Intrusion Detection System Using Data Mining Technique

Naelah Okasha
Computer Science Dept.,
Higher Institute for Computer & IT, Modern Academy, Cairo, Egypt.
e-mail: n1_okash@yahoo.com

Sherif M. Badr
Computer Science Dept.,
Higher Institute for Computer & IT, Shorouk Academy, Cairo, Egypt.
e-mail: sherif_badr@afmic.com

Magdy El-Hennawy,
Computer Science Dept.,
Higher Institute for Computer & IT, Shorouk Academy, Cairo, Egypt.
e-mail: melhennawy@ad.gov.eg

Abstract: *Intrusion detection is an approach for providing a sense of security in existing computer systems and data networks allowing them to operate in their current "open" mode more securely. An intrusion detection system (IDS) inspects all inbound and outbound network activities and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise the system. The goal of intrusion detection, then, is to identify, preferably in real time, unauthorized use, misuse, and abuse of computer systems and data networks by both system insides and external penetrators.*

Nowadays new intelligent techniques have been used to improve the intrusion detection process in computer networks. This paper presents an approach of an adaptive multi-level intrusion detection and prevention system supported with a hybrid intelligent system based on data mining for classification and pattern recognition. We have specified attack signatures, reaction with event communication and correlation that are integrated on the system, incorporating supervised and unsupervised modes, and generating intelligent reasoning.

Keywords**:** Intelligence security, Intrusion detection and prevention, Data mining, Classifier.

1. Introduction

The study of security in computer networks is a rapidly growing area, since the increased connectivity of computer systems gives greater access to outsiders, and makes it easier for intruders to avoid detection. IDSs are based on the belief that an intruder's behavior will be noticeably different from that of a legitimate user. Consequently, network attacks or intrusions such as eavesdropping on information meant for someone else, illegally accessing information remotely, and breaking into computers remotely can be recognized. Several proposals suggest methods that can prevent network attacks in closed systems, e.g., encryption techniques. But

such solutions are not suitable for infrastructure of open data networks, also it cannot protect against stolen keys or legitimate users misusing their privileges [1].

Intrusion detection is defined to be the problem of identifying individuals who are using a computer system without authorization (i.e., *crackers*) and those who have legitimate access to the system but are exceeding their privileges (i.e., the *insider threat*) by monitoring the events occurring in a computer system or data network and analyzing them for signs of possible incidents. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents [2].

Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and attempting to stop and report them to security administrators. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

Network intrusion detection systems (NIDS) and network intrusion prevention systems (NIPS) play a critical role in detecting and dropping malicious or unwanted network traffic. These have been widely deployed as perimeter defense solutions in enterprise networks at the boundary between a trusted internal network and the un-trusted Internet. This traditional deployment model has largely focused on a single-vantage point of view of NIDS/NIPS systems, placed at manually chosen (or created) checkpoints to provide coverage for all suspicious traffic [3].

The main objective of this work is to propose security architecture of an intrusion detection and prevention system for computer systems and data networks. This proposed system could be positioned at the network mailing server to monitor all passing data packets and determine suspicious connections. The proposed system should have a pre-knowledge about normal users behaviors as well as the different types of attacks and the corresponding set of suggested actions against each attack type. Therefore, it can inform the system administrator with the suspicious attack type and the corresponding suggested actions. Moreover, the proposed system should allow for new attack types to be defined, i.e. the proposed system should have an adaptive capability.

2. The Proposed System Framework

This paper introduces an architecture for an adaptive multi-level intrusion detection and prevention system for computer systems and data networks called Adaptive Multilevel Intrusion Detection & Prevention System (AMIDPS). AMIDPS augments the network security monitoring technique with a data mining technique to help classifying the network users into either normal users or intruders [4].

Key innovations of the introduced AMIDPS include its capability to learn to enhance its capability, and alarm the system administrator when the system is attacked. Moreover, the proposed AMIDPS has three levels of detection; Boolean, coarse, and fine detection levels.

The introduced AMIDPS exploits the historical user behaviors of the target system's audit trails. These historical behaviors are used to train the system's artificial training module with the most dominant features of these audit trails. The trained system can recognize the network or system accesses as normal or intrusion profiles. The next section presents the main components of AMIDPS model.

3. AMIDPS Architecture Overview

The proposed AMIDPS has four basic components, where the order of these components reflects the AMIDPS flow of work, as shown in Figure 1.

