Performance Measuring Tool for Data Mining Techniques in Intrusion Detection System

by

Muhammad Firdaus Bin Roslan

Dissertation Report submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
Information Technology

July 2006

University Technology PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

PERFORMANCE MEASURING TOOL FOR DATA MINING TECHNIQUES IN INTRUSION DETECTION SYSTEM

by

Muhammad Firdaus Bin Roslan

A project dissertation submitted to the
Information Technology Programme
University Teknologi PETRONAS
In partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION TECHNOLOGY)

Approved by,

(Mr. .Hilmi Hassan)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
July 2006

CERTIFICATION OF APPROVAL

PERFORMANCE MEASURING TOOL FOR DATA MINING TECHNIQUES IN INTRUSION DETECTION SYSTEM

by

Muhammad Firdaus Bin Roslan

A project dissertation submitted to the
Information Technology Programme
University Teknologi PETRONAS
In partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION TECHNOLOGY)

Approved by,

(Muhammad Firdaus Bin Roslan)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
July 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MUHAMMAD FIRDAUS BIN ROSLAN

ACKNOWLEDGEMENT

First of all, I would lie to express my gratitude to Allah s.w.t for His blessings, giving me the opportunity to complete this final year project.

I would like to express my appreciation to Mr. Hilmi Hassan, as my supervisor for his comments and advices, which aims in guiding me towards completing this final year project.

To Mr. Nordin and Mrs. Aliza Sarlan, thank you for your coordination and support as the Final Year Project Coordinator (Jan 2006 and July 2006) throughout the completion of this final year project (Part I and Part 2).

My big gratitude goes to my supervisors, Pn. Arniyati Ahmad and Pn. Siti Rohaidah at MIMOS Berhad, for guiding me during the industrial internship, where I had gained knowledge about Data Mining and the Intrusion Detection Systems which is the main reference and guidelines for this project.

The research project had become the main idea behind the development of this project. A bunch of thanks for their time for their help and for their willingness to share their knowledge in the research project. Not to be forgotten, to all my friends and lecturers, thank you for the time spent in keeping me focus on completing this project.

Last but not least, I would also like to express a special thank to my parents and to all my family members for their priceless support, encouragement, constant love, valuable advices, motivation which heals me from pressure and sores through out the project completion.

ABSTRACT

The research project is about to develop a performance measurement tool for Data Mining (DM) techniques in Intrusion Detection System (IDS). Basically, IDS is a network security system that is used to detect cyber attacks intrusion. By applying the Data Mining technique it might improve its accuracy as well as its efficiency in intrusion detection process especially in a large and fast network. However, there are various kinds of techniques in DM that can be used to enhance the intrusion detection process in IDS such as K-mean clustering, Support Vector Machine (SVM), Self Organizing Maps (SOM), Neural Networks, etc. Therefore, a performance measurement is required in order determine the best DM technique to be used depending on the network environment and the type of the IDS used. The performance measurement takes place at the final stage of the Knowledge Data Discovery (KDD) process which a step by step procedure in implementing the DM techniques. With the help of this new tool, it can reduce the human intervention in performance measurement process as much as possible by replacing the manual tasks with the automated approach. As a result, errors due to human conducts can be reduced. This is because a slight of error might affect the overall performance results measured. The final results are so important that it is to be used in decision making of the implementation of DM technique in IDS. The tool is comprised of three main modules: confusion matrix analysis, calculation of the detection rates and the false alarm rates, and generating the ROC curves as the final result.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS & NOMENCLATURES

KDD	Knowledge Data Discovery
DM	Data Mining
SMO	Sequential Minimal Optimization
IDS	Intrusion Detection System
RAD	Rapid Application Development
ROC	Receiver Operating Curves

CHAPTER 1
INTRODUCTION

1.0 BACKGROUND OF STUDY

Basically the IDS is a network security system that detects cyber attacks or intrusions. Moreover the IDS works by detecting 'patterns' of known intrusions and also the 'unusual patterns' of possible intrusions in the analysed or monitored (live) network traffic. The Data Mining is used to improve the intrusion detection process in IDS. The significant of having the right Data Mining techniques is to enhance the ability of the system in detecting novel intrusions.

