

Intrusion Detection System using Hybrid Differential Evolution and Group Method of Data Handling Approach

Godfrey C. Onwubolu¹, Alok Sharma²

¹Richmond Hill L4C, Canada, ²University of the South Pacific, Fiji

onwubolu@gmail.com, sharma_al@usp.ac.fj

Abstract. This paper proposes a new intrusion detection methodology based on hybrid of differential evolution (DE) and group method of data handling (GMDH). It focuses on intrusion detection based on system call sequences using text processing techniques. The hybrid DE-GMDH is used to classify a process as either normal or abnormal. This work presents the application of PCA and hybrid DE-GMDH to modeling high dimensional bench-mark DARPA-1998 database. For modeling and classifying the data, we adopted this combination of two stage PCA and hybrid DE-GMDH procedure. The presented technique shows significantly better results than other existing techniques available in the literature in achieving lower false positive rates at 100% detection rate.

Keywords

Dimensionality reduction, inductive modeling, DE, GMDH, hybrid systems, intrusion detection.

1 Introduction

With the widespread use of networked computers and the rapid growth of attacks on computers, intrusion detection systems (IDS) have received remarkable attention in recent years. There are two types of IDS depending on the audit data. The network-based systems collect data directly from the network that is being monitored while the host-based systems collect data from the host being protected. Also, IDS can be classified based on the modeling methods used: misuse or signature-based and anomaly or behavior-based detection. In misuse or signature-based detection, the signatures of known attacks in the database are kept and compared with new instances to determine if there are attacks. In anomaly or behavior-based detection, the normal behavior of the monitored system is learnt to determine if there is any deviation in it for signs of intrusions. An important feature of anomaly behavior is that it can detect unknown attacks and hence it is preferred to signature-based behavior. Behavior modeling can be done by either modeling the user behavior or process. Process behavior modeling requires system call data usage. Host-based anomaly detection systems mostly focus on system call sequences with the assumption that a malicious activity results in abnormal trace. Generally, such data can be collected by logging the system calls using operating system utilities such as Linux strace or Solaris basic security module (BSM) and the assumption is that the normal behavior can be profiled by a set of patterns of sequence of system calls. Any deviation from the normal pattern is termed an intrusion in the framework of anomaly-based IDS. An intrusion detection system needs to learn from the previously collected data and this is accomplished by data mining or machine learning techniques. It is generally accepted that the problem of intrusion detection boils down to a supervised classification problem to identify anomalous sequences, which are measurably different from the normal behaviors.

A first published work on behavior-based intrusion detection is by Denning [1]. Although anomaly-based IDS is preferred to signature-based IDS, the former is known to have unacceptable false positive rate than the later [2-3]. The reason postulated is the fact that it is hard to perfectly model a normal behavior. Consequently, a lot of research is being done in the area of anomaly-based IDS [2]. Lane and Brodley [4] propose an approach for capturing behavior of a user. Forrest *et al.* [5-6] introduce a simple anomaly detection method based on monitoring the system calls invoked by active and privileged processes.

Lee et al. [7] followed similar approach but use a rule learner for classification. In addition, data mining approach [8], hidden markov model (HMM) approach [9], rough set technique [10], variable length subsequence approach [11], discriminant method [12], principal component analysis (PCA) [13] have been used to solve the IDS problem. More recently, similarity measure-based approaches have been used by Liao and Vemuri [14-15]; robust support vector machine (RSVM) has been used by Wenjie et al. [16]; Rawat et al. [17] proposed a very efficient anomaly-based hot-based intrusion detection system, and kernel based similarity measures have been used by Sharma *et al.* [18].

In the work reported in this paper, we propose an intrusion detection methodology based on hybrid of differential evolution (DE) and group method of data handling (GMDH). The major contributions of our work are (a) the introduction of a novel method that does not use similarity measures based on commonality and frequency of occurrence of system calls and (b) realizing an anomaly-based IDS that gives an acceptably low false positive rate. We confirm our claims of better IDS by experimental analysis of the results of the proposed approach and other existing techniques in the literature.

2 The DDR Hybrid DE-GMDH Scheme for Intrusion Detection

The data dimensional reduction (DDR) hybrid DE-GMDH algorithm that is proposed for intrusion detection in this paper consists of three components: (i) the DDR for data dimensional reduction; (ii) the DE structural optimization module; (iii) the GMDH parametric optimization module.