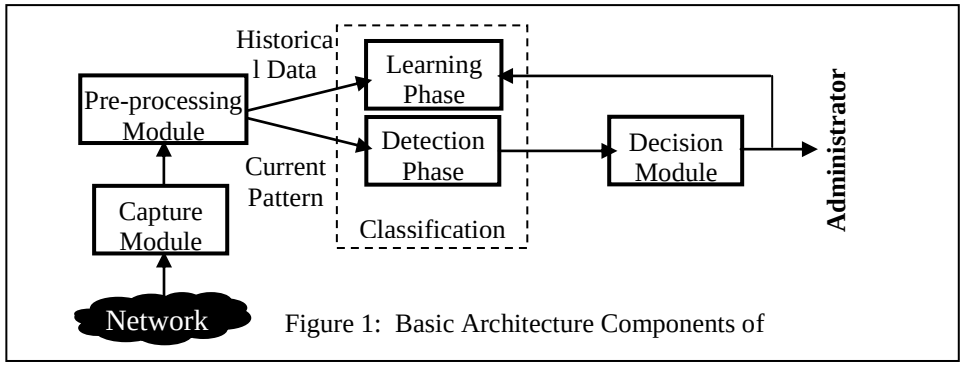

Figure 1: Basic Architecture Components of

The four components of AMIDPS are [4]:

- **The Capture Module:** through which a set of all received network packets are captured and stored. In this module a large number of audit data is captured using a network monitoring and capturing tool.
- **The Pre-processing Module:** where the captured packets are pre-processed to extract the most dominant features that represent user profiles. These extracted features are then used as inputs to the learning phase in classification module.
- **The Classification Module:** this module involves two different phases, the learning phase and the detection phase. In the learning phase, the classifier uses the pre-processed, captured network user profiles as input training patterns. In the detection phase, the classifier is used on-line to detect network intruders. The historical data describing various profiles are used to evaluate the current detected pattern, whether normal or not. To do this, the current detected pattern is compared with historical patterns.

- **The Decision Module:** once an attack is detected by the classification module, the AMIDPS decision module will trigger an alarm with the recommended and appropriate action to the system administrator. The set of actions is determined based on the system capability and its management's policy. AMIDPS decision module has two modes of operations, a fully automated mode where AMIDPS defines the recommended and appropriate action without the intervention of the system administrator. The second mode of operation is the semi-automated mode in which the system administrator will be provided with most probable actions each with a certainty factor, where he can select the recommended and appropriate one.

3.1 The Capture Module

AMIDPS utilized the capabilities of the TCP dump capture utility for Windows to gather historical network packets. It is an architecture that adds to the operating systems of the Win32 family the ability to capture data of a network (dump traffic on a network) using the network adapter of the machine. Moreover, it provides a set of APIs that can be used to facilitate justification requirements using low-level capabilities [5][6][7].

3.2 The Pre-Processing Module

The AMIDPS pre-processing module is responsible for reading, processing, and filtering the audit data to be used by the classification module. The pre-processing module maps the raw packets captured from the network by the TCP dump capture utility, to a set of patterns of the most Effective Selected Feature (ESF) which refers to the set of the non-correlated features (remaining feature). So the main function of the pre-processing module is to supply the classification module by a set of data to be used directly in the learning or detection phase. The pre-processing module consists of three sub-components:

- **Coding component:** if the classification module deals only with numeric data, so any symbolic data in the captured packets must be coded and transferred to numeric form. This is achieved by creating a transformation table containing each text/string feature and its corresponding numeric value. The component that performs this task is called coding component.
- **Correlation component:** the correlation component of AMIDPS is used to reduce the dimensionality of input features of the classification module. Reducing the input dimensionality will reduce the complexity of the classification module, and hence the training time [8]. The correlation component is a statistical module that analyzes the available data sets and identifies "certain" correlation rules between different input pairs. Out of each two correlated pairs of input features only one feature will remain. In addition, it extracts some statistics regarding each input feature. The set of the non-

correlated features (remaining feature) is called the Effective Selected Features (ESF) [9].

