available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Intrusion detection using text processing techniques with a kernel based similarity measure

Alok Sharma[a,*], *Arun K. Pujari*[b], *Kuldip K. Paliwal*[a]

[a]*Signal Processing Laboratory, Griffith University (Nathan Campus), Brisbane, QLD-4111, Australia*
[b]*AI Laboratory, University of Hyderabad, India*

ARTICLE INFO

Article history:
Received 24 December 2006
Accepted 16 October 2007

Keywords:
Intrusion detection
kNN classifier
Similarity measure
Anomaly detection
Radial basis functions

ABSTRACT

This paper focuses on intrusion detection based on system call sequences using text processing techniques. It introduces kernel based similarity measure for the detection of host-based intrusions. The *k*-nearest neighbour (*k*NN) classifier is used to classify a process as either normal or abnormal. The proposed technique is evaluated on the DARPA-1998 database and its performance is compared with other existing techniques available in the literature. It is shown that this technique is significantly better than the other techniques in achieving lower false positive rates at 100% detection rate.

1. Introduction

Intrusion detection is an important technique in the defense-in-depth network security framework and a hot topic in computer network security in recent years. In general, intrusion detection system (IDS) is classified into two categories depending on the modeling methods used: signature-based and behaviour-based detection. In signature-based detection, we have a collection of pre-defined description or signatures of attacks. The current evidence is matched against these signatures to identify a possible threat. Although signature-based detection is effective in detecting known attacks, it cannot detect the new attacks that are not pre-defined (whose signatures are not available). Behaviour-based detection, also known as anomaly detection, on the other hand, builds profile of normal behaviour and attempts to identify the patterns or activities that deviate from the normal profile. An important feature of anomaly detection is that it can detect unknown attacks. Behaviour modeling can be done by either modeling the user behaviour or process. While modeling process behaviour, system call data are one of the most common types of data used. Host-based anomaly detection systems mostly focus on system call sequences with the assumption that a malicious activity results in an abnormal trace. Such data can be collected by logging the system calls using operating system utilities, e.g. Linux strace or Solaris Basic Security Module (BSM). In this framework, it is assumed that the normal behaviour can be profiled by a set of patterns of sequence of system calls. Any deviation from the normal pattern is termed an intrusion in this framework. An intrusion detection system needs to learn the normal behaviour patterns from the previously collected data and this is normally accomplished by data mining or machine learning techniques. The problem of intrusion detection thus boils down to a supervised classification problem to identify anomalous sequences, which are measurably different from the normal behaviour. The system call sequences of normal (as well as attack) instances are used as the training set. Though anomaly-based IDS is considered better than the signature-based IDS,

* *Corresponding author.*
E-mail address: sharma_al@usp.ac.fj (A. Sharma).
0167-4048/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2007.10.003

the former suffers from having unacceptable false positive rate (Lane and Brodley, 1998). This is because of the fact that it is hard to perfectly model a normal behaviour. Unlike the traditional pattern recognition approach for classification, the aim in the present context is not only to achieve high accuracy rate but also to minimize the false positive rate. In recent years, a lot of research activities in anomaly detection focus on learning process behaviours and building the profiles with system call sequences as data sources. We briefly review the research on behaviour-based approaches to host-based intrusion detection systems.

A pioneering work in behaviour-based intrusion detection is reported in Denning (1990) where profiles of subjects (users) are learnt and statistical methods are used to calculate deviations from the normal behaviour. Lane and Brodley (1997) propose another approach for capturing a user's behaviour. A database of sequences of UNIX commands that normally a user issues is maintained for each user. Any new command sequence is compared with this database using a similarity measure. Forrest et al. (1996, 1997) introduce a simple anomaly detection method based on monitoring the system calls invoked by active and privileged processes. The profile of normal behaviour is built by enumerating all fixed length of unique and contiguous system calls that occur in the training data, and unmatched sequences in actual detection are considered abnormal. A similar approach is followed by Lee et al. (1997), but they make use of a rule learner Repeated Incremental Pruning to Produce Error Reduction (RIPPER), to form the rules for classification. Lee and Stolfo (1998) use data mining approach to study a sample of system call data to characterize the sequences contained in normal data by a small set of rules. In monitoring and detection, the sequences violating those rules are treated as anomalies (Eskin et al., 2002). Warrender et al. (1999) propose Hidden Markov Model (HMM) method for modeling and evaluating invisible events based on system calls. It is believed that the entire sequence of system calls in a process need not exhibit intrusive behaviour, but few subsequences of very small lengths may possess the intrusive characteristics. Rawat et al. (2005) showed using rough set technique that the intrusive behaviour in a process is very localized.