2.1 PCA for Data Dimensional Reduction Module of Host being Protected

PCA finds a linear transformation Φ which reduces d -dimensional data to h -dimensional feature vectors (where $h < d$) in such a way that the information is maximally preserved in minimum mean squared error sense. This linear transformation is known as PCA transform or Karhunen-Loève transform (KLT) [19]. Since the transformation is from d -dimensional feature space to h -dimensional feature space the size of Φ is $d \times h$. The h column vectors of the matrix Φ are the basis vectors. The first basis vector is in the direction of maximum variance of the given feature vectors. The remaining basis vectors are mutually orthogonal and, in order, maximize the remaining variances subject to the orthogonal condition. Each basis vector represents a principal axis. These principal axes are those orthonormal axes onto which the remaining variances under projection are maximum. These orthonormal axes are given by the dominant/leading eigenvectors (i.e. those with the largest associated eigenvalues) of the measured covariance matrix. In PCA, original feature space is characterized by these basis vectors and the number of basis vectors used for characterization is usually less than the dimensionality d of the feature space (see Sharma et al. [20]; Sharma and Paliwal [21]).

In intrusion detection systems analysis where there is need to learn the normal behavior patterns from the previously collected (usually large) data, the PCA technique is applied for two main reasons

- (i) the basis vectors that are of less importance can be discarded which would help in reducing the noise that could be present in IDS data.
- (ii) To overcome the singularity issue related with the direct application of GMDH parametric optimization.

The PCA transform can be found by minimizing mean squared error. To see this, let the feature vector be $\mathbf{x} \in \mathbf{R}^d$ (d -dimensional space), reduced dimensional feature vector be $\mathbf{z} \in \mathbf{R}^h$ and reconstructed feature vector be $\hat{\mathbf{x}} \in \mathbf{R}^d$. Then the mean squared error can be represented as

$$\text{MSE} = E[\|\mathbf{x} - \hat{\mathbf{x}}\|^2] \quad (1)$$

where $E[\bullet]$ is the expectation operation with respect to \mathbf{x} and $\|\bullet\|^2$ is the norm squared value. We know that PCA transformation Φ is of size $d \times h$ and it is used to do dimensionality reduction from d -dimensional space to h -dimensional feature space, i.e. $\Phi : \mathbf{x} \rightarrow \mathbf{z}$ or

$$\mathbf{z} = \Phi^T \mathbf{x} \quad (2)$$

The PCA transformation Φ can be obtained by minimizing mean squared error $E[\|\mathbf{x} - \hat{\mathbf{x}}\|^2]$ which turns out to be a generalized eigenvalue problem i.e.:

$$\Sigma_{\mathbf{x}} \boldsymbol{\varphi}_j = \lambda_j \boldsymbol{\varphi}_j \quad (3)$$

where $\boldsymbol{\Phi} = \{\boldsymbol{\varphi}_j : j = 1, 2, \dots, h\}$, $\boldsymbol{\varphi}_j \in \mathbf{R}^d$, and $\Sigma_{\mathbf{x}}$ is covariance matrix of all input d -dimensional vectors. The expression λ_j denotes eigenvalues corresponding to $\boldsymbol{\varphi}_j$. The eigenvectors $(\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_h)$ of $\boldsymbol{\Phi}$ should be arranged such that their corresponding eigenvalues are in descending order $\lambda_1 > \lambda_2 > \dots > \lambda_h$. This arrangement is, however, not a necessary step for PCA but it is mentioned here since the model conducts this arrangement process prior to the application of GMDH parametric optimization. Sharma and Onwubolu [22] presented this PCA approach for modeling; it was implemented in the work reported in this paper.