- **Scaling component:** since all the selected features are the most important ones that identify user profiles and none of them has a priority over the other, therefore, it is assumed that they all should have the same effect on the classification module at both the training and the testing phases. To make this assumption true, all inputs to the classification module should be scaled to fall between zero and one (0,1) (Normalization). The pre-processing module does this by the scaling/normalization component which guarantees equal range for all of the classification module input features [10]. The final output of the pre-processing module components is a subset of the captured data (packets) that will be used as inputs to the classification module.

3.3 The classification Module

The AMIDPS classification module has two phases. The learning phase (which uses the captured and then pre-processed historical data set) and the detecting phase (which uses the on-line/current captured and then pre-processed input patterns). A limited number of researches have been conducted on using data mining to detect computer intrusions.

This process takes the output of ESF which is the final feature vector and uses any classification technique to classify this vector according to the pre-learning of this classifier. The output of this process is the status (normal or intruder type). Many classification techniques can be used in the process of classifying network attacks [11][12][13].

3.3.1 The Learning Phase

Data mining is the process of finding patterns and relationships in data. At its core, data mining consists of developing a model, which is typically a compact representation of patterns found using historical data, and applying that model to new data. A model is applied to data to predict individual behavior (classification and regression), segment a population (clustering), determine relationships within a population (association), as well as identifying the characteristics that most impact a particular outcome (attribute importance) [11] [14][15] [16]. Since classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown, then the derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).

3.3.2 The Detection Phase

Once the classifier is learned, its capability of generalization, to correctly identify the different types of users, should be utilized to detect any intruder. This detection process can be viewed as a classification of input patterns to either normal users or

intruders [11][13]. The classifier then receives the numeric value that represents the user profile of the current connection(s) and classifies it/them. From this point, by the decision module, many operations can be carried out, such as the transmission of many levels of alerts or warning to the system administrator (depending on the risk level of the detected events), shot logging processes, and activation of counter-measures to isolate the host or the domain that caused the attack. During the detection process, some deviation from the right decisions may take place. The deviation of an attack behavior from the normal behavior is controlled by a threshold, which is a numerical value level to prevent a false positive occurrence. A false positive occurs when the system classifies a legitimate action as anomalous (a possible intrusion) . The threshold is also used to prevent a false negative occurrence. A false negative occurs when an actual intrusive action has occurred but the system allows it to pass as non-intrusive behavior [17].

3.4 The Decision Module

AMIDPS decision module has two modes of operation, where the operation mode selection is controlled by the system administrator. These two modes of operation are:

- The fully automated mode: where AMIDPS provides the system administrator with the intruder (attack) type and the corresponding action.
- The semi-automated mode: where AMIDPS presents the N^{th} most probable intruder (attack) type along with the corresponding recommended and appropriate actions, where the selection of the possible decision's number (N) is determined by the system administrator. In this mode of operation, each possible output decision is presented to the system administrator along with a certainty factor indicating the truth level of this decision. Regardless of the operation mode, the basic responsibility of the decision module is to send information to the system administrator about how close a compromise is to being achieved or whether a compromise has been achieved according to the rules stored in a decision table. This gives the system administrator the ability to monitor the progress of the detection phase. For each attack type there is a message (action) kept in a structure called the decision table. This table adds further modularity to AMIDPS by allowing the system administrator to modify AMIDPS's responses to each detected attack.

4. AMIDPS Architecture Description

4.1 Input Dataset

Building an efficient AMIDPS require precise analysis of existing data sets describing the behaviors of anomalous attacks. The most famous dataset is DARPA dataset. This is the dataset used for the 3^{rd} international knowledge discovery and data mining tools competition. The competition task was to build a network

intrusion detector, a predictive model capable of distinguishing between bad connections, called intrusions or attacks, and "good" normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment [18][19]. The dataset consists of 41 features and status; the 41 features are classified under 3 categories:

- Basic features of individual TCP connections.
- Content features within a connection suggested by domain knowledge.
- Traffic features computed using a two-second time window.

The DARPA intrusion detection evaluation program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate researches in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The process of building the classifier must initially start with data inspection in order to visually and statistically inspect the data set before building an effective classifier [20][21].

4.2 Data Mining:

Classification and prediction are the major techniques in data mining and widely used in various fields. In this work we present how some problems can be solved using classification and prediction techniques in data mining. By using this approach, the model performance can be predicted by using past experience knowledge discovered from the existing database. In the experimental stage, performed to test and evaluate our model, we have used selected classification and prediction techniques to propose the appropriate techniques from our training dataset. Thus, by using the experimental results, we suggest the potential classification.