Most of the IDSs that model the behaviour of processes in terms of subsequences take fixed-length contiguous subsequences of system calls. One potential drawback of this approach is that the size of the database that contains fixed-length contiguous subsequences increases exponentially with the length of the subsequences. Wespi et al. (2000) propose a variable length subsequence approach. Asaka et al. (2001) develop another approach based on the discriminant method in which an optimal classification surface is first learned from samples of the properly labeled normal and abnormal system call sequences. Wang et al. (2004) develop another Principle Component Analysis based method for anomaly intrusion detection with less computation efforts. Tandon and Chan (2005) propose to consider system calls' arguments and other parameters, along with the sequences of system calls. They make use of the variant of a rule learner LERAD (Learning Rules for Anomaly Detection).

In order to detect the deviation of anomalous system call sequences from the normal set of sequences, Liao and Vemuri (2002a) used a similarity measure based on the frequencies of system calls used by a program (process), rather than the temporal ordering. Their approach draws an analogy between text categorization and intrusion detection, such that each system call is treated as a word and a set of system calls generated by a process as a document. They used a '*bag of system calls*' representation. Liao and Vemuri (2002a,b) adopted this representation to profile the behaviour according to the trace of each process independently and a *k*NN method is used for classification. In this method, each system call is treated as a word and a collection of system calls during the execution of a process is treated as a document. The system call trace of a process is converted into a vector and cosine similarity measure is used to calculate the similarity among processes. In another study (Wenjie et al., 2003) by the same group, the Robust Support Vector Machine (RSVM) is applied to anomaly-based IDS. Recently, the emphasis of this RSVM study is on exhibiting the effectiveness of the method in the presence of noisy data. Rawat et al. (2006) proposed a very efficient anomaly-based host-based intrusion detection system. A new similarity measure is proposed and it is shown that by *k*NN classifier with the new measure, one can reduce the false positive rate substantially without sacrificing the detection rate. The authors have shown that by defining appropriate similarity measures, the detection by simple *k*NN can be as efficient as the sophisticated classification techniques like SVMs. The success of any such classification is hinged on two important aspects – the similarity measure and the classification scheme.

In this paper, we investigate the kernel based similarity measure for detection of host-based intrusion. We use the same classification scheme, namely *k*NN, as used by Liao and Vemuri (2002a) and Rawat et al. (2006). We demonstrate here that such a similarity measure captures the similarity in far better manner than the earlier measures. We show that the detection rates are better by the proposed scheme.

2. Similarity measure

Let $\Sigma = \{s_1, s_2, s_3, \ldots, s_m\}$ be a set of system calls where $m = |\Sigma|$ is the number of system calls. The training set D is defined as a set of labeled sequences $\{\langle Z_i, c_i\rangle | Z_i \in \Sigma^*; c_i \in \{0, 1\}\}$ where Z_i is an input sequence of system calls or a process, c_i is a corresponding class label denoting 0 for ''normal'' label and 1 for ''intrusion'' label and Σ^* is the set of all finite strings of symbol Σ. Given the data set D, the goal of the learning algorithm is to find a classifier h: $\Sigma^* \rightarrow \{0, 1\}$ that maximizes detection rate and minimizes false positive rate.

In the *bag of system calls* representation, the training set is represented as a matrix $A = [a_{ij}]$, where a_{ij} is the number of occurrence of system call s_i in process Z_j. In this representation, the ordering information of adjacent system calls in the input sequence is not considered to be significant and only the frequency of each system call in the bag is preserved.

The vector-space model of Information Retrieval (IR) is also often used to represent the set of processes. A process is represented as a binary (0–1) vector in terms of single or multiple occurrences of system call. The value 1 represents the occurrences of a system call in a process and its absence is represented by 0. Thus we define a matrix $B = [b_{ij}]$, where $b_{ij} = 1$, if

ith system call s_i is present in the jth process Z_j, and $b_{ij}=0$, otherwise.

Vemuri and his group (Liao and Vemuri, 2002a; Rawat et al., 2006) have defined the following measures, which we use in our scheme to calculate the similarity between processes.

2.1. Cosine similarity measure

The cosine similarity measure $\lambda(Z_i, Z_j)$ between any two processes Z_i and Z_j is defined as follows:

$$\mathrm{CosSim}(Z_i, Z_j) = \lambda(Z_i, Z_j) = \frac{Z_i \cdot Z_j}{\|Z_i\| \cdot \|Z_j\|} \quad (1)$$

where $\|\cdot\|$ is the Euclidean norm.

2.2. Binary similarity measure

The binary similarity measure $\mu(Z_i, Z_j)$ between any two processes Z_i and Z_j is defined as follows:

$$\mu(Z_i, Z_j) = \frac{\sum_{l=1}^{m} (B_i \wedge B_j)_l}{\sum_{l=1}^{m} (B_i \vee B_j)_l} \quad (2)$$

where the summation is a component-wise aggregation on l and B_j is the jth column of B corresponding to the process Z_j. The symbols $\wedge$ and $\vee$ are the bitwise AND and OR operators.

