International Journal of Networked and Distributed Computing, Vol. 2, No. 2 (April 2014), 78-90

An Ensemble Approach for Cyber Attack Detection System: A Generic Framework

Shailendra Singh
Member, IEEE
National Institute of Technical Teacher's Training and Research
Bhopal, India
shailendra@ieee.org
www.nitttrbpl.ac.in

Sanjay Silakari
Member, IEEE
Rajiv Gandhi Technological University, Bhopal, India
ssilakari@rgtu.net
www.rgpv.ac.in

Abstract

Cyber attack detection is based on assumption that intrusive activities are noticeably different from normal system activities and thus detectable. A cyber attack would cause loss of integrity, confidentiality, denial of resources. The fact is that no single classifier is able to give maximum accuracy for all the five classes (Normal, Probe, DOS, U2R and R2L). We have proposed a Cyber Attack Detection System (CADS) and its generic framework, which performs well for all the classes. This is based on Generalized Discriminant Analysis (GDA) algorithm for feature reduction of the cyber attack dataset and an ensemble approach of classifiers for classification of cyber attacks. The ensemble approach of classifiers classifies cyber attack based on the union of the subsets of features. Thus, it can detect a wider range of attacks. The C4.5 and improved Support Vector Machine (iSVM) classifiers are combined as a hierarchical hybrid classifier (C4.5-iSVM) and an ensemble approach combining the individual base classifiers and hybrid classifier for best classification of cyber attacks. The experimental results illustrate that the proposed Cyber Attack Detection System is having higher detection accuracy for the all classes of attacks with minimize training, testing times and false positive alarm.

Keywords-Generalized Discriminant Analysis improved Support Vector Machine, C4.5, Cyber Attack Detection System, Hybrid system, Ensemble approach

1. Introduction

Attacks on computer infrastructures are becoming an increasingly serious problem. Network cyber detection is an important aspect of computer network security. Computer security is defined as the protection of computing systems against threats to confidentiality, integrity, and availability. Confidentiality means that information is disclosed only according to policy, integrity

means that information is not destroyed or corrupted and that the system performs correctly and availability means that system services are available when they are needed. Computing systems refer to computers, computer networks, and the information they handle. These threats and others that are likely to appear in the future have lead to the design and development of cyber attack detection systems. According to webopedia [1] cyber attack detection system inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.

In this paper, we present Cyber Attack Detection System (CADS) and its generic framework, which perform well for all the classes of attack. In this framework, we have used four tiers architecture to enhance the adoptability of the cyber attack detection system. The first tier is dedicated to data collection and preprocessing of the data. The Second tier is meant for the feature extraction technique, third tier is dedicated for classification of cyber attacks and fourth tier is dedicated for user interface for reporting the events. The motivation for using the ensemble approach is to improve the accuracy of cyber attack detection system when compared to using individual approaches. The ensemble approach combines the best results from the different individual classifier resulting in more cyber attack detection accuracy and layered framework enhanced the adaptability of the system.

The Generalized Discriminant Analysis (GDA) [2] algorithm is used for feature reduction of KDDCUP2009 dataset. The ensemble classifier classifies the cyber attack on reduced features of dataset. The ensemble approach of classifiers classifies cyber attack based on the union of the subsets of features. Thus, it can detect a wider range of attacks. The C4.5 [3] and improved Support Vector Machine (iSVM) classifiers are combined as a hierarchical hybrid classifier (C4.5-iSVM) and an ensemble approach combining the individual base classifiers (C4.5 and iSVM) and hybrid classifier (C4.5-iSVM) for best classification of cyber attack. The results of SVM classification only depend on support vectors (SVs) in the dataset, which is always a relatively small part of the whole training set. Thus, it is capable of handling large training dataset. The experimental results illustrate that Cyber Attack Detection System is having higher detection accuracy (minimize the false positive alarm) for the all classes of attacks and minimize training and testing times of the classifiers.

2. Related Work

Earlier studies have utilized a rule-based approach for cyber attack detection, but had a difficulty in identifying new attacks or attacks that had no previously describe patterns[4][5]. Lately the emphasis is being shifted to learning by examples and data mining paradigms. Neural networks have been extensively used to identify both misuse and anomalous patterns [6] [7]. Recently, kernel based methods, SVMs and their variants [8] [9][10] are being proposed to detect cyber attacks. It has been already proved that the feature extraction (reduction) technique enhanced the performance of the classifiers [11][12]. Although above mentioned classifiers have improved cyber attack detection accuracy but the fact is that no single classifier is able to give maximum accuracy for all the five classes (Normal, Probe, DOS, U2R and R2L).

DIDS [13] is a distributed intrusion detection system consisting of host managers and LAN managers doing distributed data monitoring and sending notable events to the director. The centralized director then analyzes these events to determine the security state of the system as a whole. The centralized director is clearly the bottleneck to the distributive approach of DIDS. As there is only one level of hierarchy with all host and LAN managers reporting to a single director, it lacks scalability. EMERALD [14] is a framework for performing distributed intrusion detection. It employs monitors at the levels of hosts, domains and enterprises to form an analysis hierarchy. It uses a subscription-based communication scheme both within and between monitors. But the inter-monitor subscription scheme is hierarchical thus limiting access to the events or results from the layer immediately below. GrIDS [15] construct activity graphs representing hosts and network activity. It models an organization as a hierarchy of departments and hosts.