In data mining, trees can be described as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data. Data comes in records of the form:

$$\mathbf{(x, y) = (x_1, x_2, x_3..., x_k, y)}$$

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector **x** is composed of the input variables, x_1, x_2, x_3 etc., that are used for that task.

4.3 Selection Criteria of Decision Tree Classifier

The selection of decision tree in our model was intended after a wide survey in data mining field. The conclusion of this search lead us to the conclusion that the decision tree is the appropriate choice, Decision trees seem now to be the best choice for building our classifier since it deals with both integer and real numbers.

Another important and interesting feature in decision trees is the ***IF Then*** nature rules which are simple to be implemented [22][23]. Decision trees, however, are powerful and popular in knowledge discovery and data mining. Decision trees are used in exploring large and complex bodies of data in order to discover useful patterns.

4.4 Model Building Using SPSS Clementine

Using SPSS Clementine package with the decision trees classifier (C5.0 algorithm classifier) help in the following:

- Input features can be discrete or continuous data.
- Input features can be numeric and symbolic data.
- All input features have equal weight range with same effect.
- Reduce the dimensionality of input features, to reduce the complexity of the stream architecture by using the correlation technique.

SPSS incorporation is a leading worldwide provider of predictive analytics software and solutions. Founded in 1968, today SPSS has more than 250,000 customers worldwide, served by more than 1,200 employees in 60 countries. SPSS has released a data analysis and modeling tool called PASW modeler (formerly Clementine), which is a rich tool capable of building powerful data statistics and modeling functionality. Some basic features of Clementine is the ease of use through its visual interface, time saving, proven performance, statistics integration, automatic data preparation and documentation of the thoughts or processes used in the creation of a model, and efficiently sharing these with others in the organization.

5. Data Analysis and Experimental Results:

The function of our network-based model (AMIDPS) is to monitor, classify and detect attacks in traffic passing to or from the network. That network attacks uses harmful network connections to harm the victim machines. So, traffic passing through the proposed model must be inspected in all lower layers before forwarding it to the target machine in order to protect them. We have used SPSS Clementine as data analysis and modeling tool to build C5.0 decision tree to work as a classifier for network intrusions and attacks. We applied 10% noise over data to improve classification. The generated model consists of 70 rules. The default rule is normal (final else after all if conditions).

5.1 Clementine Stream:

The first step in data modeling is to recognize and understand the data. Here we have tried to reveal as much information about the features and its relations to the output, as possible (whether connection status is normal or attacks) [24].

To enable SPSS Clementine to build stream to inspect data source used in building Decision Tree, calculate various statistical measures like count in the sample (probability) distribution, then the mean, variance, etc… have to be calculated. Figure 2 represents the stream has been built in SPSS Clementine to perform statistical inspection of data.

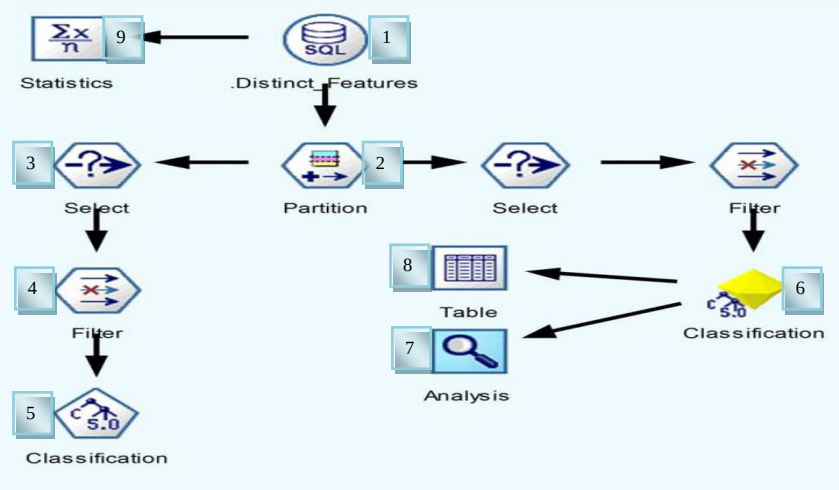

Figure 2: SPSS Clementine stream.