The cosine similarity measure was used by Liao and Vemuri (2002a). This measure was modified by Rawat et al. (2006). Rawat et al. (2006) observed that while calculating the similarity score, there is no weight accorded to processes having more number of common system calls. Since the cosine similarity measure considers only the frequencies of the system calls appearing in the processes, sometimes it may produce erroneous results. They define a similarity measure that depends not only on frequencies of system calls but also on the number of common system calls between the processes. They call their technique binary weighted cosine (BWC) similarity $\mathrm{Sim}(Z_i, Z_j)$ which is defined (Rawat et al., 2006) as

$$\mathrm{Sim}(Z_i, Z_j) = \mu(Z_i, Z_j) \cdot \lambda(Z_i, Z_j) \quad (3)$$

The motive behind multiplying μ and CosSim is that $\mathrm{CosSim}(Z_i, Z_j)$ measures the similarity based on the frequency and $\mu(Z_i, Z_j)$ is the weight associated with Z_i and Z_j. In other words, $\mu(Z_i, Z_j)$ tunes the similarity score $\mathrm{CosSim}(Z_i, Z_j)$ according to the number of similar and dissimilar system calls between the two processes. Therefore, the similarity measure $\mathrm{Sim}(Z_i, Z_j)$ takes frequency and the number of common system calls into consideration while calculating similarity between two processes.

3. k-Nearest neighbours with kernel similarity measure

The kNN method classifies any new process by examining the behaviour of the majority of k closest training processes. In theoretical terms, we are taking a small volume around the new process and the maximum likelihood estimate of the probability that this process belongs to a given class (normal or attack) is given by the proportion of training points in this volume that belongs to the class. The kNN is closely related to the kernel method as the kernel method considers the volumes of fixed radius whereas the kNN methods consider a volume with radius equal to the distance of the kth nearest neighbour. Thus while computing the proportion of training process inside the volume belonging to a class, the denominator becomes a random variable in the kernel method whereas it is fixed in the nearest neighbour method. There have been several extensions to these methods which attempt to combine the benefits of both the techniques. These include smoothly decaying kernel functions or differential weights on the nearest neighbour points according to their distances. Yet another possibility is to use kernel function as a similarity measure and follow the kNN method for classification. The kernel trick has been applied to several problems. For a detailed description about kernel methods, see Schölkopf and Smola (2002) and Shawe-Taylor and Cristianini (2005). The kernel methods allow us to work in a feature space of higher dimension. If $\phi(x)$ is a mapping of a point x in the input space to the feature space, then the kernel calculates the dot product in the feature space of corresponding points. Thus $K(x, x') = \phi(x) \cdot \phi(x')$, where '$\cdot$' denotes the dot product. Common kernel functions are radial $K(x, x') = \exp(-(1/2)\|x - x'\|^2)$ and polynomial $K(x, x') = (1 + x \cdot x')^d$. Distance in the feature space may be calculated by means of the kernel. With x and x' in the input space then the distance in the feature space is given by $K(x, x')$.

A matrix $A = [a_{ij}]$ can be computed from all the normal processes where a_{ij} denotes the frequency of ith system call in the jth process. To facilitate categorizing the process Z into either normal or abnormal class, the process Z is first converted into a vector. The kNN classifier then compares it with all the processes A_j in A to determine the k-nearest neighbours, by calculating the proposed kernel similarity measure defined as follows:

$$\mathrm{Sim}(Z, A_j) = \exp\left(-\frac{1}{2}\|Z - A_j\|^2 / \|A\|^2\right) \quad (4)$$

We have also defined a smoother function than Eq. (4) as follows:

$$\mathrm{Sim}(Z, A_j) = \exp\left(-\frac{1}{2}\|Z - A_j\|^2 / \|A\|\right) \quad (5)$$

We call Eq. (4) radial basis function (RBF) measure and Eq. (5) smooth RBF (SRBF) measure. The advantage of SRBF measure over RBF measure is lower false rate. Although the advantage is difficult to prove theoretically, it can be shown experimentally. This smoothness provides more sensible average value of the collection of k-nearest neighbours which is crucial in anomaly categorization.