AAFID [16] is a framework for a distributed intrusion detection system that employs autonomous agents at the lowest level for data collection, analysis, transceivers and monitors at the higher levels of the hierarchy for controlling the agents and obtaining a global view of activities. It suggests the use of filters within hosts at data selection and abstraction layers, providing a subscription-based service to agents. The Common Intrusion Detection Framework (CIDF) [17] goes one step further as it aims to

enable different intrusion detection and response components to incorporate and share information and resources in a distributed environment. The Coordinated And Response Distributed System (CARDS) [18] aims at detecting distributed attacks that cannot be detected using data collected from single location. CARDS decompose global representations of distributed events indicating the attacks. Bernardes and dos Santos Moreira [19] have proposed a hybrid framework with partially distributed decision making under the control of a centralized agent manager. Agent-based hierarchical architectures include the Intelligent Agents for Intrusion Detection [20] with a centralized data warehouse at the root, data cleaners at the leaves, and classifier agent in between.

In the above mentioned framework for cyber attack detection system, we have added one more module called Feature Extraction Module in our generic framework which is essential for dimensionality reduction of huge data for real time cyber attack detection.

3. Cyber Attack Dataset

These audit data of normal activities are generated by the MIT Lab [21] through simulating activities in a real information system used by the U.S. Air Force. Intrusions are simulated on the background of normal activities. Because intrusive activities in this small sample data are very limited, we use audit data of normal activities only from this sample data. For each TCP/IP connection, 41 various quantitative (continuous data type) and qualitative (discrete data type) features were extracted among the 41 features, 34 features are numeric and 7 features are symbolic. The data contains 24 attack types that could be classified into four main categories:

- DOS: Denial Of Service attack.
- R2L: Remote to Local (User) attack.
- U2R: User to Root attack.
- Probing: Surveillance and other probing.

3.1 *Denial of service Attack*

Denial of service (DOS) is class of attack where an attacker makes a computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate user access to a machine.

Table 1. Denial of service attack (DOS)

Attack type	Service	Mechanism	Effect of the attack
Apache2	http	Abuse	Crashes httpd
Back	http	Abuse	Slows down server response
Land	http	Bug	Freezes the machine
Mail bomb	N/A	Abuse	Annoyance
SYN flood	TCP	Abuse	Denies service on one or more ports
Ping of death	Icmp	Bug	None
Process table	TCP	Abuse	Denies new processes
Smurf	Icmp	Abuse	Slows down the network
Syslogd	Syslog	Bug	Kills the Syslogd
Teardrop	N/A	Bug	Reboots the machine
Udpstorm	Echo	Abuse	Slows down network

3.2 *Remote to Local (User) Attacks*

A remote to local (R2L) attack as given in the table 2 is a class of attacks where an attacker sends packets to a machine over network, then exploits the machine's vulnerability to illegally gain local access to a machine.

Table 2. Remote to user attack (R2L)

Attack type	Service	Mechanism	Effect of the attack
Dictionary	telnet, rlogin, pop, ftp, imap	Abuse features	Gains user access
Ftp-write	ftp	Misconfig	Gains user access
Guest	telnet, rlogin	Misconfig	Gains user access
Imap	Imap	Bug	Gains root access
Named	Dns	Bug	Gains root access
Phf	http	Bug	Executes command as http user
Sendmail	Smtp	Bug	Executes commands as root
Xlock	Smtp	Misconfig	Spoof user to obtain password
Xnsoop	Smtp	Misconfig	Monitor key strokes remotely

3.3 *User to Root Attacks*

User to root (U2R) attacks as given in the table 3 is a class of attacks where an attacker starts with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.

Table 3. User to root attack (U2R)

Attack type	Service	Mechanism	Effect of the attack
Eject	User session	Buffer overflow	Gains root shell
Ffbconfig	User session	Buffer overflow	Gains root shell
Fdformat	User session	Buffer overflow	Gains root shell
Loadmodule	User session	Poor environment sanitation	Gains root shell
Perl	User session	Poor environment sanitation	Gains root shell
Ps	User session	Poor Temp file Managment	Gains root shell
Xtern	User session	Buffer overflow	Gains root shell

3.4 *Probing*

Probing is class of attacks as given in the table 4 where an attacker scans a network to gather information or find known vulnerabilities. An attacker with map of machine and services that are available on a network can use the information to notice to exploit.

Table 4. Probe attacks

Attack type	Service	Mechanism	Effect of the attack
Ipsweep	Icmp	Abuse of feature	Identifies active machines
Mscan	Many	Abuse of feature	Looks for known vulnerabilities
Nmap	Many	Abuse of feature	Identifies active ports on a machine
Saint	Many	Abuse of feature	Looks for known vulnerabilities
Satan	Many	Abuse of feature	Looks for known vulnerabilities

