sequence of DoS, probe, R2L and U2R attack analysis, hence most of the probe attacks that are resembling the DoS activity can be identified at first phase itself.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

For confining the attack traffic to the starting hierarchy in the system, Ordered approach is very powerful. Experiments are executed even without implementing the Ordered Approach. In this case only a single system is considered which is trained with two classes (normal and attack), all the Probes, Denial-Of-Service, REMOTE-TO-LOCAL, and USER-TO-ROOT attacks are labeled as "attack." Experiments both with and without feature selection are also done. For the process of feature selection, 21 features are taken into consideration, which are selected by applying the union operation on the feature sets of all the four attack types. These results are then compared with the Ordered Approach.It can be noticed that a system that implements a Ordered Approach with feature selection is more efficient and more accurate in detecting attacks, especially with respect to the USER-TO-ROOT, the REMOTE-TO-LOCAL, and the Probes.It is very important to understand that the time should be considered in relative terms rather than absolute terms this is for the comfortable utilization of the scripts. However, in real environment, high speed can be achieved by executing the entire system in languages with proficient compilers. An example is the "C Language." In addition, pipelining can be put into effect in multicourse processors, where each core may represent a single level, and because of pipelining; multiple I/O operations can be replaced by a single I/O operation providing very high speed of operation.

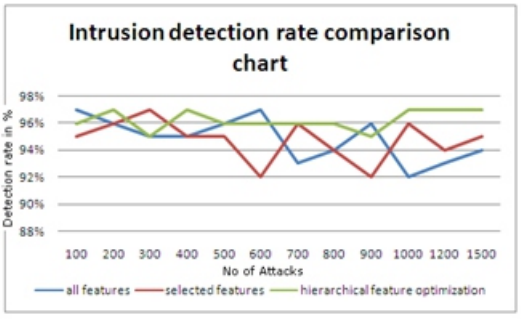

Figure 3: comparison chart for attack detection rate with no feature selection, feature selection, ordered feature optimization.

As of the results it is evident that prediction rate is stable for HFO-ANIDS, and performed better when compared with traditional feature selection approach and approach that use all features. Process time for traditional feature optimization and proposed ordered feature optimization reflecting nearness in fig 5, but when consider the false alarming rate, the proposed HFO-ANIDS is significantly better than the traditional feature optimization models. Another significant advantage of proposed model is that determines the type of attack.

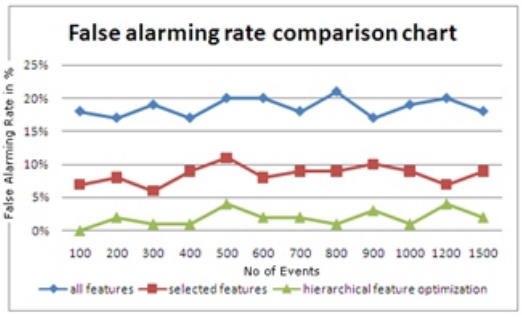

Figure 4: False alarming rate comparison between no feature selections, feature selection, ordered optimization.

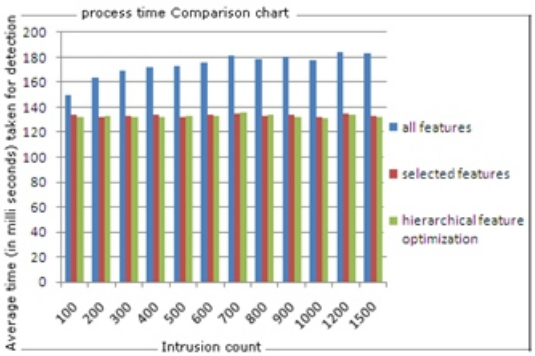

Figure 5: Time taken for attack detection between "no feature selections", "feature selection", "Ordered feature optimization".

Conclusion:

Overall the two problems of Accuracy and Efficiency for creating strong and competent trespass detection systems were discussed in this paper. We have identified that Ordered Feature optimization for ANIDS with Support Vector Machine (SVM) classifier is competent enough in developing the attack detection rate and reducing the False Alarm Rates(FAR). For every competent trespass detective system a low False Alarm Rate is necessary. In addition to this the choosing of features and using the Ordered Approach effectively lessen the time needed to train and test the method. Though the relational data is utilized in experiments, it is proved that SVM, a machine learning techniques helps in finding the attacks efficiently and they prove that other models utilized in the process with relational data are not so capable. To state this method is more competent in identifying the REMOTE-TO-LOCAL and USER-TO-ROOT attacks than other methods we tested this methods which proved by showing in the increase of 1.5 times for the REMOTE-TO-LOCAL and the USER-TO-ROOT in finding the attacks.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Furthermore it is also discussed the significance of the HFO-ANIDS in real practices. We introduced a mathematical model to optimize the QPSO. At last is stated through this paper that HFO-ANIDS has more benefits that the number of ordered phases are adaptive and can adjust based on the environment in which the system is used, providing ease to the network administrators.