We have also proposed a kernel extension of binary weighted cosine measure (Rawat et al., 2006). These measures can be defined as follows:

$$\mathrm{Sim}(Z, A_j) = \mu(Z, A_j) * \exp\left(-\frac{1}{2}\|Z - A_j\|^2 / \|A\|^2\right) \quad (6)$$

$$\mathrm{Sim}(Z, A_j) = \mu(Z, A_j) * \exp\left(-\frac{1}{2}\|Z - A_j\|^2 / \|A\|\right) \quad (7)$$

We call Eq. (6) binary weighted radial basis function (BWRBF)

Table 1 – List of 50 unique system calls

access, audit, auditon, chdir, chmod, chown, close, creat, execve, exit, fchdir, fchown, fcntl, fork, fork1, getaudit, getmsg, ioctl, kill, link, login, logout, lstat, memcntl, mkdir, mmap, munmap, oldnice, oldsetgid, oldsetuid, oldutime, open, pathdonf, pipe, putmsg, readlink, rename, rmdir, setaudit, setegid, seteuid, setgroups, setpgrp, setrlimit, stat, statvfs, su, sysinfo, unlink, vfork

and Eq. (7) smooth binary weighted radial basis function (SBWRBF). The motive behind multiplying the RBF measure by $\mu(Z, A_j)$ is to better tune the similarity score according to the number of similar and dissimilar system calls between the two processes.

4. An illustration using a toy problem

Consider a set of 10 unique system calls S associated with two processes Z_1 and Z_2. Let us consider a third process Z to measure the similarity score for possible intrusion. The variables are defined as follows:

$S = \{\text{access}, \text{audit}, \text{chomd}, \text{close}, \text{ioctl}, \text{login}, \text{mmap}, \text{open}, \text{pipe}, \text{su}\}$

$Z_1 = \{\text{open}, \text{close}, \text{close}, \text{close}, \text{close}, \text{access}, \text{access}, \text{access}\}$

$Z_2 = \{\text{open}, \text{ioctl}, \text{mmap}, \text{pipe}, \text{access}, \text{login}, \text{su}, \text{su}, \text{audit}, \text{audit}\}$

$Z = \{\text{open}, \text{close}, \text{ioctl}, \text{mmap}, \text{pipe}, \text{pipe}, \text{access}, \text{access}, \text{login}, \text{chmod}\}$

The similarity score using CosSim (Eq. (1)) of the new process Z with Z_1 and Z with Z_2 is:

$\text{CosSim}(Z, Z_1) = 0.6048$ and $\text{CosSim}(Z, Z_2) = 0.5714$

Table 2 – List of 54 attacks used in testing data set

1.1_it_ffb clear, 1.1_it_format_clear, 2.2_it_ipsweep, 2.5_it_ftpwrite, 2.5_it_ftpwrite_test, 3.1_it_ffb_clear, 3.3_it_ftpwrite, 3.3_it_ftpwrite_test, 3.4_it_warez, 3.5_it_warezmaster, 4.1_it_080520warezclient, 4.2_it_080511warezclient, 4.2_it_153736spy, 4.2_it_153736spy_test, 4.2_it_153812spy, 4.4_it_080514warezclient, 4.4_it_080514warezclient_test, 4.4_it_175320warezclient, 4.4_it_180326warezclient, 4.4_it_180955warezclient, 4.4_it_181945warezclient, 4.5_it_092212ffb, 4.5_it_141011loadmodule, 4.5_it_162228loadmodule, 4.5_it_174726loadmodule, 4.5_it_format, 5.1_it_141020ffb, 5.1_it_174729ffb_exec, 5.1_it_format, 5.2_it_144308eject_clear, 5.2_it_163909eject_clear, 5.3_it_eject_steal, 5.5_it_eject, 5.5_it_fdformat, 5.5_it_fdformat_chmod, 6.4_it_090647ffb, 6.4_it_093203eject, 6.4_it_095046eject, 6.4_it_100014eject, 6.4_it_122156eject, 6.4_it_144331ffb, test.1.2_format, test.1.2_format2, test.1.3_eject, test.1.3_httptunnel, test.1.4_eject, test.2.1_111516ffb, test.2.1_format, test.2.2_xsnoop, test.2.3_ps, test.2.3_ps_b, test.2.5_ftpwrite, test.2.4_eject_a, test.2.2_format1

Table 3 – Liao–Vemuri scheme

Threshold	False positive rate	Detection rate
0.52	0.0000	0.3519
0.74	0.0000	0.3519
0.78	0.0000	0.3704
0.84	0.0000	0.3704
0.86	0.0000	0.7407
0.89	0.0009	0.7593
0.91	0.0019	0.7593
0.94	0.0021	0.8148
0.96	0.0023	0.8333
0.99	0.0096	0.9630
0.9934	0.2284	1.0000

It can be observed that there are only three common system calls out of eight between Z and Z_1 and six common system calls out of eight between Z and Z_2. Thus hypothetically Z_2 is more similar to Z than Z_1. However, the results shown above indicate the opposite. Now consider the similarity measure using BWC (Eq. (3)) which is shown below:

$\text{Sim}(Z, Z_1) = 0.2268$ and $\text{Sim}(Z, Z_2) = 0.3429$

$\text{Sim}(Z, Z_1)/\text{Sim}(Z, Z_2) = 0.6614$

The results shown above validate the hypothesis. The ratio $\text{Sim}(Z, Z_1)/\text{Sim}(Z, Z_2)$ provides a collective observation of the process Z with Z_1 and Z_2. According to the hypothesis (for this particular case) the ratio should be less than unity. It is, however, difficult to state here the possible theoretical value of the ratio which might give some insight about the similarity among the three processes. The only way to understand the efficiency of measures is to test them under some large data set.