The explanation of the flow of operation described in Figure 2 is as follows:

1. SQL Node which connects to ACCESS database file features containing the feature records to be statistically inspected.
2. Partition Node which split the data into two separate subsets (Training and Testing).
3. Select Node which selects a subset of records according to a fixed condition either the training part or the testing part.
4. Filter Node which is used if it is needed to remove unwanted record.
5. C5.0 Node which is a node used to build a predictive decision tree.
6. C5.0 Node which is the model which had been trained and now it is going to be used in testing
7. Table Node which is used to view and display the output in a tabular form.
8. Analysis Node which is used to show a report after executing the model.
9. Statistics Node responsible for creating statistical report about data source like mean, standard deviation, skewness.

5.2 Assumptions: (general)

The data set was organized as records, each record represents one TCP/IP connection; a connection is a sequence of TCP/IP packets. Each connection has 41 features labeled as either normal, or as an attack, with exactly one specific attack type. Based on this data set, attacks can be categorized into four main categories. This categorization was based on the distinct evidence that intruders leave when they connect to the network. Attacks four main categories can be:

- Denial of Service Attacks.
- User to Root (U2R) Attacks.
- Remote to User (R2U) Attacks.
- Probing (Prob) Attacks.

The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic, which is equivalent to about five million connection records. Similarly, the two weeks of test data, corresponds to about two million connection records. A reduced version of this raw data of about 10% was selected. These data were used for building the detection model and for testing the generated model, other data files of different number of connection records were used for testing the classifier performance (Testing Dataset).

Training the classifier was carried out using eighteen features from forty one features in the datasets which relates to windows operating system which reduce the data set from 438,331 records to 76,604. This can happen by removing all redundant records to build this rule set. A correlation process was done between individual fields of the windows data for the eighteen features to reduce the data set size and the training time which reduces the input features to fourteen only. To evaluate any IDS, there exist four main cases that must be taken into consideration as shown in Table 1.

Table 1: Main Four Cases To Evaluate Any IDS

	Intrusion	No Intrusion
IDS Alarm	An intrusion has occurred and the IDS have generated an alarm. (Correct alarm)	No intrusion has occurred, but the IDS have generated an alarm. (False alarm)
IDS Rejection	An intrusion has occurred but IDS have not generated an alarm. (False rejection)	No intrusion has occurred and the IDS have not generated an alarm. (Correct rejection)

Related to these four cases, two parameters of IDS are defined as follows **[25][26]:**

- The *accuracy* of IDS: which equal the number of *correct alarms* divided by the number of *correct alarms plus false alarms.* The more accurate IDS is, the fewer false alarms it generates and the higher this parameter is.
- The *completeness* of IDS: which equal the number of *correct alarms* divided by the number of *correct alarms plus false rejections.* The more complete IDS is, the fewer intrusions remain undetected and the higher this parameter is.

In the ideal case, an IDS would be 100% complete (it detects all intrusions) and 100% accurate (it produces no false alarms). Good IDS must minimize the misclassification rate (number of false positive and false negative). When an IDS classifies a normal connection as an attack (false positive), the harm is less than classifying an attack as a normal connection (false negative). In addition, wrongly classifying one attack as another attack (wrong types detected attacks) is still better than classifying this attack as a normal behavior. The experimental work has used the following samples of data:

- Two groups of data sets (DS), the first group use the eighteen features which are related to the windows OS to build the model and test it, the second group use fourteen features only (correlated) from the dataset to build the model and test it.
- Two subgroups of data sets for each of the above two groups (18, 14). In the first one, all records are of DARPA data sets which is related to Windows (76,604 records) but the attacks' numbers are not equal (*real environment*), the second one contains 100 records for each 22 attacks and equal number of normal resulting about (2200) to get data sets of 4400 records (*equal chance*).
- The training is done with the percentage 10%, 50%, and 90% respectively of each of the above 4 groups.
- The testing is processed over the reminder of each data set or all the data sets.
- All the data sets doesn’t contain any features with summation zero.(not clear feature).

Table 2 summarizes the experimental input data sets.

Table 2: Experimental Input Datasets.

