Intrusion Detection Model based on Differential Evolution

M. Sailaja
Associate Professor
HOD ECE,
JNTUK College of Engineering

R. Kiran Kumar
Associate Professor
CSE Dept.,
Krishna University,
Machilipatnam

P. Sita Rama Murty
Assistant Professor
IT Dept.,
Sri Sai Aditya Institute of Science and Technology,
Surampalem

ABSTRACT
Information systems need to be constantly monitored and audited; analysis of security event logs in heavy traffic networks is a challenging task. In this paper we considered Differential Evolution for the intrusion detection problem. We used NSL_KDD dataset for our experiments which is derived from the standard KDD CUP 99 Intrusion Dataset. We also provided the comparative results of the differential evolution with the state of the art classification algorithm like SVM. We reduced the dimension/features of the NSK_KDD datasets using rough set algorithm and ran DE and SVM this increased the speed of the evolutionary algorithm without compromising the detection rate.

General Terms
Intrusion Detection Systems, Optimization algorithms

Keywords
Common Intrusion Detection Framework (CIDF), Differential Evolution (DE), Support Vector Machines (SVM),

1. INTRODUCTION
Information security has become so crucial today that organizations are spending millions of dollars to secure their classified data. Information security is a complex task, and it is a continuous process, hackers/ attackers are coming up with new/modified and improved attacks every day. There is a need to establish the comprehensive information security policy within all organizations. This is to ensure the confidentiality, integrity and availability of the vital corporate and customer information. Information security is provided using a defense in depth strategy. Defense in depth is a layered approach, with prediction, prevention and detection mechanisms implemented as different layers.

Customer and corporate data have become hot commodities. To protect the sensitive data we need to implement Multi-Layered security, the first layer of defense is firewalls, cryptography, password protection etc., and intrusion detection and prevention act as the second layer of defense.

Intrusion Detection Systems (IDS) monitors the network traffic and finds the connections that deviate from the regular profile and these are indentified as attacks. IDSs are good but these alone are not sufficient for providing security. When we compare firewalls and IDSs; firewalls will prevents the attack, and once a firewall is perfectly configured the network administrator need not think of it. IDSs will detect the intrusion, and it requires human interference to check the log of IDSs and take a remedy action to decrease the loss. IDSs can be configured as an access control device i.e., it can prevent attacks by sending the TCP Reset, or closing the connection whenever it finds any suspicious connection.

IDSs can be classified as Network based IDSs and Host based IDSs. Network based IDSs monitor packets on the network and attempts to discover intrusions. Host based IDSs base their detection of the information obtained from a single host.

Differential Evolution (DE) Algorithm is a latest evolutionary optimization technique, which is population based, powerful and robust. DE is applied in image classification, optimization problems and various other problems. To the knowledge of the author this algorithm is not used for intrusion detection up to now.

The rest of the paper is organized as follows; Section II contains related work where we presented about IDSs and DE. Section III contains information about the Experimental setup, data pre-processing, and experimental results. Section IV contains conclusion and future work.

2. INTRUSION DETECTION SYSTEMS (IDS)
IDS continuously monitor the network packets and whenever it encounters abnormal connections it will counter either by sending an email to the network administrator or by raising an alarm. IDSs can also be configured in such a way that they can prevent access to the suspected connections. Intrusion detection techniques are classified as anomaly detection and misuse detection. Anomaly detection signals deviations from the normal behavior as intrusions, and this is an unsupervised method which can detect previously unknown attacks. Misuse detection matches the connection patterns with the attack patterns (known signatures) and whenever a match occurs it is signaled as intrusion, this technique fails in detecting new attacks.

IDS researchers have proposed different approaches for intrusion detection. Techniques from domains like machine learning and data mining are rigorously being used for intrusion detection.