The RBF-based similarity measures are illustrated in this toy problem as follows:

BWRBF : $\text{Sim}(Z, Z_1) = 0.2031$ and $\text{Sim}(Z, Z_2) = 0.4226$, $\text{Sim}(Z, Z_1)/\text{Sim}(Z, Z_2) = 0.4805$

SBWRBF : $\text{Sim}(Z, Z_1) = 0.0104$ and $\text{Sim}(Z, Z_2) = 0.0772$, $\text{Sim}(Z, Z_1)/\text{Sim}(Z, Z_2) = 0.1342$

Table 4 – BWC scheme

Threshold	False positive rate	Detection rate
0.52	0.0000	0.3704
0.55	0.0000	0.3704
0.60	0.0004	0.4074
0.65	0.0004	0.5556
0.70	0.0032	0.8889
0.74	0.0044	0.9074
0.78	0.0057	0.9444
0.80	0.0062	0.9444
0.84	0.0083	0.9444
0.86	0.0095	0.9444
0.89	0.0170	0.9815
0.90	0.0238	0.9815
0.90099	0.0465	1.0000

Table 5 – BWRBF scheme

Threshold	False positive rate	Detection rate
0.52	0.0000	0.3519
0.60	0.0000	0.3519
0.65	0.0002	0.3704
0.70	0.0004	0.4444
0.74	0.0021	0.5000
0.78	0.0042	0.5556
0.80	0.0049	0.6296
0.84	0.0057	0.9444
0.86	0.0061	0.9444
0.89	0.0083	0.9444
0.94	0.0091	0.9815
0.96	0.0096	0.9815
0.99	0.0098	0.9815
0.999	0.0098	0.9815
0.9999	0.0098	1.0000

RBF : $\mathrm{Sim}(Z, Z_1) = 0.5415$ and $\mathrm{Sim}(Z, Z_2) = 0.7043$,
$\mathrm{Sim}(Z, Z_1)/\mathrm{Sim}(Z, Z_2) = 0.7688$

SRBF : $\mathrm{Sim}(Z, Z_1) = 0.0276$ and $\mathrm{Sim}(Z, Z_2) = 0.1286$,
$\mathrm{Sim}(Z, Z_1)/\mathrm{Sim}(Z, Z_2) = 0.2148$

The results obtained above indicate that the proposed measures are hypothetically correct and efficient than cosine similarity measure (Liao and Vemuri, 2002a). To observe their efficiencies over the techniques proposed by Liao and Vemuri (2002a) and Rawat et al. (2006), we conduct experiments on DARPA database (DARPA, 1998) as described in the next section.

5. Experimental setup and results

We use BSM audit logs from the 1998 DARPA data (DARPA, 1998) for training and testing of our algorithm. We use the same data set that is used by Liao and Vemuri (2002a) and Rawat et al. (2006).

There are 50 unique system calls in the training data. All the 50 system calls are shown in Table 1.

Table 6 – SBWRBF scheme

Threshold	False positive rate	Detection rate
0.52	0.0004	0.4815
0.55	0.0004	0.5000
0.60	0.0004	0.6111
0.65	0.0006	0.7778
0.70	0.0015	0.8889
0.74	0.0034	0.9259
0.78	0.0048	0.9444
0.80	0.0057	0.9630
0.84	0.0062	0.9630
0.86	0.0066	0.9630
0.89	0.0083	0.9630
0.91	0.0089	0.9630
0.91074	0.0091	1.0000

Table 7 – RBF scheme

Threshold	False positive rate	Detection rate
0.52	0.0000	0.3519
0.99	0.0000	0.3519
0.999	0.0000	0.3704
0.9999	0.0004	0.4259
0.99999	0.0006	0.8519
0.999991	0.0006	0.8889
0.999994	0.0011	0.9259
0.999996	0.0023	0.9444
0.999998	0.0038	0.9630
0.999999	0.0044	1.0000

There are about 2000 normal sessions reported in the four days of data and the training data set consists of 605 unique processes. There are 412 normal sessions on the fifth day and we extract 5285 normal processes from these sessions. We use these 5285 normal processes as testing data. In order to test the detection capability of our method, we considered 55 intrusive sessions as test data as taken by Rawat et al. (2006). But we found that there is one process that is exactly similar to the training data and hence we removed it from the session list. Thus we consider only 54 intrusive sessions instead. Table 2 lists these attacks. A number in the beginning of the name denotes the week and day followed by the name of the session (attack).