2.2 The Group Method of Data Handling

The GMDH was first developed by Ivakhnenko [23] as a multivariate analysis method for complex system analysis modeling and identification. In this way, GMDH was used to circumvent the difficulty of knowing a priori knowledge of mathematical model of the process being considered. Therefore, GMDH can be used to model complex systems without having specific knowledge of the system. In a system identification problem, assume that a single valued output y (classes, in our case), of an unknown system, behaves as a function of m input values (samples of features in the reduced dimensional space, in our case), such that

$$y = f(z_1, z_2, \dots, z_m) \quad (4)$$

Given n training observations of these input-output data pairs $(z_{ij}; y_i)$, $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$ then the system identification problem is to approximate the function \bar{f} with an approximate function referred to as the complete form. In the application under consideration in this paper, z_{ij} define the attributes (features) while y_i define the class of labels. The GMDH approach is one of the heuristic algorithms primarily designed to solve the system identification problem. This section illustrates the conventional GMDH approach. To illustrate the technique, let be $\mathbf{z} = [z_1, z_2, \dots, z_h]^T$ the h -dimensional feature vector and let its corresponding label (state of nature) be y . According to the GMDH paradigm the relationship between the input vector \mathbf{z} and its label y can be represented by an infinite Volterra-Kolmogorov-Gabor (VKG) polynomial of the form [23]:

$$y = a_0 + \sum_{i=1}^m a_i z_i + \sum_{i=1}^m \sum_{j=1}^m a_{ij} z_i z_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m a_{ijk} z_i z_j z_k \dots \quad (5)$$

where $a_0, a_i, a_{ij}, a_{ijk} \dots$ are coefficients or weights of this (so called) multiple inputs single output self-organizing network. This is the discrete-time analogue of a continuous time Volterra series and can be used to approximate any stationary random sequence of physical measurements. Ivakhnenko showed that the VKG series can be expressed as a cascade of second order polynomials using only pairs of variables [23]. The corresponding network can be constructed from simple polynomial and delay elements. As the learning procedure evolves, branches that do not contribute significantly to the specific output can be pruned, thereby allowing only the dominant causal relationship to evolve.

In the classical GMDH algorithm, there is the need to construct $\alpha = \binom{m}{2} = m(m-1)/2$ new variables $y_1, y_2, y_3, \dots, y_\alpha$, in the *training dataset* for all independent variables (columns of \mathbf{Z}), two at a time so that there are n data triples of the form $\langle (z_{ir}, z_{is}, y_i), r, s \in (1, 2, \dots, m), s > r; i = 1, 2, \dots, n \rangle$. The mathematical description of these two new variables or neurons represented by quadratic polynomials is of the form

$$\hat{y}_i = a_0 + a_1 z_{ir} + a_2 z_{is} + a_3 z_{ir}^2 + a_4 z_{is}^2 + a_5 z_{ir} z_{is} \quad \text{at points } (z_{i,r}, z_{i,s}) \quad (6)$$

The matrix formulation of this system of equation is simply, $\mathbf{Za} = \mathbf{Y}$ where $\mathbf{a} = [a_0, a_1, a_2, a_3, a_4, a_5]^T$ is the vector of coefficients. The least square technique from multiple-regression analysis provides the most fundamental formula to obtain the coefficients in the following form: $\mathbf{a} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{Y}$.

2.3 Differential Evolution Scheme

The differential evolution (DE) algorithm introduced by Storn and Price [24] is a novel parallel direct search method, which utilizes N_p parameter vectors as a population for each generation G . It was primarily designed for continuous domain space formulation. The steps involved in the classical DE are summarized here

- Step 1: Initialization;
- Step 2: Mutation;
- Step 3: Crossover;
- Step 4: Selection;
- Step 5: Stopping criteria.

2.3.1 Permutative-based DE

Permutative-based DE differs from continuous DE due to the fact that it can handle permutative-based type combinatorial optimization problems. The mechanisms to cater for this are mainly the way in which initialization is done together with two other schemes for transformation from permutation form into continuous form in Step 2 in Section 2.3 and transformation from continuous form into permutation form after Step 2 in Section 2.3 [25].

2.3.2 Enhanced Permutative-based DE

As the name implies, the enhanced permutative-based DE uses the same basis as the permutative-based DE except that it has more enhancement strategies [26].