1.1 Intrusion Detection System (IDS)

Intrusion detection is the process of monitoring and analyzing the events occurring in a computer system, to detect signs of security problems. Intrusion Detection System (IDS) is a network security system that detects the cyber attacks intrusion. The IDS is a combination of hardware and software that is ensemble in such a way in order to perform the intrusion detection.

The diagram below is an example of a basic structure of typical IDS:

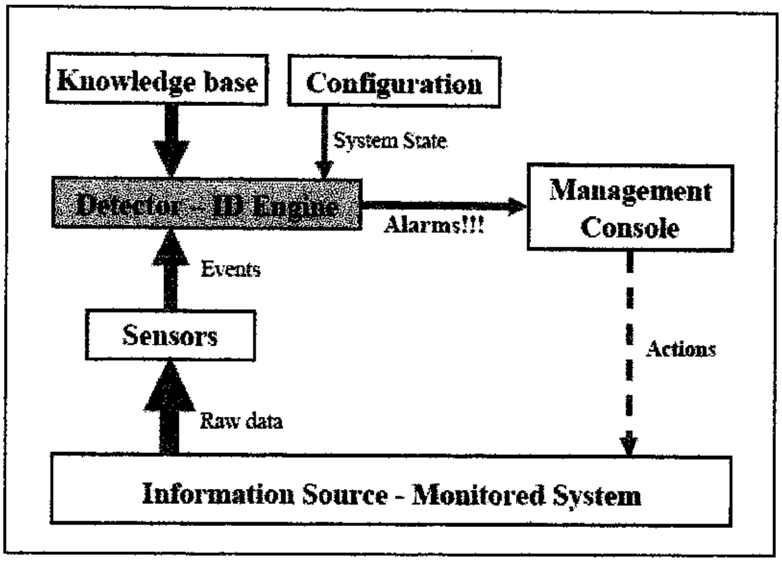


Figure 1: Basic structure of typical IDS

The basic mechanism normally for IDS is when the attack occurs, the sensors will detect unusual pattern in the network traffic that is being monitored by the monitoring system. The signature pattern is analyzed by the system and when the attack pattern is confirmed to be the attack, the system will immediately triggers an alarm for further responsive action by the security unit.

The IDS can be categorized into two: the real-time IDS and the Offline IDS. The real-time IDS analyze the data while the session is in progress. This type of IDS will raise an alarm immediately when the attack is detected. The-offline IDS however, analyzes the data when the information about the sessions are already collected (post analysis).

1.2 Why Data Mining?

The IDS is used to detect 'patterns' of known intrusions and also the 'unusual patterns' of possible intrusions in the analysed or monitored network traffic. However, there is some limitation in the IDS.

The signature based IDS for example, only recognizes the particular attack type that is matched with the collection of signature pattern in the database. The system is incapable to detect new attacks patterns or novel intrusions. Another reason is about the characteristic of the attack itself that had become more sophisticated than before, that is difficult to be detected by the IDS.

The difficulties in detecting the attacks are probably because of several factors as the following:

- **Attack stealth ness**
  The attackers have the ability to hide their actions from monitoring system or IDS, by covering their tracks by editing the system logs or reset a modification date on a file that is being replaced or modified.

- **Novel intrusions**
  Novel intrusion is undetectable by a signature based IDS. The novel intrusion can be detected as anomalies by observing significant deviations from normal network behaviour, which is offered by Data Mining.

- **Distributed or coordinated Attack**
  Requires for attack correlation.

With all the limitations, the IDS system requires additional intelligent methods that can be applied on to the Intrusion Detection process that is Data Mining. Data Mining happens to be the solution towards the problem as it has the ability to perform data analysis and uncover or extract important data patterns (attack).