DS No.	Features No.	No of Record	Training %	Test %
1	18	76,604	10	90
2				100
3			50	50
4				100
5			90	10
6				100
7		4400	10	90
8				100
9			50	50
10				100
11			90	10
12				100
13	14	76,604	10	90
14				100
15			50	50
16				100
17			90	10
18				100
19		4400	10	90
20				100
21			50	50
22				100
23			90	10
24				100

5.3 Results:

To get the best possible performance, 72 Clementine streams has been built according to the DSs stated in Table 1, all DSs were trained, and their performance (correct classification rate and attack miss ratio) was measured by two artificial intelligence techniques, they are:

- Data mining with decision tree (two models used C5.0 and CRT) and
- Neural network.

All experimental results for the different techniques/models of all the DSs show superiority C5.0 over all DSs (14 & 18 features and real environment & equal chance) as shown in Figure 3 and Figure 4. For C5.0 model (the best results) the IDS parameters (correct alarm/false alarm/false rejection/correct rejection) and also

the *accuracy and completeness* of IDS are calculated and summarized in Figure 5 & Figure 6.

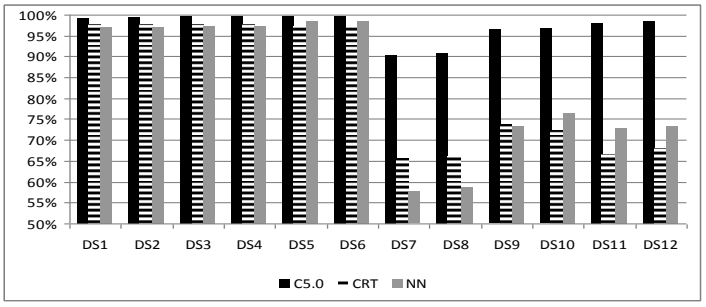

Figure 3: Classification Ratio 18 Features

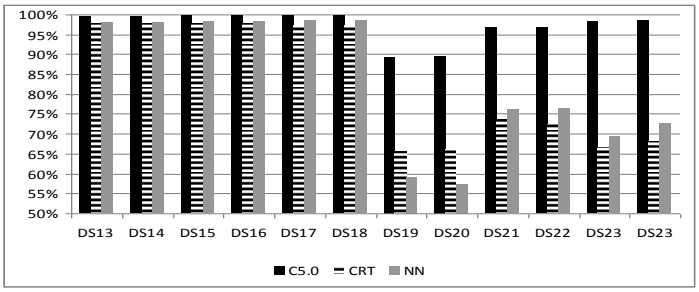

Figure 4: Classification Ratio 14 Features

For 100% completeness and accuracy, **Figure 5** and **Figure 6** show that this occurs only for DSs 11, 12, 23, 24. The DSs 11, 12 are the same as DSs 23, 24 after correlation (18 input features reduced to 14 features), but the correlated input features classification ratio is much better than that without correlation, so the increasing of input features mislead the training and testing.

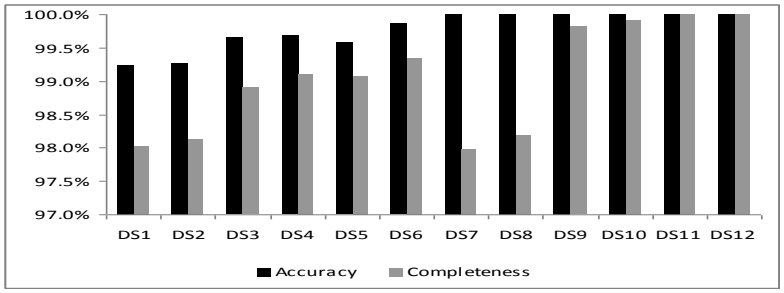

Figure 5: C5.0 Classification Ratio 18 Features

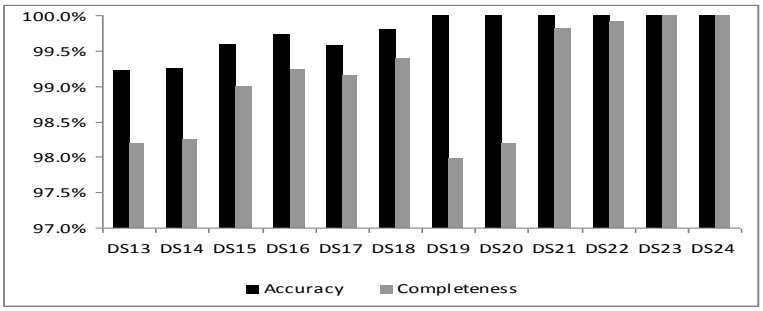

Figure 6: C5.0 Classification Ratio 14 Features

The results of using correlated features (used for smaller memory size and for training & testing time); has been analyzed to show the classification ratio for each of the three detection levels. Figure 7, 8, and 9 show more details about the classification ratio of DS 12 & DS 24 for each attack types/categories.