An intrusive session is said to be detected if any of the processes associated with this session is classified as abnormal. Thus detection rate is defined as the number of intrusive sessions detected, divided by the total number of intrusive sessions. We perform the experiments with $k = 5$. Tables 3–9 show the results for $k = 5$ using Liao–Vemuri scheme, BWC scheme, BWRBF scheme, SBWRBF scheme, RBF scheme and SRBF scheme. The first column in the tables shows the threshold values used in the experiments. Entries in column two are false positive rates. Finally, the entries in column three show the abnormal processes detected as abnormal, i.e. detection rate.

Table 8 – SRBF scheme

Threshold	False positive rate	Detection rate
0.52	0.0004	0.3889
0.65	0.0004	0.3889
0.70	0.0004	0.7593
0.84	0.0004	0.7593
0.86	0.0004	0.7778
0.89	0.0004	0.7963
0.91	0.0004	0.7963
0.94	0.0004	0.8333
0.96	0.0006	0.8519
0.965	0.0006	0.8889
0.97	0.0009	0.9074
0.975	0.0009	0.9259
0.99	0.0034	0.9630
0.99255	0.0038	1.0000

Table 9 – False positive rate versus detection rate for all the techniques

Method	False positive rate in %	Detection rate in %
Liao–Vemuri (Liao and Vemuri, 2002a)	22.84	100
BWC (Rawat et al., 2006)	4.65	100
BWRBF	0.98	100
SBWRBF	0.91	100
RBF	0.44	100
SRBF	0.38	100

It can be observed from Table 3 that for Liao–Vemuri scheme the false positive rate is very high (22.84%) at a detection rate of 100%. The reason of this high false positive rate is perhaps the degree of similarity of attack data with the normal processes. From Table 4 it is evident that BWC is a better technique than Liao–Vemuri as it provides lesser false positive rate (4.65%) at a detection rate of 100%. However, this false positive rate (4.65%) still may not be acceptable. The BWRBF scheme (see Table 5) gives only 0.98% false positive rate at 100% detection. This improvement is far better than both the previous techniques. The next proposed scheme is SBWRBF (see Table 6) which provides a false positive rate of 0.91% (an improvement of 0.07% over BWRBF) at a detection rate of 100%. Tables 7 and 8 illustrate the RBF scheme and SRBF scheme. It can be seen that the false positive rate by these techniques is 0.44% and 0.38%, respectively, at a detection rate of 100%. The results are summarized in Table 9. We believe that these results are the best reported results on this database to the best of our knowledge.

The receiver operating characteristic (ROC) curve is also used for comparing these techniques. The ROC curve is a graph between detection rate of attacks and false positive rate. The ideal ROC curve would be the y-axis, i.e. straight line perpendicular to the x-axis at the origin. In other words 0% false positive rate at 100% detection rate. Fig. 1 illustrates a comparison of ROCs of Liao–Vemuri, BWC and BWRBF schemes at $k = 5$. It can be observed from the figure that BWRBF converges to 100% detection rate faster than the other two schemes. A similar conclusion can be made from Fig. 2 where a comparison between BWC and SBWRBF is performed. Here also the proposed scheme (SBWRBF) is performing better than BWC scheme. Fig. 3 depicts ROC for BWRBF, SBWRBF, RBF and SRBF schemes. It is clear that all the proposed techniques are performing far better than the previous techniques. In addition, SRBF is dominating among all the other presented techniques.

The results obtained by the proposed schemes clearly demonstrate their effectiveness over the previously implemented techniques in terms of getting significant lower false positive rate at 100% detection rate.

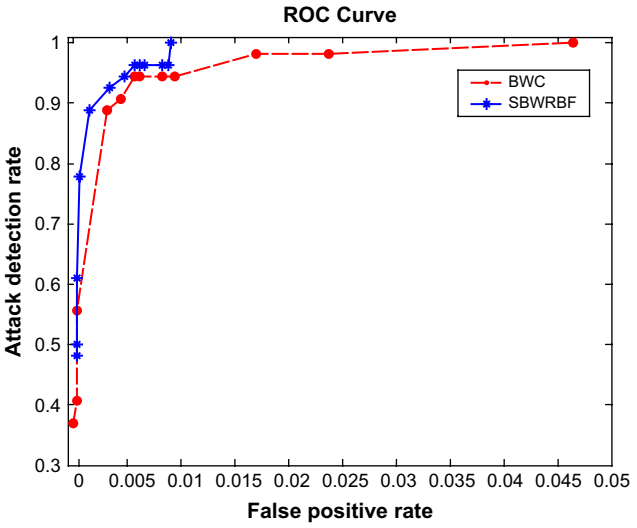

Fig. 2 – ROC curve for BWC and SBWRBF schemes at $k=5$.