2.4 The Hybrid DE-GMDH Scheme

Although GMDH has been applied to solving the classification problem (see for example, Lemke and Mueller, 2003)[27] the results obtained were not as promising as expected due to the fact that classical GMDH has its strength in regression problems rather than in classification problems. The hybrid DE-GMDH scheme introduced by Onwubolu [28-29] (see Figure 1) overcomes the shortcomings of the conventional GMDH algorithm for the intrusion detection classification problem described in Section 3. It comprises of several components including structural and parametric optimization schemes. The current version has the module described in Section 2.1, as its pre-processor module. The elements of the reduced dimensional feature vector of $\mathbf{z} \in \mathbf{R}^h$ are candidates for entry in a pair-wise manner to the hybrid DE-GMDH modeling system.

2.4.1 Structural Optimization with DE

The DE design is responsible for selecting the number of input variables (attributes), selection of the input variables (attributes), and selection of polynomial order (linear, quadratic, trilinear, tri-quadratic, etc.). From iteration (generation) to iteration (generation) of the DE, the best nodes (neurons) are automatically found based on regression and parametric optimization, and progress is made until termination conditions are satisfied.

The summary of the overall architecture (Figure 1) of the hybrid DE-GMDH modeling system is as follows:

- Step 1: Reduce the high dimensional dataset from d to h dimension using PCA pre-processor (Section 2.1)
- Step 2: Initialize a population of discrete trial solutions (Section 2.3)
- Step 3: Evaluate the objective function (fitness) for each discrete current solutions in the population (Section 2.2)
- Step 4: Convert the permutative-based current solutions into continuous current solutions (Section 2.3.2)
- Step 5: Apply DE strategy to transform current solutions into new solutions using the inbuilt crossover and mutation schemes (Section 2.3.1 and 2.3.2)
- Step 6: Convert the continuous new solutions into permutative-based new solutions (Section 2.3.2)
- Step 7: Repair solutions to realize discrete new solutions of unique values (Section 2.3.2)
- Step 8: Improve solutions through standard crossover and mutation schemes (Section 2.3.1 and 2.3.2)
- Step 9: Execute steps 2—7 until reaching a specified cut-off limit on the total number of iterations
- Step 10: Improve solutions further through local search routine (Section 2.3.1 and 2.3.2)

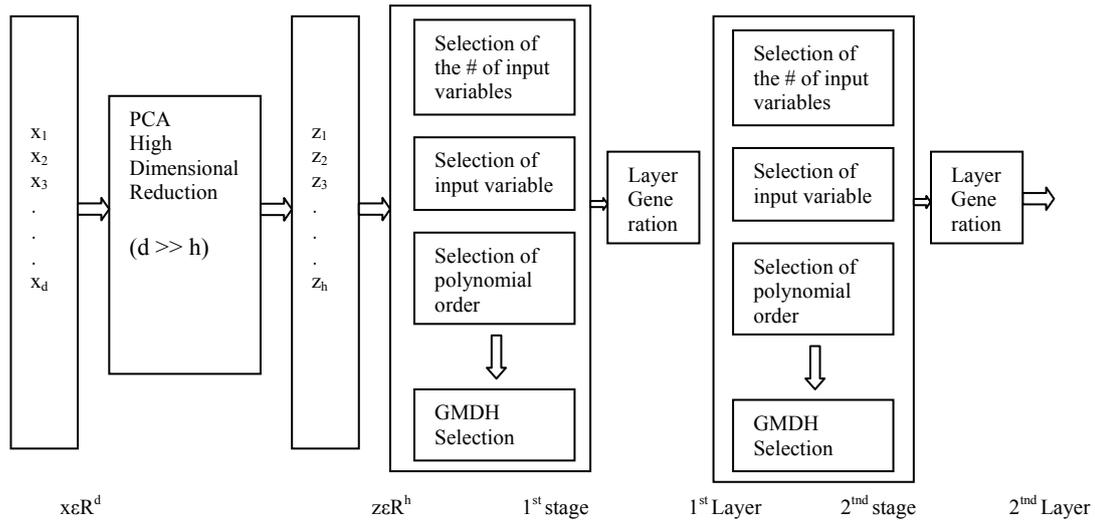


Fig. 1. Overall architecture of the hybrid DE-GMDH modeling system

To understand network realization let us take an example. Suppose we have input dataset of 5 dimensional vectors (record). Each vector has only and only one output. Take a scenario where the order of the polynomial is 2 and the number of inputs to a node is 2. Then, the inputs to a node(neuron) are shown in Table 1 which are just for layer 1. The DE mechanics detailed in Section 2.2.1 are vehicles for propagating the network from layer to layer.