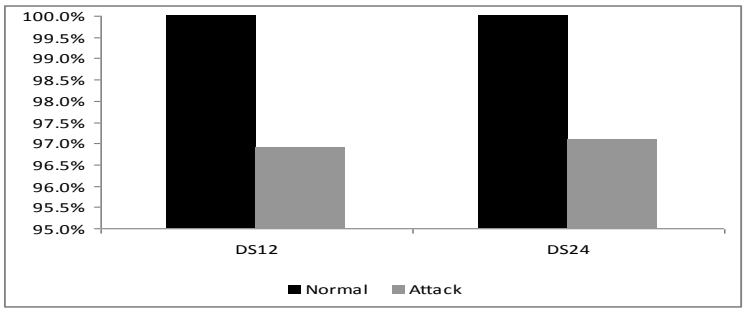

Figure 7: Boolean Level of DS12 & DS24

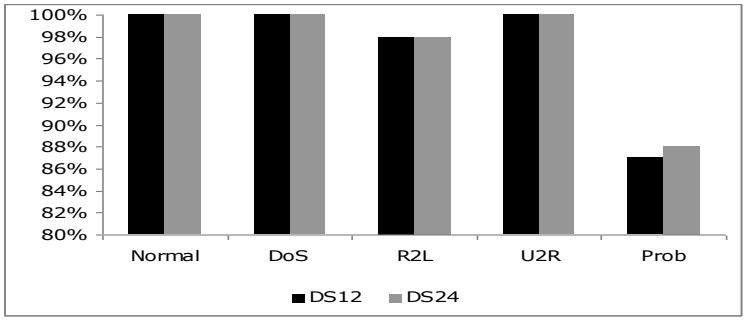

Figure 8: Coarse Level of DS12 & DS24

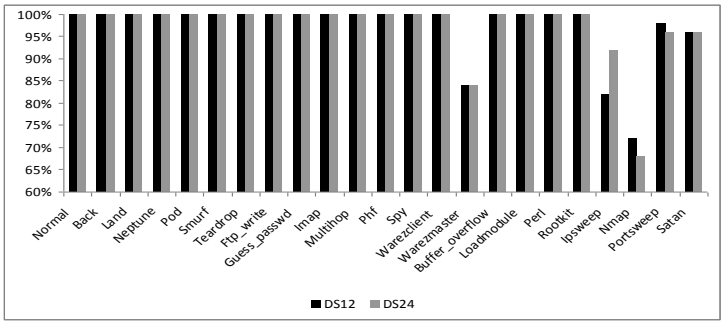

Figure 9: Fine Level of DS12 & DS24

Despite the classification ratio is high (over all datasets) but the classification ratio for the 4 categories is not same and also there is some attacks' type (fine level) in which the classifier failed to classify the behavior record whether it is normal or attack, or failed to determine the exact type of attacks.

6. Conclusion:

Intrusion detection is becoming a growing problem as computer networks grow and the dependency on them in human life increases. This paper introduces an intrusion detection model, named AMIDPS, which augments a network monitoring system with a data mining suite to help the system administrator take the recommended and appropriate anti-attacks actions. Moreover, it analyzes the different requirements and issues involved in the different phases of intrusion detection systems. AMIDPS adds set of significant features. It combines network monitoring techniques and data mining technique. Moreover, it utilizes the existing network capturing tools (e.g., TCP dump) without any additional overhead. In addition, it analyzes the captured audit records and performs a set of predetermined processes, such as coding of symbolic (textual) data into numeric data, reducing any redundant features of the captured audit data and only keeping the non-correlated features, normalizing all extracted features to be between [0,1] to avoid any bias due to the different ranges of input features. It can besides, selectively and based on the security level, be used to detect intruders in three different detection levels, which are Boolean detection level (either normal or intruder), coarse detection level (either normal or one of 4 intruder categories), or fine detection level (either normal or one of 22 intruder types). It provides the system administrator with the intruder category/type along with the appropriate action, AMIDPS classifier can be accommodated with any new normal and/or intruder user profiles whenever these profiles are known. Besides, this model allows new technique, which is the data mining with decision tree compared with the neural network technique.