1.3 Data Mining in IDS Techniques

The steps in the process of Data Mining in IDS are almost similar to the basic process in knowledge discovery in Data Mining. The basic steps in Data Mining techniques for Intrusion Detection are as shown as in the figure below:

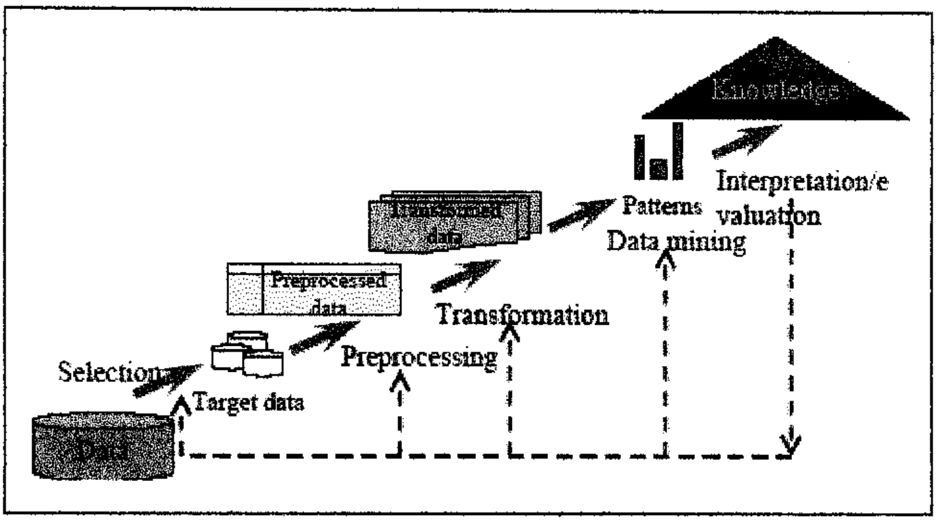


Figure 2: The basic process of Data Mining in IDS

Target Data

The target data is data that is used in the analysis and the monitoring process in the IDS. The data analysed is consist of a list of captured packets gained from the live network or in the IDS training network simulation. These datasets consists of multiple attributes that are known as the packets headers.

Data Pre-processing

The pre-processing process is consists of data cleaning, data integration and data reduction. The pre-processing process is important in the beginning of the process to ensure that the data used in the analysis is ready for mining as the selected data might probably being incomplete, noisy or inconsistent.

Data Transformation

Data also had to be transformed into form of appropriate mining. This includes smoothing, aggregation, generalization, normalization and attributes construction. This is because not all the attributes in the dataset are needed for the analysis. In this phase, the datasets, which is being prepared earlier in the pre-processing process, is transferred into the database.

The Mining

Data mining technique can be implements into both Intrusion Detection Models of the IDS. They are: Data mining for the Misused Detection Model and Data Mining for the Anomaly Detection Model.

Interpretation and Evaluation

The final results from the model generation will be evaluated and interpreted into meaningful information, which is the final product of the knowledge discovery process. The information can also be used for performance measure and analysis.

1.4 Performance Measure

As for this project, the system that is about to be developed will be used at the final stage of the KDD process, that is at the Interpretation and Evaluation stage mentioned before.

To measure the performance of Data Mining technique, firstly, we have to determine the detection rates and the false alarm rates. The detection rate is the ratio between the number of correctly detected attacks and the total number of attacks and the false alarm rates is the ratio between the number of normal connections that are incorrectly misclassified as attacks and the total number of normal connections. Basically the higher the detection rates or the lower the false alarm rates the better the performance.

In the final stage, the final data is in the form of confusion matrix. The confusion matrix is consists of number of false positive and false negative values that can be used to calculate the detection rates and the false alarm rates. These values however, are not predetermined and require us to decide manually using the Receiver Operating Curve (ROC) theory.

In the end, the data is presented into a visual presentation using the ROC. Therefore it would be wonderful to have the system or tool that is able to do all the evaluation process and presents the results automatically which is what this project is all about.

The developed system is a tool or program that be used to simplify the process by simply accepting false positive and false negative values from the user, compute them to get the detection rates and the false alarm rates and generates the ROC as its final result.

1.5 The Developed System

To implement a particular Data Mining (DM) technique for intrusion detection in the Intrusion Detection System (IDS) requires a careful preparation of the datasets that will be the inputs of the learning tool. These datasets or sets of data comprise a collection of captured packets that are collected using a special application known as "sniffer" (e.g. tcpdump program).