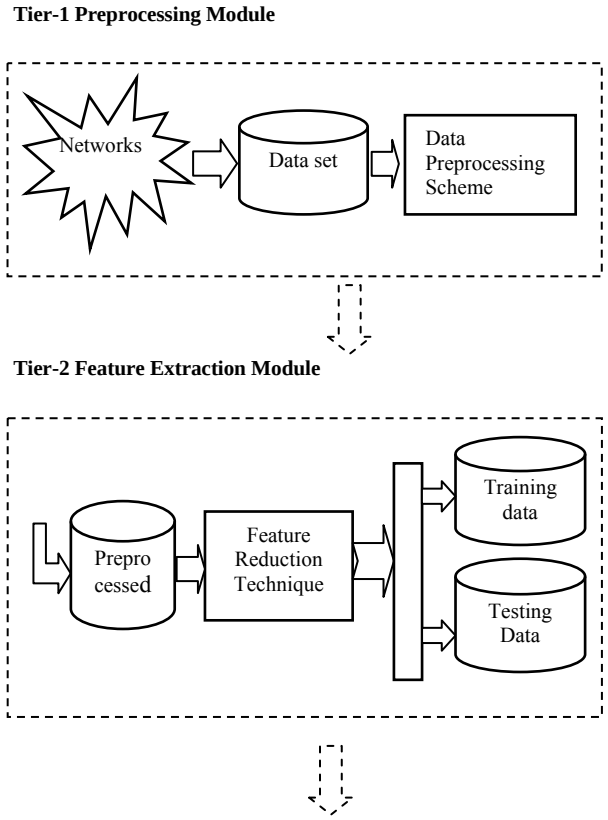
4. Architecture of Proposed Framework For Cyber Attack Detection System

In our proposed generic framework for cyber attack detection system, we used four tiers architecture for cyber attack detection system.

The first tier is dedicated to data collection and preprocessing of the data because the TCP dump dataset is not directly used. It is not in proper format hence our algorithm converts it into desired format which is fed to next tier. The second tier is meant for the feature extraction technique. As we know that all the features of the dataset is not contributed much in cyber attack detection process due to this there is curse of dimensionality therefore, we need to remove the redundant features from the dataset in order to improve the performance of the classifiers. In this tier, we use GDA feature reduction algorithm for extracting suitable features.

The third tier is fully dedicated for classification of cyber attacks. In this tier, we used ensemble approach of classifiers to get high detection accuracy. The last fourth tier is used for user interface for reporting the attacks. The key effort is directed towards making the generic framework for cyber attack detection system. Some of the important features of this framework are given below

Generic architecture: We have divided our proposed frame work into four tiers each tier independent to other tiers. We can select appropriate technique according to our requirement because complete architecture is modular.



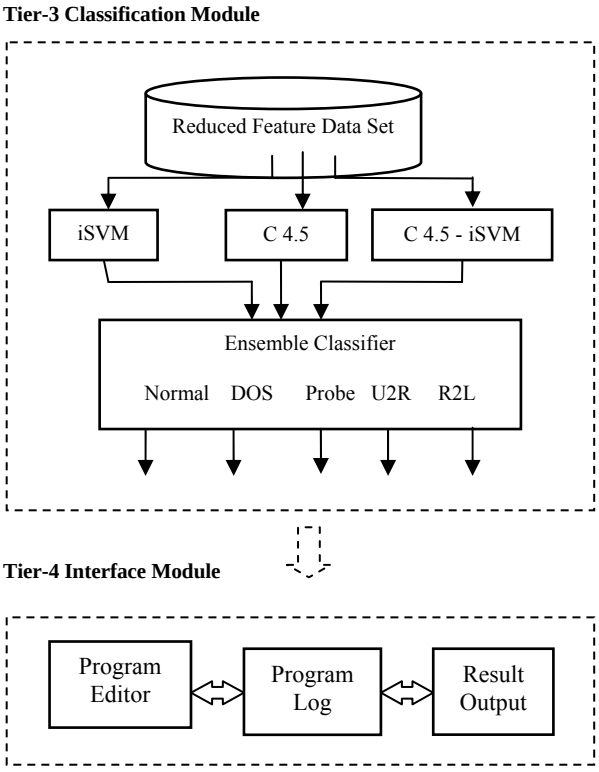


Fig. 1. Architecture of proposed generic framework for Cyber Attack Detection System.

Portability: Portability of the cyber attack detection system with respect to operating systems and computer architecture is an also an issue. Java as an interpreted language is used to provide portability for cyber attack detection system.

Efficiency: In this framework, we have used collection of classifiers which is having high detection rate of specific classes in order to enhance overall efficiency of the system.

Upgradability: A cyber attack detection system based on a component-based architecture available in modular form which enhances the upgradability of the system. New features can easily be added to such a system.

Scalability: In this architecture we have used improved support vector machine which is capable of handle a very large data with same accuracy.

4.1 *Tier-1:Preprocessing Module*

In this section, we describe our data mining algorithms, and illustrate how to apply these algorithms to generate detection models from audit data. Here, audit data refers to (pre-processed) time stamped audit records, each with a number of features (fields).

4.2 *Tier-2:Feature Extraction Module*

Cyber attack detection systems have become important and widely used tools for ensuring network security. Since the amount of audit data that a cyber attack detection system needs to examine is very large even for a small network, classification by hand is impossible. A cyber attack detection system must therefore reduce the amount of data to be processed. This is extremely important if real-time detection is desired. Feature extraction algorithm applies a mapping of the multidimensional space into a space of lower dimensions. Feature extraction [11][12] includes feature construction, space dimensionality reduction, sparse representations, and feature selection. All these techniques are commonly used as pre processing to machine learning and statistical tasks of prediction, including pattern recognition and regression. Although such problems have been tackled by researchers for many years, there has been recently a renewed interest in feature extraction. A number of new applications with very large input spaces critically need space dimensionality reduction for improving the efficiency of the classifiers.