Tab. 1. Inputs to a node(neuron)

Vectors	Solution vectors	Node(neuron) inputs
Vector 1	5, 4, 3, 2, 1	5, 4
Vector 2	1, 3, 5, 4, 2	1, 3
Vector 3	2, 1, 4, 5, 3	2, 1
Vector 4	4, 3, 1, 5, 2	4, 3
Vector 5	1, 5, 2, 3, 4	1, 5

2.4.2 Parametric Optimization with GMDH

The objective function (performance index) is a basic instrument guiding the evolutionary search in the solution space. For the third solution vector (see Table 1) the generated polynomial would be:

$$f(x_2, x_1) = c_1 + c_2 x_2 + c_3 x_1 + c_4 x_2 x_1 + c_5 x_2^2 + c_6 x_1^2$$

where c_1, c_2, \dots, c_6 are the constants evaluated using training dataset. As discussed in Section 2, the least square technique from multiple-regression analysis provides the formula to obtain the coefficients in the following form: $\mathbf{c} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$.

2.5 Procedure of the General Learning DE- GMDH Algorithm

The general learning procedure for constructing the DE-GMDH model can be described as follows:

- (1) Create an initial population randomly (DE structures and its corresponding learning parameters).
- (2) Structural optimization is achieved by the DE variation operators described in Section 2.2.1.
- (3) If better structure is found, then go to step 4; otherwise go to step 2.
- (4) Parametric optimization is found using pseudo-inverse or singular value decomposition (SVD) described in Section 2.2.2.
- (5) If the maximum number of local is reached or no better parameter vector is found for a significantly long time, then go to step 6; otherwise go to step 4.
- (6) If a satisfactory solution is found, then the algorithm is stopped; otherwise go to step 2.

3 Intrusion Detection Methodology

For the intrusion detection problem solved in the work reported in this paper, the same steps used for modeling general complex systems were used for classification without any modification. The input parameters (attributes of the reduced space) and the output parameters (class labels) were divided into training and testing datasets according to the

problem description described in Section 4. The training dataset was used for learning. The input parameters for the testing dataset were then presented to the hybrid DE-GMDH system for generalization. The results obtained were merely compared with the desired outputs; if they corresponded, then the test data was classified as with existing class, otherwise a new class was created. No further pre-processing was needed.

3.1 Training Phase

The dimension reduction stage has been applied to reduce the feature space. Once the dimension is reduced then it is processed by the hybrid DE-GMDH stage.

3.2 Testing Phase

The test samples are not used for model building but the model built is used for testing. In this paper, we present hybrid DE-GMDH-type networks that are comprehensively described by concise short-term polynomials, which are comprehensible for computer experts. The parameters used for both dimensional reduction and optimization (DE) are given in Table 2.

Tab. 2. Parameter used for both dimensional reduction and optimization (DE)

<i>Dimensional reduction</i>	Parameter values
Reduced dimension, h	3
<i>Optimization (DE)</i>	
Population size	50
Crossover value, CR	0.3
Mutation value	0.1

4 Experimentation Setup and Results

The experimental dataset used for the intrusion detection problem described in this paper is the BSM audit logs from the 1998 DARPA data [29]. We use the same dataset that is used by Liao and Vemuri [14], Rawat et al. [17], and Sharma et al. [18]. There are 50 unique system calls in the training data. All the 50 system calls are shown in Table 3. There are 2000 normal sessions reported in the four days of data and the training dataset consists of 605 unique processors. There are 412 normal sessions on the fifth day and we extract 5285 normal processes from these sessions. We use the 5285 normal processes as testing data. In order to test the detection capability of our proposed method, we considered 55 intrusion sessions as test data as taken by Rawat et al. [17]. 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. Consequently, we consider only 54 intrusion sessions instead. Table 4 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 intrusion session is said to be detected if any of the processes associated with this session is classified as abnormal. We define some metric for measuring the quality of our IDS solutions:

$$\text{detection rate} = \frac{\# \text{ of intrusion sessions detected (true positives)}}{\# \text{ of intrusion session}}$$

$$\text{false positive rate} = \frac{\# \text{ of normal processes detected as abnormal (false positives)}}{\# \text{ of normal session}}$$

$$\text{accuracy} = \frac{\# \text{ of true positives} + \# \text{ of false positives}}{\# \text{ of input sequences}}$$