2.1 Components of IDS (CIDF)

Several IDSs are being used world-wide, of three major components/ modules; Data Collection, Intrusion Detection Engine, and Response module. CIDF workgroup has been formed and they modeled Components of Intrusion Detection Framework; CIFD framework contains four major components (1) E-Boxes (2) D-box (3) A-box (4) C-box; the components of this model and the relationship among those components is clearly shown in Fig 1.

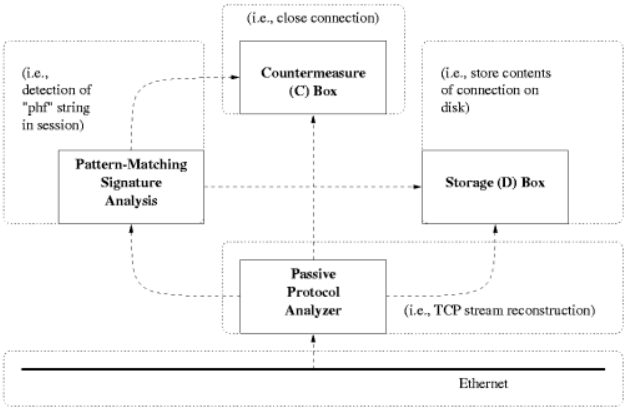

Fig 1: Components of CIDF[6]

2.1.1 Data Collection (E-boxes)

To achieve high accuracy and low false positives we need to have reliable and complete data, which tells about the significance of the E-box. The purpose of E-box is to collect information about security events and to provide the same to the rest of the system. Collecting all information is expensive, and the key is collecting the distinguished information.

2.1.2 Data Storage (D-box)

The data collected need to be stored, D-box is the means for storing the collected data.

2.1.3 Intrusion Detection Engine (A-box)

A-box is the crucial component, where event data is analyzed, for finding anomalous data. Lot of research is going on IDE since last two decades. IDE identifies the intrusions by analyzing data that is collected at E-box. In late 70's and early 80's all the log data is collected is printed to be audited by the reviewers for the unusual or malicious behavior. It is very tidy and fatigue duty to audit printed log data piled up for four to five feet. This is tidy job is now taken care by data mining or machine learning algorithms.

2.1.4 Decision Engine (C-box)

How should the IDSs respond when an intrusion is identified? Report to the network administrator or take action. This completely depends on the security policy made by the organization. IDS can be configured such that they can react aggressively when they detect an intrusion by blocking the attackers address or even attacking the culprit.

2.2 Performance Metrics

The performance metrics for machine learning algorithms are Accuracy, Precision, Recall, and Roc. Confusion matrix is something which is very helpful in calculating performance metrics. Model for the confusion matrix is given in Fig 2.

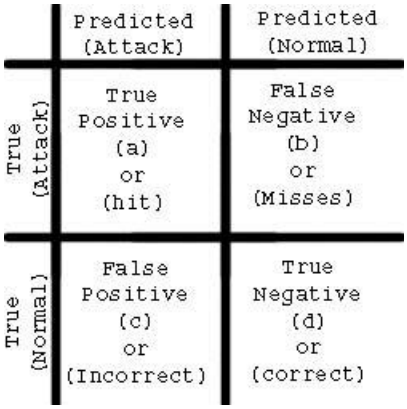

Fig 2: Confusion Matrix

True Positive (TP): An attack record/connection being correctly classified as an attack. This is also called as hit.

False Positive (FP): A normal record/connection incorrectly classified as attack.

True Negative (TN): Normal records correctly identified as Normal.

False Negative (FN): Attack connections incorrectly classified as normal.

Sensitivity: Sensitivity measures the proportion of actual positives which are correctly identified as such.

Sensitivity = TP/(TP+FN)

Specificity: Specificity measures the proportion of negatives which are correctly identified.

Specificity = TN/(TN+FP)

3. BACKGROUND

In this paper we considered Differential Evolution algorithm in Intrusion Detection Engine, we compared the results with the state of the art classification algorithms like SVM, Random Forest, etc.