The Generalized Discriminant Analysis algorithm is used for feature reduction of cyber attack dataset. Due to the large variations in the attack patterns of various attack classes, there is usually a considerable overlap between some of these classes in the feature space. In this situation, a feature transformation mechanism that can minimize the between-class scatter is used. The Generalized Discriminant Analysis GDA [2] is a method designed for nonlinear classification based on a kernel function ϕ which transform the original space X to a new high-dimensional feature space $Z: \emptyset: X \rightarrow Z$. The within-class scatter and between-class scatter matrix of the nonlinearly mapped data. We have used GDA for reducing dimensionality of KDDCUP2009 intrusion detection dataset. Each feature vectors is labeled as an attack or normal. The distance between a vector and its reconstruction onto those reduced subspaces representing different types of attacks and normal activities is used for identification.

The between-class scatter matrix and within-class scatter matrix of the nonlinearly mapped data is

$$B^{\emptyset} = \sum_{c=1}^{C} M_c \, m_c^{\emptyset} (m_c^{\emptyset})^T \quad (1)$$

$$W^{\emptyset} = \sum_{c=1}^{C} \sum_{x=X_c} \emptyset(x) \, \emptyset(x)^T \quad (2)$$

Where:

$m_c^{\emptyset}$ is the mean of class X_c in Z and M_c is the number of samples belonging to X_c.

The aim of the GDA is to find projection matrix $U^{\emptyset}$ that maximizes the ratio

$$U_{opt}^{\emptyset} = argmax \frac{|(U^{\emptyset})^T B^{\emptyset} U^{\emptyset}|}{|(U^{\emptyset})^T W^{\emptyset} U^{\emptyset}|} = [u_1^{\emptyset}, \dots \dots, u_N^{\emptyset}] \quad (3)$$

We have to find eigenvalues λ_i and eigenvectors $u^{\emptyset}$ solution of the equation:

$$B^{\emptyset} u_i^{\emptyset} = \lambda_i W^{\emptyset} u_i^{\emptyset} \quad (4)$$

The largest eigenvalue of (4) gives maximum of the following quotient of the inertia:

$$\lambda_i = \frac{B^{\emptyset} u_i^{\emptyset}}{W^{\emptyset} u_i^{\emptyset}} \quad (5)$$

As the eignvectors are linear combinations of Z elements, there exist coefficients α_{ci} $(c = 1 \dots, C; i = 1 \dots .. M_c$;) such that

$$u^{\emptyset} = \sum_{c=1}^{C} \sum_{i=1}^{M} \alpha_{ci} \emptyset(x_{ci}) \quad (6)$$

Where x_{ci} is the i^{th} sample of the class c. The solution is obtained by solving

$$\lambda_i = \frac{\propto^T KDK \propto}{\propto^T KK \propto} \quad (7)$$

Where coefficient vector $\propto = (\propto_c)$, $c = 1 \dots .. C$ is a vector of weights with $\propto_c = (\propto_{ci})$, $i = 1 \dots . M_c$.

Where $\propto_c$ is coefficient of the vector $u^{\emptyset}$ in the class c. and α^T is the transpose of coefficient vector.

the data. The number of classes of KDDCUP2009 dataset is five. Therefore, the optimal number of eigenvectors for the data transformation is equal to four. After feature

The kernel matrix $K(M \times M)$ is composed of the dot products of nonlinearly mapped data, i.e.

$$K = (K_{kl})_{k=1 \dots C, \; l=1 \dots . C} \quad (8)$$

Where

$K_{kl} = (k(x_{ki}, x_{lj}))_{i=1 \dots M_k, j=1 \dots M_l}$ The matrix $D(M \times M)$ is a block diagonal matrix such that

$$D = (D_c)_{c=1 \dots C} \quad (9)$$

Where

The c^{th} on the diagonal has all elements equal to $1/M_c$. Solving the eigenvalue problem yields the coefficient vector $\propto$ which define the projection vectors $u^{\emptyset} \in Z$. A projection of a testing vector x_{test} is computed as

$$(u^{\emptyset})^T \emptyset \, (x_{test}) = \sum_{c=1}^{C} \sum_{i=1}^{M} \propto_{ci} \, k(x_{ci}, x_{test}) \quad (10)$$

The algorithm of proposed Generalized Discriminant Analysis (GDA) technique is given below:

Step1: Compute the matrices K and D by solving the equation (8) and (9),

Step2: Decompose K using eigenvectors decomposition,

Step3: Compute eigenvectors $u^{\emptyset}$ and eigenvalues λ_i of the equation (4),

Step4: Compute eigenvectors $u^{\emptyset}$ using $\alpha_{ci}$from equation (6) and normalize them,

Step5: Compute projections of test points onto the eigenvectors $u^{\emptyset}$from equation (10).

The Linear Discriminant Analysis (LDA) [22] scheme is then applied to the mapped data, where it searches for those vectors that best discriminate among the classes rather than those vectors that best describe

reduction of KDDCUP2009 dataset the reduced features are fed to the classifiers.