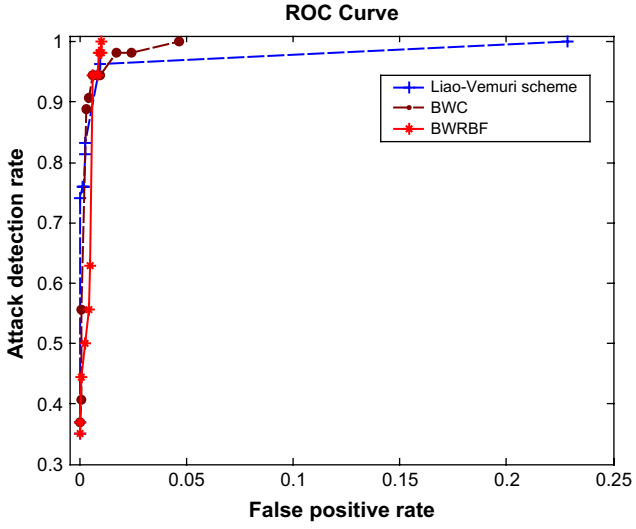

Fig. 1 – ROC curve for Liao–Vemuri, BWC and BWRBF schemes at $k=5$.

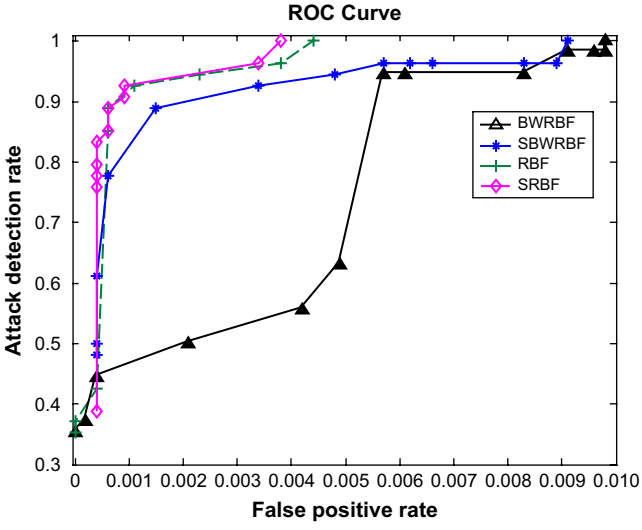

Fig. 3 – ROC curve for BWRBF, SBWRBF, RBF and SRBF schemes at $k=5$.

6. Conclusion

The kernel based similarity measure schemes have been introduced. The four types of measures were BWRBF, SBWRBF, RBF and SRBF. It is evident from the experiments that the proposed schemes produced very low false positive rate (as minimum as 0.38%) at the detection rate of 100%. This is clearly a significant achievement in the context of intrusion detection.

REFERENCES

Asaka M, Onabuta T, Inoue T, Okazawa S, Goto S. A new intrusion detection method based on discriminant analysis. IEICE Transaction on Information and Systems 2001;E84D(5): 570–7.

Denning DE. An intrusion-detection model. In: Proceedings of the 1986 IEEE symposium on security and privacy (SSP '86). IEEE Computer Society Press; 1990. p. 118–33.

DARPA 1998 data. MIT Lincoln Laboratory, <http://www.ll.mit.edu/IST/ideval/data/data index.html>; 1998.

Eskin E, Arnold A, Prerau M, Portnoy L, Stolfo S. A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Barbara D, Jajodia S, editors. Applications of data mining in computer security. Kluwer Academic Publishers; 2002. p. 77–102.

Forrest S, Hofmeyr SA, Somayaji A, Longstaff TA. A sense of self for Unix processes. In: Proceedings of the 1996 IEEE symposium on research in security and privacy, Los Alamos, CA; 1996. p. 120–28.

Forrest S, Hofmeyr SA, Somayaji A. Computer immunology. Communications of the ACM 1997;40(10):88–96.

Lane T, Brodley CE. An application of machine learning to anomaly detection. In: Proceedings of the 20th national information system security conference, Baltimore, MD; 1997. p. 366–77.

Lane T, Brodley CE. Temporal sequence learning and data reduction for anomaly detection. In: Proceedings of the fifth ACM conference on computer and communication security; 1998.

Lee W, Stolfo S, Chan P. Learning patterns from Unix process execution traces for intrusion detection. In: Proceedings of the AAAI97 workshop on AI methods in fraud and risk management. AAAI Press; 1997. p. 50–6.

Lee W, Stolfo S. Data mining approaches for intrusion detection. In: Proceedings of the seventh USENIX security symposium. USENIX Association; January 1998. p. 79–94.

Liao Y, Vemuri VR. Use of *k*-nearest neighbor classifier for intrusion detection. Computers & Security 2002a;21(5):439–48.

Liao Y, Vemuri VR. Using text categorization techniques for intrusion detection. In: Proceedings of the USENIX security 2002, San Francisco, US; 2002b. p. 51–9.