The datasets has to undergo the Data Mining or the Knowledge Data Discovery (KDD) process and Attributes Analysis (neglected the DM technique used). There are several stages in the KDD process and each phase requires particular tools to manage the datasets. The application that is commonly used is the DBMS (such as the PostgreSQL) as the datasets can be efficiently managed in the form of databases. The stages in the KDD process are: Data Cleaning, Data Integration, Data Selection, Data Transformation, Data Mining, Pattern Evaluation and Knowledge Presentation.

The knowledge analysis tool such as the *WEKA* program *(see References)*, provides a virtual environment of the IDS and generates the algorithm of the particular DM technique examined. For example, to examine the SVM *(Support Vector Machine)* DM techniques, the WEKA program will generates the Sequential Minimal Optimization *(SMO)* algorithm function to train datasets. The final results of this process can be achieved in the form of confusion matrix that can be later on be used in the performance measure stage during the final stage of the KDD process (Pattern Evaluation).

The main purpose of this project is to develop a working prototype of the performance measuring tool by using the Rapid Application Development (RAD) approach. The Rapid Application Development approach is belong to the Evolutionary Development software process model whereby the initial system is rapidly developed according to the specification defined by the end-user and it is continuously refined from time to time based on the end-user's input.

Basically, there are three main activities involves in the prototype development; specification, development and validation. All these three activities are done concurrently through out the development process. That is from the initial version released until to the final (confirm) version released.

2.0 PROBLEM STATEMENT

The performance measure on the particular DM technique in IDS is according to the detection rates and the false alarm rates of the particular technique that is later on to be interpreted into the ROC graph.

However the performance measurement process is done manually that requires human intervention. That is from the analysis of confusion matrix gained (raw data), followed by the calculation of detection rates and false alarm rates until the generating of the ROC graph (presentable data).

The main problem in implementing the Data Mining in IDS lies in determining the best DM technique to be used and the one that might fit well with the system. In order to determine the best DM techniques, researchers have to do experiments by using the knowledge analysis tools (such as *WEKA*) and datasets as inputs.

With concern of the emerging researchers around the world who are also doing the same research on this matter, a few standard datasets is released for this purpose. Datasets is consists of collection of captured packets that had been collected from live or virtual network which is also includes with many types of intrusions.

The intrusion detection in the IDS detects intrusions by examining these packets and labelled them as attack or normal. In DM, the packets are extracted based on particular packets headers. Depending on the DM technique, not all attributes used in the process.

To determine the best DM technique to be used, is by measuring the performance of the technique in detecting known and unknown intrusions before it is deploy into the real IDS.

With that, a complete research had been done about the development of a tool that capable in measuring performance of the Data Mining (DM) techniques used in the Intrusion Detection System (IDS). The system accepts the final results of the *WEKA* knowledge analysis tool which is in the form of confusion matrix and computes it to generate visual presentation as its final result.

2.1 Problem Identification

According to my experience in the experiments done in my previous research project, there are three main processes involved in the performance evaluation that need to be considered:

- **Confusion Matrix Analysis**

  The confusion matrix is consists of the false positive and false negative values that need to be determined according to the ROC theory. This is the most important part of the process as the results from this stage will be used and a small mistake here might also affect the results of whole process afterwards.

- **Detection Rates and False Alarm Rates Calculations**

  The false positive and false negative values gained are used in the calculation to determine the detection rates and the false alarm rates. This calculation process will ensures the accuracy of final results. If the calculation results gained is not too accurate, one has to start the experiment all over again from the beginning in order to get the best results. If errors occur here, it might take up to some time to find the errors and it will be a waste of time if it has to be reconsidered or to do it from the start.

- **ROC Construction**

  The ROC graph can be constructed using various applications, however the crucial part is during the data entry, whereby the values entered must be precisely copied from the actual results. Therefore mistakes due to typing errors for example, might also affects the final results.

According to the situations above, there are many possibilities for human errors to occur during the evaluation stage and this might slow up the process and exposed to inaccurate performance results.