Tab. 3. Table 2 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, setgrp, setrlimit, stat, statvfs, su, sysinfo, unlink, vfork

Tab. 4. Table 2 List of 54 attacks used in testing dataset

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_15373spy_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_095246eject, 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

5 Discussion of Results

Table 5 shows the results for $k = 5$ using Liao-Vemuri scheme; binary weighted cosine (BWC) by Rawat et al. [17] scheme; binary weighted radial basis function (BWRBF) scheme, smooth binary weighted radial basis function (SBWRBF) scheme, basis function (RBF) scheme, smooth radial basis function (SRBF) scheme by Sharma et al. [18] and the present one in this paper, hybrid DE-GMDH.

Tab. 5. 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 (Sharma et al., 2007)	0.98	100
SBWRBF (Sharma et al., 2007)	0.91	100
RBF (Sharma et al., 2007)	0.44	100
SRBF (Sharma et al., 2007)	0.38	100
Present paper: hybrid DE-GMDH	0.00	100

It can be observed from Table 5 that the false positive rate for Liao-Vemuri scheme is very high (22.84%) at a detection rate of 100%. BWC by Rawat et al. [17] performs better than that of Liao-Vemuri scheme with less false positive rate (4.65% at a detection rate of 100%). The kernel schemes proposed by Sharma et al. [18] have very small false positive rates of 0.98%, 0.91%, 0.44%, and 0.38% at a detection rate of 100% for the BWRBF, SBWRBF, RBF, and SRBF schemes respectively. The hybrid DE-GMDH scheme performs better than all other schemes with 0% false positive rate and a detection rate of 100%. To the best of our knowledge, the result of the proposed hybrid DE-GMDH scheme on this dataset is the best reported result.

6 Conclusion

This paper has introduced the hybrid DE-GMDH scheme for solving the intrusion detection problem. The result obtained when compared to the best previous BWRBF, SBWRBF, RBF, and SRBF schemes show that the hybrid DE-GMDH scheme realizes a 0% false positive rate and a detection rate of 100%. It is evident that this is a significant achievement in the context of intrusion detection.

Some of the advantages of the hybrid DE-GMDH scheme over the other schemes for the intrusion detection problem are: (i) no similarity measure is needed, (ii) defining thresholds is not necessary, (iii) features defining the problem are used directly and their dimension could be reduced using PCA without compromising the solution, (iv) computational speed is very high.