References:

[1] Gustavo Isaza, Andrés Castillo, Manuel López, and Luis Castillo, Á. Herrero et al "*Towards Ontology-Based Intelligent Model for Intrusion Detection and Prevention*",. AISC 63, pp. 109–116.springerlink.com, Springer-Verlag Berlin Heidelberg, 2009.

[2] Juniper Networks "*Network and Security Manager, Configuring Intrusion Detection and Prevention Devices Guide*", Inc., California, USA, 2010.

[3] Vyas Sekar, Ravishankar Krishnaswamy, Anupam Gupta, Michael K. Reiter, "*Network-Wide Deployment of Intrusion Detection and Prevention Systems*", 2010.

[4] Sherif M. Badr "*Security Architecture for Internet Protocols*", Ph. D. dissertation MTC, 2001

[5] L. Todd Heberlein, "*Network Security Monitor*", Technical Report, Columbia University, 1995.

[6] Steven McCanne, B. Jacobsen, Craig Leres "*TCP dump*". ftp://ftp.ee.lbl.gov

[7] "*Win Dump: TCP dump for Windows*," Http://netgroup-serv.polito.it/windump/

[8] Yongseok P. "*Event Correlation*" IEEE potentials, vol. 20 , No. 2, p. 34-5, April-May 2001.

[9] L.Wenke, "*A framework for constructing features and model for intrusion detection systems*," Ph.D. thesis, Columbia University, 1999.

[10] W. Mendenhall, "*Introduction to Probability and Statistics*", fifth edition, Wadsworth Publishing Company, Inc.

[11] E. F. Ian H. Witten, "*Practical Machine Learning Tools and Techniques*", Morgan Kaufmann Publishers, second edition, 2006.

[12] S. S. J. LeeW. and M. K.W, "*A data mining framework for building intrusion detection models*," Proceedings of the 1999 IEEE Symposium on Security and Privacy, 1999.

[13] E. J. Ulf Lindqvist, "*How to systematically classify computer security intrusion,*" IEEE Symposium on Security and Privacy, May 1999.

[14] S. V. Mark F. Hornick, Erik Marcade, "*Java Data Mining: Strategy, Standard, and Practice,*" Morgan Kaufmann Publishers, 2006.

[15] J. Han and M. kamber, "*Data Mining:Concepts and Techniques*," Morgan Kaufmann Publishers, seconed ed., 2006.

[16] S. Landau and B. S. Everitt, "*A Handbook of tatistical Analyses using SPSS*," CHAPMAN and HALL/CRC, second ed., 2006.

[17] N. R. Pal and L. Jain, "*Advanced Techniques in Knowledge Discovery and Data Mining*," Springer, 2005.

[18] D. A. R. P. Agency, "*Darpa intrusion detection evaluation*," URL:http://www.ll.mit.edu/IST/ideval/index.html

[19] D. J. F. I. G. K. R. K. S. W. W. M. Z. R. P. Lippmann, R. K. Cunningham, "*Results of the 1999 darpa off-line intrusion detection evaluation*," Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection (RAID99), West Lafayette, 1999.

[20] "*Kdd-99*," The third International Conference on Knowledge Discovery and Data Mining URL:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[21] I. G. J. W. H. K. P. K. D. M. D. W. S. E. W. D. W. R. K. C. R. P. Lippmann, D. J. Fried and M. A. Zissman, "*Evaluating intrusion detection systems*," The 1998 DARPA Off-line Intrusion Detection Evaluation, Proceedings DARPA Information Survivability Conference and Exposition (DISCEX) 2000, Vol 2, pp. 12-26, IEEE Computer Society Press, Los Alamitos, CA, 2000.

[22] Mohamed Ibrahiem Amer, "*Intrusion Detection and Prevention System*,", master thesis, Military Technical Collage, 2009.

[23] J. T. Bjrn Schelter, Matthias Winterhalder, "*Handbook of time series analysis*".

[24] "*Intrusion detection generics and state of the art*", RTO Technical Report, 2002.

[25] K. A. El-Fatah, "*Multi Agent Intrusion Detection System*" master thesis, Military Technical Collage, 2006.

[26] E. A. M. K. C. Ozgur Depren, Murat Topallar, "*An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks*," Elsevier, 2005.