Rawat S, Gulati VP, Pujari AK. A fast host-based intrusion detection system using rough set theory. Transactions on Rough Sets 2005:144–61.

Rawat S, Gulati VP, Pujari AK, Vemuri VR. Intrusion detection using text processing techniques with a binary-weighted cosine metric. Journal of Information Assurance and Security 2006;1:43–50.

Schölkopf B, Smola AJ. Learning with kernels. The MIT Press; 2002.

Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. Cambridge University Press; 2005.

Tandon G, Chan PK. Learning useful system call attributes for anomaly detection. In: Proceedings of the 18th international Florida Artificial Intelligence Research Society (FLAIRS) conference; 2005. p. 405–10.

Wang W, Guan X, Zhang X. A novel intrusion detection method based on principle component analysis in computer security. In: Proceedings of the international IEEE symposium on neural networks, Dalian, China. Lecture notes in computer science, vol. 3174; August 2004. p. 657–62.

Warrender C, Forrest S, Pearlmutter B. Detecting intrusions using system calls: alternative data models. In: Proceedings of the 1999 IEEE symposium on security and privacy; 1999. p. 133–45.

Wenjie H, Liao Y, Vemuri VR. Robust support vector machines for anomaly detection in computer security. In: International conference on machine learning, Los Angeles, CA; 2003.

Wespi A, Dacier M, Debar H. Intrusion detection using variable-length audit trail patterns. In: Proceedings of the third international workshop on the recent advances in intrusion detection (RAID'2000). LNCS, vol. 1907; 2000.

Alok Sharma received the BTech degree from the University of the South Pacific (USP), Fiji in 2000, MEng degree from Griffith University, Australia with academic excellence award in 2001 and PhD degree in the area of Pattern Recognition from Griffith University in 2006. He participated in various projects carried out in conjunction with Motorola (Sydney), Auslog Pty. Ltd. (Brisbane), CRC Micro Technology (Brisbane) and French Embassy (Suva). His research interests include pattern recognition, computer security and human cancer classification. He reviewed several articles from the journals like IEEE Transactions on Neural Networks, IEEE Transaction on Systems, Man, and Cybernatics, Part A: Systems and Humans, IEEE Journal on Selected Topics in Signal Processing, and Pattern Recognition. Presently he is serving as an academic and Head of Division of the Division of Electrical/Electronics, School of Engineering and Physics, USP.

Arun K. Pujari received his PhD from Indian Institute of Technology, Kanpur, India. He is currently professor of Computer Science at the LNM IIT, Jaipur. Prior to this he served at University of Hyderabad, Hyderabad, India as Dean, School of MCIS. His research interests include Intrusion Detection, Temporal and Spatial Constraint satisfaction problem, Automated reasoning, Data mining and Combinatorial Algorithms. He has authored a text book on Data mining.

Kuldip K. Paliwal was born in Aligarh, India, in 1952. He received the B.S. degree from Agra University, Agra, India, in 1969, the M.S. degree from Aligarh Muslim University, Aligarh, India, in 1971 and the PhD degree from Bombay University, Bombay, India, in 1978.

He has been carrying out research in the area of speech processing since 1972. He has worked at a number of organizations including Tata Institute of Fundamental Research, Bombay, India, Norwegian Institute of Technology, Trondheim, Norway, University of Keele, UK, AT&T Bell Laboratories, Murray Hill, New Jersey, USA, AT&T Shannon Laboratories, Florham Park, New Jersey, USA, and Advanced Telecommunication Research Laboratories, Kyoto, Japan. Since July 1993, he has been a professor at Griffith University, Brisbane, Australia, in the School of Microelectronic Engineering. His current research interests include speech recognition,

speech coding, speaker recognition, speech enhancement, face recognition, image coding, pattern recognition and artificial neural networks. He has published more than 250 papers in these research areas.

Dr. Paliwal is a Fellow of Acoustical Society of India. He has served the IEEE Signal Processing Society's Neural Networks Technical Committee as a founding member from 1991 to 1995 and the Speech Processing Technical Committee from 1999 to 2003. He was an Associate Editor of the IEEE Transactions on Speech and Audio Processing during the periods 1994–1997 and 2003–2004. He also served as Associate Editor of the IEEE Signal Processing Letters from 1997 to 2000. He was the General Co-Chair of the Tenth IEEE Workshop on Neural Networks for Signal Processing (NNSP2000). He has co-edited two books: ''Speech Coding and Synthesis'' (published by Elsevier), and ''Speech and Speaker Recognition: Advanced Topics'' (published by Kluwer). He has received IEEE Signal Processing Society's best (senior) paper award in 1995 for his paper on LPC quantization. He is currently serving the Speech Communication journal (published by Elsevier) as its Editor-in-Chief.