4.3 *Tier-3:Classification Module*

The classification module consists of the individual classifiers and a collection of classifiers in order to improve the detection accuracy of the classifier are given below-

4.3.1 Improved Support Vector Machine (iSVM)

The SVM [23] identifies the best separating hyperplane between the two classes of the training samples within the feature space by focusing on training cases placed at the edge of the class descriptors. In this way, not only an optimal hyper plane is fitted, but also less training samples are effectively used; thus high classification accuracy is achieved with small training sets. We construct m SVM model where m is the number of classes. The m^{th}SVM is trained with all of the examples in the m^{th} class with positive labels, and all other examples with negative labels. Thus, given l training data $(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_l, y_l),\ i = 1,2, \ldots\ldots, l,$ where $x_i \epsilon R^l$ and $y_i \epsilon \{1,2,..k\}$ is the class of $x_m$the m^{th} SVM solves the following optimization problem:

$$\Phi(w, b, \xi) = \sum_{m=1}^{k}(w_m \cdot w_m) + C\left(\sum_{i=1}^{\ell}\sum_{m \neq y_i}^{k} \xi_i^m\right)$$
$$s.t.\left(w_{y_i} . \emptyset(x_i)\right) + b_{y_i} \geq (w_m . \emptyset(x_i)) + b_m + 2 - \xi_i^m,$$
$$\xi_i^m \geq 0, i = 1, \ldots\ldots\ \ell\ \ m \epsilon \{1 \ldots\ldots\ldots, k\} \backslash y_i$$
(11)

Where $\emptyset(x_i)$ is nonlinear function that maps x into a higher dimensional space $w, b,$ and ξ are the weight vector, bias and slack variable, respectively. C is constant and determined a priori. Searching for the optimal hyperplane in equation (11) is quadratic programming problem. Minimizing $\frac{1}{2}\sum_{m=1}^{k}(w_m \cdot w_m)$ means that we would like to maximize $2/\|w_m\|$, the margin between two classes of attack data. Where data are not linearly separable, there is penalty terms $C\left(\sum_{i=1}^{\ell}\sum_{m \neq y_i}^{k} \xi_i^m\right)$ which can reduce the number of training errors. The basic concept behind SVM is to search for a balance between the regularization term $\frac{1}{2}\sum_{m=1}^{k}(w_m \cdot w_m)$ and training errors. After solving the equation (11) to get k decision functions

$$f_k(x) = \sum_{i=1}^{\ell} \alpha_i^k\ y_i^k\ K(x, x_i) + b^k \qquad (12)$$

Here, kernel $K(x, x_i)$, is a Gaussian kernel function, α_j^k Lagrange multiplier and b^k is bias of class k. To improve the classification accuracy of the SVM classifier we will modify Gaussian kernel function $K(x, x_i)$ in data dependent way used in iSVM.

The choice of the kernel greatly affects the SVM's ability to classify data points accurately. We modify existing Gaussian kernel according to our need. This modified kernel gives better performance as compared to the original Gaussian kernel.

A nonlinear SVM maps each samples of input space R into a feature space F through a nonlinear mapping $\emptyset$. The mapping $\emptyset$ defines an embedding of S into F as a curve submanifold.

Denote $\emptyset(x)$ the mapped samples of S in the featured space; small vector dx is mapped to:

$$\emptyset(dx) = \nabla\emptyset . dx = \sum_i \frac{\partial}{\partial x^{(i)}} \emptyset(x) dx^{(i)} \qquad (13)$$

Where $\nabla\emptyset = \frac{\partial}{\partial x^{(i)}} \emptyset(x)$.

The squared length of $\emptyset(dx)$ is written as:

$$ds^2 = |\emptyset(dx)|^2 = \sum_{i,j} g_{ij}(x) dx^{(i)} dx^{(j)} \qquad (14)$$

Where

$$g_{ij}(x) = \left(\frac{\partial}{\partial x^{(i)}} \emptyset(x)\right) . \left(\frac{\partial}{\partial x^{(j)}} \emptyset(x)\right), \qquad (15)$$

The dot denoting the summation over index α of $\emptyset$. The $n \times n$
Positive-definite matrix $G(x) = (g_{ij}(x))$ is the Riemannian metric tensor induced in S.

$$g_{ij}(x) = \frac{\partial}{\partial x^{(i)}} \frac{\partial}{\partial x^{(j)}} K(x, x_i) \qquad (16)$$

We can increase the margin or the distances (ds) between classes to improve the performance of the SVM. Taking eq.(14) in to account, this leads us to increase the Riemannian metric tensor around the boundary and to

reduce it around other samples. In view of eq.(16), we can modify the kernel K such that $g_{ij}(x)$ is in large around the boundary.

Modifying kernel based on the structure of the Riemannian geometry. Assume the kernel can be modified as:

$$\widetilde{K}(x, x_i) = p(x)p(x_i)K(x, x_i) \quad (17)$$

is called a conformal transformation of a kernel by factor p(x). We take the kernel function used in SVM is Gaussian Kernel, i.e.:

$$K(x, x_i) = \exp(-\|x - x_i\|^2/\sigma^2) \quad (18)$$

Here, the parameter σ is kernel width. It is proved that the corresponding Riemannian metric tensor is changed into:

$$g_{ij}(x) = \frac{1}{\sigma^2}\delta_{ij} \quad (19)$$

After modifying the kernel Riemannian metric tensor is changed into:

$$\tilde{g}_{ij}(x) = p_i(x)p_j(x) + p^2(x)g_{ij}(x) \quad (20)$$

To ensure that p(x) has large value around the support vector (SV), by the conformal transformation of the Gaussian kernel,

$$p_i(x) = \partial p(x)/\partial x_i \quad (21)$$

For maximum p(x) the value of $p_i(x) = 0$.

In order to ensure that p(x) has large values at the support vector positions, it can be constructed in a data dependent way as:

$$p(x) = \sum_{i \in SV} \alpha_i \exp(-\|x - x_i\|^2/2\tau^2) \quad (22)$$

Where τ is a free parameter and summation runs over all the support vectors. As we see $p_i(x)$ and $p(x)$ are large when x is close to support vectors and those are small when x is for away from SVs then, when x is close to support vectors the $g_{ij}(x)$ around support vectors is increased. So, the spatial resolution around the boundary is enlarged and classification ability of SVM becomes stronger.

$$f_k(x) = \sum_{i=1}^{\ell} \alpha_i^k \, y_i^k \, \widetilde{K}(x, x_i) + b^k \quad (23)$$

We summarized the procedure of the proposed Algorithm as follows:

Step1: Train SVM with primary Gaussian kernel $K(x, x_i)$ to extract the information of SVs, then modify Gaussian kernel K according to the formula (17) and (22).