3.1 Differential Evolution (DE)

DE is introduced by Storn and Price in 1997 [1]. DE is a Population based powerful and robust algorithm for solving real world global optimization problems. DE belongs to the family of Evolutionary Algorithms and these are very much like Genetic algorithms. DE uses operations like crossover, mutation, and selection on population to minimize the objective function over the course of successive generations. Initially population (NP) is selected randomly in such a way that it covers entire search space.

$$X = X_3 + w(X_1 - X_2)$$

Where the last term (x1-x2) is the mutation step size; 'w' is the scale factor and effective values are less than 1.

With every generation a "new parameter vectors are generated by adding the weighted difference between two population vectors to a third vector" [2].

3.1.1 Mutation Operator

In Mutation Operation a mutant vector is generated with respect to each individual in the current population. Based on the method of creating this mutant vector various schemes of DE are proposed. To classify different variants the notation used is DE/x/y/z, where 'x' is vector to be mutated; 'y' is the number of different vectors used; 'z' denotes the crossover scheme.

3.1.2 Crossover

To increase the diversity, Crossover phase is applied, following the mutation phase. This is a discrete recombination phase where elements from the parent vector are combined with elements from the mutant vector to produce the offspring.

3.1.3 Selection

DE implements greedy selection, i.e., the generated offspring replaces the parent only if the offspring is superior to parent otherwise the parent will remain.

As the dimensionality of the search space increases the performance of the evolutionary algorithms will decrease. It is recommended to reduce the features/dimensionality to speed up the algorithm convergence at the same without losing the efficiency. A proper tradeoff should be maintained between the dimensionality and details of the data.

3.2 Feature Selection

As the dimensionality of the search space increases the complexity of the optimization problem increases. The dataset we have considered has 41 attributes and a label. Differential Evolution algorithm takes long time to converge.

As the number of features increases it add detail to the problem but after some extent it will lead to confusion, and a problem of over-fitting may also occur. To avoid this we use feature selection.

Feature selection is a process in which a best subset of features is selected from the original features according to the objective function. Jin et al [3] suggests correlation coefficient between fields proposing that if the correlation fields of i and j is larger than 0.8, then there is a strong correlation and it is enough if we select one of them. We reduced the dimensionality of this dataset from 41 attributes to 8 attributes. These attributes are service, duration, scr_bytes, dst_bytes, count, srv_count, dst_host_srv_count, dst_host_diff_srv_rate, serror_rate and dst_host_same_src_port_rate [4].

4. EXPERIMENTAL SETUP

We conducted simulations on NSL_KDD dataset (derived from KDD CUP'99 Intrusion Dataset). The taxonomy of attacks in NSL_KDD 20% dataset is shown in Fig 3. In the dataset we considered for this experiment we considered only two classes Anomaly and Normal. All the attacks like Probe, Dos, R2L, U2R are labeled as Anomaly.

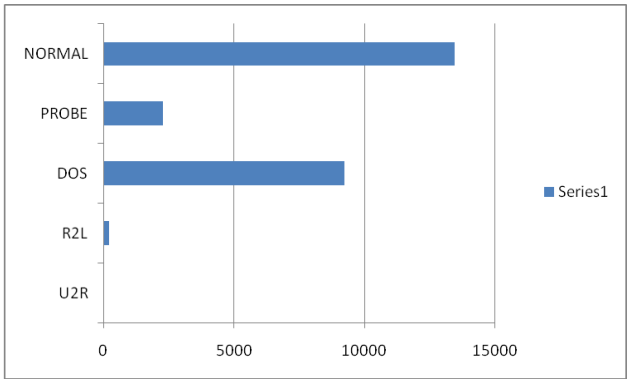

Fig 3: Taxonomy of attacks in NSL_KDD Dataset

KDD CUP'99 Intrusion Dataset is treated as standard dataset and has been used by researchers to analyze Intrusion detection engine (A-box). In this data set every connection has 41 attributes. This data set contains four major attack classes, which were sub-classified into 22 different attacks, and a normal class. The attacks were classified as Probe, DOS, U2R, and R2L.