References

- [1] Denning, D.E.: An intrusion-detection model. In: *Proceedings of the 1986 IEEE Symposium on Security and Privacy (SSP '86)*. IEEE Computer Society Press; 1990, p. 118-131.
- [2] Axeleson, S.: *Research in Intrusion Detection Systems: A Survey*, Technical Report No. 98-17, Dept. of Computer Engineering, Chalmers University of Technology, Gteborg, Sweden, 1999.
- [3] Lane, T., Brodley, C.E.: Temporal sequence learning and data reduction for anomaly detection. In: *Proceedings of the fifth ACM Conference on Computer and Communication Security*, 1998.
- [4] Lane, T., Brodley, C.E.: An application of machine learning to anomaly detection. In: *Proceedings of the 20th National Information System Security Conference*, Baltimore, MD, 366-377, 1997.
- [5] Forrester, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of self for Unix processes. In: *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, Los Alamos, CA, 120-128, 1996
- [6] Forrester, S., Hofmeyr, S.A., Somayaji, A.: A computer immunology. *Communication of the ACM*, 40(10), 88-96, 1997
- [7] Lee, W., Stolfo, S., Chan, P.: Learning patterns from Unix process execution traces for intrusion detection. In: *Proceedings of the AAAI97 workshop of AI methods in fraud and risk management*. AAAI Press, 50-56, 1997.
- [8] Lee, W., Stolfo, S.: Data mining for intrusion detection. In: *Proceedings of the seventh USENIX Association*, 79-94, January, 1998.
- [9] Warrender, C., Forrest, S., Pearlmutter, B., Detecting intrusions using system calls: alternative data models. In: *Proceedings of the 1999 IEEE Symposium on Research in Security and Privacy*, 133-145, 1999.
- [10] Rawat, S., Gulati, V. P., Pujari, A. K.: A host-based intrusion detection system using rough theory. *Transactions on Rough Sets*, 144-161, 2005.
- [11] Wepsi, 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.
- [12] Asaka, M., Onabuta, T., Inoue, T., Okazawa, S., Goto, S.: A new intrusion detection system based on discriminant analysis. *IEICE Transaction on Information and Systems* 2001, E84D (5): 570-577
- [13] Wang, W., Guan, X., Zhang, X.: A novel intrusion detection method based on principal component analysis in computer security. In: *Proceedings of the International IEEE Symposium on Neural Networks*, Dalian, China. *Lecture Notes in Computer Science*, vol. 3174, 657-662, August 2004.
- [14] Liao, Y., Vemuri, V. R.: Use of k-nearest neighbor classifier for intrusion detection. *Computer & Security* 21(5), 439-448, 2002a.
- [15] Liao, Y., Vemuri, V. R.: Using text categorization techniques for intrusion detection. In: *Proceedings of the USENIX security 2002*, San Francisco, US, 51-59, 2002b.
- [16] Wenji, H., Liao, Y., Vemuri, V. R.: Robust support vector machines for anomaly detection in computer security. In: *Interantional Conference on Machine Learning*, Los Angeles, CA, 2003.
- [17] Rawat, S., Gulati, V. P., Pujari, A. K., Vemuri, V. R.: Intrusion detection using text processing techniques with a binary-weighted cosine metric. *Journal of International Assurance and Security*, 1, 43-50, 2006.
- [18] Sharma, A., Pujari, A. K., Paliwal, K. K.: Intrusion detection using text processing techniques with a kernel based similarity measure, *Computer & Security* 26, 448-495, 2007.
- [19] Fukunaga, K.: *Introduction to statistical pattern recognition*. Academic Press Inc., Hartcourt Brace Jovanovich, Publishers, 1990
- [20] Sharma, A. and Paliwal, K.K., Onwubolu, G.C.: Class-dependent PCA, LDA and MDC: a combined classifier for pattern classification. *Pattern Recognition*, 39(7), p. 1215-1229, 2006.
- [21] Sharma, A. and Paliwal, K.K., Fast principal component analysis using fixed-point algorithm", *Pattern Recognition Letters*, 28, p. 1151-1155, 2007.
- [22] Sharma, A., Onwubolu, G.C.: A Hybrid Approach for Modeling High Dimensional Medical Data, *Proceedings of International Workshop on Inductive Modeling, Prague, Czech, 2007*
- [23] Ivakhnenko, A. G.: The Group Method of Data Handling-A rival of the Method of Stochastic Approximation. *Soviet Automatic Control, vol 13 c/c of avtomatika*, 1, 3, (1968) 43-55.
- [24] Storn, R. M., Price, K. V. and Lampinene, J. A.: *Differential Evolution: A Practical Approach to Global Optimization*, Springer-Verlag, Berlin 2005.
- [25] Onwubolu, G. C.: Optimization using differential evolution, *Institute of Applied Science Technical Report, TR-2001/05*, 2001.
- [26] D. Davendra, G. C. Onwubolu, Scheduling flow shops using enhanced differential evolution algorithm, *European Conference on Modeling and Simulation (ECMS)*, Prague, Czech, 2007.
- [27] Lemke, F., Mueller, J. A.: Medical data analysis using self-organizing data mining technologies, *Systems Analysis Modeling Simulation*, 43(10), 1399-1408, 2003.
- [28] Onwubolu, G. C.: Design of Hybrid Differential Evolution and Group Method of Data Handling for Inductive Modeling, *Proceedings of International Workshop on Inductive Modeling, Prague, Czech*, 87-95, 2007
- [29] Onwubolu, G. C.: Design of hybrid differential evolution and group method in data handling networks for modeling and prediction, *Information Sciences*, 178, 3618-3634, 2008, doi:10.1016/j.ins.2008.05.013.
- [30] DARPA 1998 MIT Lincoln Laboratory, <http://www.ll.mit.edu/IST/ideval/data/dataindex.html>, 1998.