Step2: Train the SVM with the modified Gaussian kernel $\widetilde{K}$.

Step3: Iteratively apply the above two steps until the best performance is achieved.

4.3.2 C4.5 Classifier

The C4.5 classifier is one of the classification algorithms in data mining. The classification algorithm is inductively learned to construct a model from the pre-classified dataset. Inductive learning means making general assumptions from the specific examples in order to use those assumptions to classify unseen data. The classifier may be viewed as mapping from a set of attributes and X is the vector of their values $\{x_1, x_2, \ldots \ldots x_n\}$ Attribute space is defined as set containing all possible attribute vectors and is denoted by Z. Thus X is element of Z $(X \epsilon Z)$. The set of all the classes is denoted by $C = \{c_1, c_2 \ldots . . c_n\}$. A classifier assigns a class $c \epsilon C$ to every attribute of the vector $X \epsilon Z$. The classifier can be considered as a mapping $f: X \rightarrow C$. This classifier is used to classify the unseen data with a class label.

4.3.3 *Hybrid (C4.5-iSVM) Classifier*

A hybrid classifier uses the approach of integrating C4.5 and improved Support Vector Machine (iSVM) into a single classifier. Each learning model works in different manner and exploits different set of features. Integrating different learning models gives better performance than the individual learning or decision models by reducing their individual limitations and exploiting their different

mechanisms. In a hierarchical hybrid classifier, each layer provides some new information to the higher level. The dataset first pass through the C4.5 and node information is generated. Node information is generated according to the rules generated by the C4.5 classifier. All the dataset records are assigned to one of the terminal nodes, which represent the particular class. This node information along with the original set of attributes is passed through the SVM to obtain the final output.

4.3.4 *Ensemble Classifier*

Several researchers have investigated the combination of different classifiers to form an ensemble classifiers [24][26].

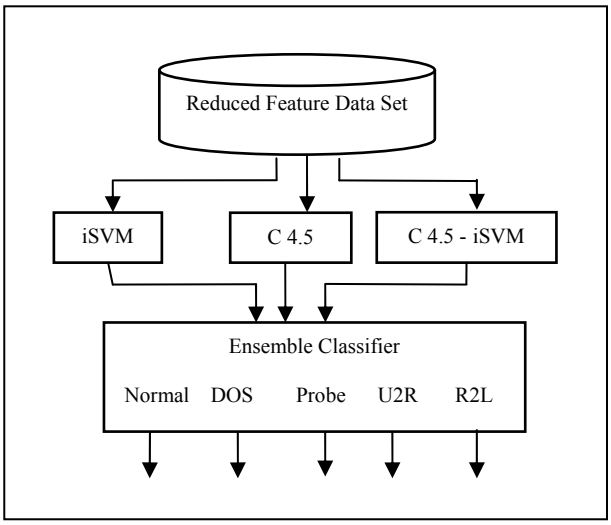


Fig. 2. Architecture of Ensemble Classifiers

An important advantage for combining redundant and complementary classifiers is to increase accuracy and better overall generalization. In this approach, we combine individual C4.5, iSVM and hybrid (C4.5-iSVM) classifier. Empirical observation shows that different classifier provide complementary information about the patterns to be classified. Although for a particular problem one classifier works better than other, a set of misclassified pattern would not necessarily overlap.

This different information combined together yields better performance than individual classifiers. The idea is not to rely on single classifier for decision on cyber attack instead information from different individual classifiers is combined to take the final decision, which is popularly known as the ensemble approach. The effectiveness of the ensemble approach depends on the accuracy and diversity of the base classifiers.

Each classifier is assigned different weights according the performance of the training data. Using these weights and outputs of the classifier, scores were calculated. For example, for class 1 if the C4.5 works best, followed by SVM and C4.5-iSVM classifier, the C4.5 is assigned the highest weight, followed by the C4.5-iSVM and SVM is assigned the lowest weight. Each classifier has different weights depending on their performance on the training data for five different classes. So for a particular data record if all of them have different opinions, their scores are considered and the highest score is declared as the actual output of the ensemble approach. The architecture of the ensemble approach is depicted in Figure 2.

4.4 *Tier-4:User Interface Module*

The most complex and full-featured IDS can be useless if it does not have good mechanisms to allow users to interact with and control it. We have not looked in full detail into the user interface problem, although some issues are mentioned.

A user interface has to interact with a monitor and it has to use the API that the monitor exports to request information and to provide instructions. This separation allows different user interface implementations to be used with an AAFID system. A Graphical User Interface (GUI) could be used to provide interactive access to the IDS, while a command-line based interface could be used in scripts to automate some maintenance and reporting functions. The Graphic User Interface (GUI) of the cyber attack detection system display screen includes three main components: (1) Program Editor, (2) Program Log, and (3) Result Output as shown in Figure 1.