KDD CUP'99 Intrusion Dataset was created by processing the Tcp/Ip-dump data portions of the 1998 DARPA IDS evaluation database, created by MIT Lincoln Labs. This dataset is extensively used by the researchers for over a decade and some researchers questioned the validity of this dataset criticizing that this synthetic dataset doesn't represent real world data. Tavelli et al in his paper [2] suggested NSL-KDD dataset that solved some of the inherent problems of the KDD CUP 99 Intrusion dataset. According to this one of the major deficiencies of the KDD intrusion dataset is its redundant records, which may bias the detection capability of learning algorithms.

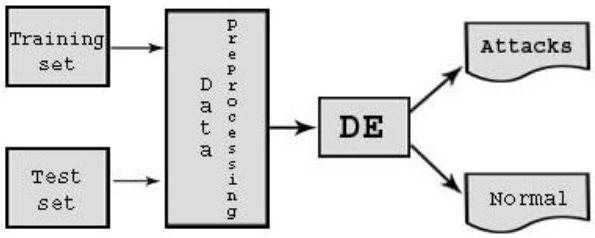

Fig 4: Overall Structure of the Proposed Method

In Fig 4, we have given the structure of the proposed method, where we are using DE algorithm in Intrusion Detection. We ran simulations using NSL_KDD 20% training dataset which contains 25,192 connections/records.

We ran the algorithm by taking the population (number of particles) as 10^2 i.e., 100; Number of iterations 200; 10 fold cross validation, and fitness function: Sens*Spec, DE algorithm with this fitness function has yielded best results.

We also did the experiment on Support Vector Machines using LIBSVM [6] package with these parameters: SVM type is C-SVC, kernel function is Radial Basis Function, 10 Fold Cross Validation. The results were given in Table 1.

PC Configuration- All the algorithms are run on Intel Core i3 CPU with 3GB Ram. The programming language used is Java.

Table 1. Results of Experiments

	NSL_KDD			NSL_KDD(reduced features)	
	Accuracy	**FP Rate**		**Accuracy**	**FP Rate**
DE	95.78	1.6		92.12	0.4
SVM	95.35	4.2		95.31	4.5
RBF Network	92.67	4.3		88.91	10.1

5. RESULTS

The experimental results show that Differential Evolution has yielded better detection rate and low false positive compared with SVM and RBF Network for NSL_KDD Dataset and SVM has shown good Detection Rate in reduced dataset; numerical results are shown in Table 1.

6. CONCLUSION

The numerical optimization algorithm Differential Evolution can be applied for intrusion detection. Sometimes high dimensionality leads to reduced performance which is known as "curse of dimensionality", so it is necessary to reduced the features, here we used rough set theory to reduce the feature set, which reduced the features at 1:4 ratio without effecting the detection rate of the considered algorithms.

7. REFERENCES

[1] R Storn and K Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization 11: 341-359, 1997.

[2] Xiaobu Liu, Chao Yu, and Zhihua Cai., Differential Evolution Based Band Selection in Hyperspectral Data Classification, ISICA 2010, LNCS 6382, pp. 86-94, 2010.

[3] Mahbood Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani., A Detailed Analysis of the KDD CUP 99 Data set, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[4] H. Jin, J. Sun, H. Chen, and Z. Han., A Fuzzy Data Mining Based Intrusion Detection System, Proceedings of 10thInternational Workshop on future Trends in Distributed Computing Systems (FTDCS04) IEEE Computer Society, Suzhou, China, May 26-28, 2004, 191-197.

[5] Surat Srinoy., Intrusion Detection Model Based on Particle Swarm Optimization and Support Vector Machine, Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2007), 186-192.

[6] H. T. Ptacek and N. T. Newsham., Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection; Secure Networks, Inc., January 1998.

[7] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ ~cjlin/libsvm.