References:

[1]. Stolfo, J., Wei, F., Lee, W., Prodromidis, A., Chan, P.K.: Cost-based Modeling and Evaluation for Data Mining with Application to Fraud and Intrusion Detection. Results from the JAM Project by Salvatore (1999)

[2]. Banzhaf, W., Nordin, P., Keller, E.R., Francone, F.D.: Genetic Programming: An Introduction on the Automatic Evolution of Computer Programs and its Applications. Morgan Kaufmann Publishers, Inc (1998)

[3]. Steinberg, D., Colla, P.L., Kerry.: MARS User Guide. Salford Systems, San Diego (1999)

[4]. Joachims, T.: Making Large-Scale SVM Learning Practical. LS8-Report, University of Dortmund (2000)

[5]. Joachims, T.: SVMlight is an Implementation of Support Vector Machines (SVMs) in C. Collaborative Research Center on Complexity Reduction in Multivariate Data (SFB475), University of Dortmund (2000)

[6]. Vladimir, V.N.: The Nature of Statistical Learning Theory. Springer (1995)

[7]. Sung, A.H., Mukkamala, S., Lassez, J-L., and Dawson, T.: Computationally Intelligent Agents for Distributed Intrusion Detection System and Method of Practicing. United States Patent Application No: 10/413,462 (Pending) (2003)

[8]. Mukkamala, S., Sung, A.H., Abraham, A.: Distributed Multi-Intelligent Agent Framework for Detection of Stealthy Probes, Third International Conference on Hybrid Intelligent Systems, Design and Application of Hybrid Intelligent Systems, IOS Press. (2003) 116-125

[9]. Mukkamala, S., Sung, A.H.: A Comparative Study of Techniques for Intrusion Detection. Proceedings of 15th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society Press; (2003) 570-579

[10]. Kendall, K.: A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. Master's Thesis, Massachusetts Institute of Technology (1998)

[11]. Webster, S.E.: The Development and Analysis of Intrusion Detection Algorithms. Master's Thesis, Massachusetts Institute of Technology (1998)

[12]. Brameier, M, Banzhaf, W.: A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining. IEEE Transactions on Evolutionary Computation; Vol.5(1). (2001) 17-26

[13]. Sung, A.H., Xu, J., Ramamurthy, K., Chavez, P., Mukkamala, S., Sulaiman, T., Xie, T.: Static Analyzer for Vicious Executables (SAVE). Presented in Work-in-progress Section of IEEE Symposium on Security and Privacy (2004)

[14]. Sung, A.H., Xu, J., Chavez, P., Mukkamala, S.: Static Analyzer for Vicious Executables (SAVE). To appear in the Proceedings of 20th Annual Computer Security Applications Conference, ACSAC (2004)

[15] KDD Cup 1999 Intrusion Detection Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 2010.

[16] Overview of Attack Trends, http://www.cert.org/archive/pdf/attack_trends.pdf, 2002.

[17] Y.-S. Wu, B. Foo, Y. Mei, and S. Bagchi, "Collaborative Intrusion Detection System (CIDS): A Framework for Accurate and Efficient IDS," Proc. 19th Ann. Computer Security Applications Conf. (ACSAC '03), pp. 234-244, 2003.

[18]. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005).

[19]. Tavallaee M.E, Bagheri W. Lu and Ghorbani A. (2009), "A Detailed Analysis of the KDD CUP 99 Data Set", Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 53-58.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[20]. Xu, X.: Adaptive Intrusion Detection Based on Machine Learning: Feature Extraction, Classifier Construction and Sequential Pattern Prediction. International Journal of Web Services Practices 2(1-2), 49–58 (2006).

[21]. Li, Y., Guo, L.: An Active Learning Based TCM-KNN Algorithm for Supervised Network Intrusion Detection. In: 26th Computers & Security, pp. 459–467 (October 2007)

[22]. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993).

[23]. "Nsl-KDD data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/NSL-KDD.

[24]. Panda M. and Patra M.R (2008), "A Comparative study of Data Mining Algorithms for Network Intrusion Detection", Proceedings of the 1st Conference on Emerging Trends in Engineering and Technology, pp. 504-507, IEEE Computer Society, USA.

[25]. Langley P. and Simon H. A (1995), "Applications of machine learning and rule induction", Communications of the ACM, Vol.38, No. 11, pp. 55–64.

[26]. Amor N.B, Benferhat S. and Elouedi Z (2004), "Naïve Bayes vs. Decision Trees in Intrusion Detection Systems", Proceedings of 2004, ACM Symposium on Applied Computing, pp. 420-424.

[27] Hansman, S. & Hunt, R., 2005. A taxonomy of network and computer attacks.Computers Security, 24(1), p.31-43. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167404804001804.