5. Experimental Setup and Results

In our experiments, we perform five-class classification. The KDDCUP2009 dataset is not a normalized dataset. Therefore, it needs preprocessing of dataset before given to

feature reduction algorithm. Logarithmic scaling (with base 10) was applied to the very large features. The KDDCUP2009 dataset consist of 1,25,973 records for training and 25,192 records for testing. The normal data belongs to class 1, probe belongs to class 2, DoS belongs to class 3, user to super-user belongs to class 4, remote to local belongs to class 5. All the experiments were performed on an Intel Xeon with a 2.4 GHz CPU and 2 GB of RAM. We have implemented the proposed framework in Java. We used the KDDCUP2009 dataset To evaluate the performance of our proposed cyber attack detection system. In the training phase the system constructs a model using the training data to give maximum generalization accuracy (accuracy on unseen data). The test data is passed through the constructed model to detect the cyber attack in testing phase.

5.1 *Improved Support Vector Machine (iSVM) Classifier*

Improved Support Vector Machine (iSVM) separate the data into two classes, classification into additional classes by applying one against all (OAA) method. In the OAA method, a set of binary classifiers (k parallel SVMs, where k denotes the number of classes) is trained to be able to separate each class from all others. Then, each data object is classified to the class for which the largest decision value has been determined. Then voting strategy aggregates the decisions and predicts that each data object is in the class with the largest vote.

Table 5. Performance of the improved SVM classifier

Attack Classes	Training time(s)	Testing time(s)	Accuracy (%)
Normal	5.01	0.26	100
Probe	3.11	0.21	99.98
DOS	14.56	2.13	100
R2L	3.02	0.21	85.54
U2R	4.17	1.07	78.61

Note the same training dataset (1,25,973) used for training the iSVM, C4.5 hybrid classifier (C4.5-iSVM) and ensemble classifier and the same testing dataset (25,192) used for testing all classifiers being used to validate the performance. The objective is to separate normal and attack patterns. We repeat this process for all classes. Training is done using the modified Gaussian kernel function; an important point of the kernel function is that it defines the feature space in which the training set examples will be classified. Table 5 summarizes the results of the experiments.

5.2 *C4.5 Classsifier*

We constructed five different classifiers which is binary classifier. The data is portioned into the two classes of Normal and Attack patterns where attack is collection of four classes (Probe, DOS, R2L and U2R) of attacks. The objective is to separate normal and attack patterns. We repeat this process for all the five classes. First a classifier was constructed using the training data and then testing data was tested with the constructed classifiers to classifier to classify the data into normal and attack patterns. Table 6 summarizes the results of the test data. It shows the training and testing time in seconds for each of the five classes and their accuracy.

Table 6. Performance of the C4.5 classifier

Attack Classes	Training time(s)	Testing time(s)	Accuracy (%)
Normal	8.25	1.01	99.36
Probe	4.11	0.85	99.86
DOS	19.08	3.78	96.83
R2L	3.01	0.20	68.67
U2R	3.89	0.69	84.00

5.3 *Hybrid (C4.5-iSVM) Classifier*

The hybrid (C4.5-iSVM) model has two steps for constructing the classifier. The data set first passed through the C4.5 and the node information is generated. Training and test data along with node information is given to the iSVM. The iSVM gives the final output of the hybrid (C4.5-iSVM). The performance of hybrid classifier is shown in Table 4 and 5. Hybrid classifier works better than the individual C4.5 and iSVM classifier.

Table 7. Performance comparison of the three classifiers

Attack Classes	iSVM (%)	C4.5 (%)	Hybrid (C4.5-iSVM)
Normal	100	99.36	100
Probe	99.98	99.86	99.98
DOS	100	96.83	100
R2L	85.54	68.67	85.54
U2R	78.61	84.00	95.00

5.4 *Ensemble Classifier*

We first construct C4.5, iSVM and hybrid (C4.5-iSVM) classifiers individually to obtain good generalization performance. Test data set passed through each individual model and the corresponding output(s) are used to decide the final output. The performance of the ensemble approach gives better performance for detecting U2R and R2L classes of attacks then all three individual models.

Table 8. Performance comparison of the ensemble classifier

Attack Classes	iSVM (%)	C4.5 (%)	Hybrid (C4.5-iSVM)	Ensemble Classifier
Normal	100	99.36	100	100
Probe	99.98	99.86	99.98	100
DOS	100	96.83	100	100
R2L	85.54	68.67	85.54	97.16
U2R	78.61	84.00	84.00	98.26

As shown in Figure 3, the ensemble classifier gives better cyber attack detection accuracy for all the classes of attacks. It is clearly shown in the figure that C4.5 is having low detection accuracy for all the classes.

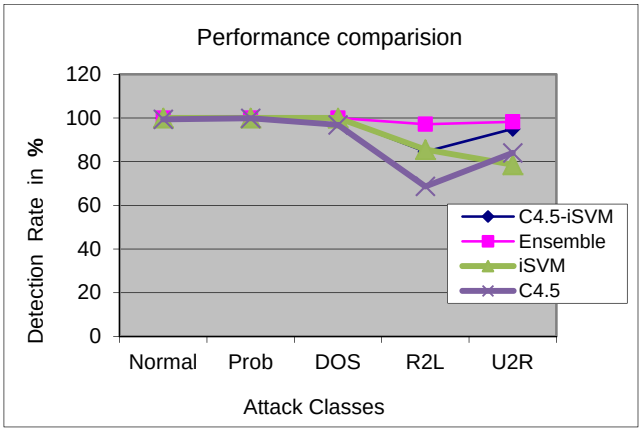


Fig. 3 Performace Comparision of Ensemble Classifier with other classiers

We get improved detection accuracy with iSVM classifier but still it shows low detection accuracy for the classes R2L and U2R class is improved by hybrid (C4.5-iSVM) classifier with this classifier we get 85.54% accuracy for R2L and 95% for U2R classes. In an ensemble classifier we are able to improve the overall accuracy for all the classes (Normal 100%, DOS 100%, Probe 100%, R2L 97.16% and U2R 98.26%).

6. Conclusion

In this paper, we have investigated some new techniques for cyber attack detection system and evaluated their performance based on the benchmark KDDCUP2009 cyber attack data. We have explored C4.5 and iSVM as an cyber attack models. Next, we designed a hybrid C4.5-iSVM model and ensemble approach with C4.5, iSVM and C4.5 – iSVM models as base classifiers. Empirical results reveal that C4.5 gives better or equal accuracy for Normal and Probe classes and the iSVM gives better accuracy for Normal and DOS classes. The hybrid C4.5-iSVM classifier improves accuracy for R2L and U2R classes when compared to individual accuracy of classifiers.

The ensemble classifiers gave the best performance for Probe and R2L classes. The ensemble approach gave 100% accuracy Probe class, and this suggests that if proper base classifiers are chosen 100% accuracy might be possible for other classes too. Finally we propose an ensemble approach with new framework for cyber attack detection system to make optimum use of best performances delivered by the individual base classifier and ensemble classifiers.

References

1. Cyber Attack: http://www.webopedia.com.
2. G.Baudt and F. *Anouar Generalized Discriminant Aanlyis Using a Kernal Approach Neural Computation*, 2000.
3. J.R. Quinlan, C4.5 *Programs for machine learning Morgan Kaufmann* 1993.
4. Anderson D, Lunt TF, Javitz H, Tamaru A,, Valdes A. *Detecting unusual program behavior using the stastistical component of the next-generation intrusion detection expert system* (NIDES). SRI-CSL-95-06, Menlo Park, CA: SRI International; 1995.
5. Mahoney M, Chan PK. *An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection. Sixth International Symposium on Recent Advances in Intrusion Detection*; 2003. pp. 220–37.
6. Mukkamala S, Janowski G, Sung AH. *Intrusion detection using neural networks and support vector machines. Proceedings of Hybrid Information Systems Advances in Soft Computing*, Heidelberg: Springer; ISBN 3790814806, pp.121–38. 2001.
7. Mukkamala S, Sung AH. *Feature selection for intrusion detection using neural networks and support vector machines.* J Transport Res Board Natl Acad, Transport Res Record No 1822; 33–9. 2003.

8. Xiong, Sheng-Wu, Liu Hong-bing, Niu Xiao-xiao, *Fuzzy support vector machines based on FCM clustering. Proceddings of the fourth international conference on Machine Learning and Cybernetics*, Guangzhou, China: IEEE, pp. 2608-2613. Aug 18-21, 2005.
9. Liu Yi-hung, Chen Yen-ting, *face recognition using total margin based adaptive fuzzy support vector machines.*IEEE Transactions on Neural Networks, 18(1): pp 178-192. 2007.
10. Wei Yu-xin, Wu Mu-qing. *KFDA and clustering based multiclass SVM for intrusion detection. The Journal of china universities of posts and telecommunications* volume 15, issue 1, pp.123-128, March 2008.
11. V. Venkatachalam, S. Selvan. *Performance comparison of intrusion detection system classifiers using various features reduction techniques* International Journal of Simulation vol.9 no.1, pp.30-38, 2007.
12. Gopi K. Kuchimanchi,Vir V. Phoha, Kiran S.Balagani, Shekhar R. Gaddam, *Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems*, Proceedings of the IEEE on Information, 2004.
13. S. Snapp, J. Brentano, and G. Dias et al. *DIDS (Distributed Intrusion Detection System) – motivation, architecture,and an early prototype.* In Proceedings of the 14th National Computer Security Conference, October 1991.
14. Porras A, Neumann PG. EMERALD*: Event monitoring enabling responses to anomalous live disturbances.* In: Proceedings of the National Information Systems Security Conference; 1997.pp. 353–65.
15. S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. *GrIDS-a graph based intrusion detection system for large networks.* In Proceedings of the 19th National Information Systems Security Conference, September 1996.
16. Eugene H. Spafford and Diego Zamboni. *Intrusion detection using autonomous agents. Computer Networks*, 34 (4):547–570, October 2000.
17. Staniford-Chen S, Tung SB, Schnackenberg D. *the common intrusion detection framework (CIDF).* Proceedings of the information survivability workshop, Orlando FL, October 1998.
18. Ning P, Jajodia S, Wang XS*. Design and implementation of a decentralized prototype system for detecting distributed attacks.*Comput Commun 25:pp.1374-91, 2002.
19. Bernardes MC, dos Santos Moreira E*. implementation of an intrusion detection system based on mobile agents. In: Proceedings of the international symposium on software engineering for parallel and distributed systems*, 2000.pp.158-64.
20. Helmer G, Wong J, Honavar V, Miller L*. Intelligent agents for intrusion detection.* Available from http://citeseer.nj.nec.com/ helmer98intelligent.html. 1998.
21. KDDCUP2009dataset, August2003 http://kdd.ics.uci.edu /databases/ KDDCUP2009/KDDCUP2009.html.
22. Kim HC et al. *Face recognition using LDA mixture model*. In: Proceedings int conf. on pattern recognition, 2002.
23. C.Burges. *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, 2(2):pp. 121-167, 1998.