2.2 Significant of the Project

In short, the 'performance measure' process can be divided into three main phases which will use as the developed system flow: The first phase is to determine the false positive and false negative values that can be derived from the confusion matrix.

The confusion matrix used must be also the final results of the datasets that had undergone a proper DM process which is also known as the Knowledge Data Discovery (KDD). This phase is referred as confusion matrix analysis.

Then we move on to the second phase which calculates the data rate and the false alarm rate from the false positive and negative values gained earlier. The third phase which is also the final stage in the performance measure is the visual presentation.

This process can be made possible by generating a graph known as the ROC or Receiver Operating Curves. The ROC is used later on by the system analyst for decision making in selecting the best Data Mining (DM) technique for their current IDS.

Therefore the developed system is actually handled three different processes: the confusion matrix analysis, the calculation of the detection rates and false alarm rates and the ROC graph generator.

3.0 OBJECTIVE AND SCOPE OF STUDY

Below is the list of objectives of this project:

- To develop a tool for performance measure for Data Mining Techniques in IDS experiments.
- To provide a system that offers simplicity, systematic, accurate and efficient way to measure the performance of the Data Mining Techniques in IDS experiments.
- To provide a better solution to overcome human errors in the performance measuring process by securing the data integrity and automation.
- To improve the efficiency of the experiments done in the related research field by introducing a tool that is able to accommodate such task.

The scope of the study will be narrowed down to:

- Focus on the confusion matrix analysis, the calculation of detection rate and false alarm rate and the ROC graph generator.
- The system will produced accurate results with fewer errors.
- The researchers of the related research field will be the target users.
- Producing the prototype for a system, which accepts the values of the confusion matrix and can be directly, determined the value of False Positive, False Negative, Normal and Attack.
- Research on the algorithm or the method to enable the system to manipulate the inputs data with the least of human intervention and at the same times preserves the user-friendliness in interaction between the system and the end-user.

3.1 The Relevancy of the Project

This tool is developed so that it can be used in the performance measurement process (Performance Measuring Tool for Data Mining Techniques in IDS) so that the final analysis data gained from the KDD process which is in the form of confusion matrix can be handled in a kind of systematic, fast, accurate, proper and efficient way. The developed system will assist researchers in their experiments in the related research project.

CHAPTER 2
LITERATURE REVIEW AND THEORY

1. The Thread

Cyber attacks or cyber intrusions are describes as actions towards an attempt to bypass the security mechanisms of a computer system. There are two parties that can be the potential attackers. They are the outsiders which is refers to the attackers that accessing the system from the Internet and also the authorized users who attempt to gain and misuse non-authorized privileges that violates others privacy in the local network inside the organization. Attacks or intrusions can be briefly described according to several categories as shown below:

- Attack type
- Number of network connections involved
- Source of attack
- Environment

2. Attack Type

Attacks are divided into four main groups of attack type according to its behavior of attack:

- **Denial of Service (DoS) Attacks**
  The DoS way of attack is by shutting down a network, computer, or process or else deny the use of resources or services to the authorized users.

- **Probing and Scanning (Probe) Attacks**

  The probing and scanning attack or Probe for short is when the attacker uses a network services to collect information about the host such as the valid IP addresses, the services its offers or the operating system used.

- **Compromises Attacks (R2L & U2R)**

  The attackers use known vulnerabilities and the weakness in the security to gain access. There are apparently two types of compromise attack the **Remote to Login (R2L)** attack and the **User to Root (U2R)** attack.

  - **The R2L** attack is when the attacker gains access to a machine as an unauthorized user (the attacker don't has an account on that machine) over a network and commits harmful operations.

  - **The U2R** on the other hand is when the attacker him/herself has an authorized or privileges access to the local area network but commit irresponsible acts onto the system.

- **Trojan Horses or Worms**

  This type of attacks keeps on replicating aggressively on other host, whereby the Trojan horses spread by letting itself to be downloaded by user thus, the worm has a self–replicating ability.

3. Number of network connection involved

Attacks can also be classified according to the quantity of network connection involved. Basically there are two types of attacks' involvement, the multiple network connections or burst connections and in the single network connection. The multiple network attacks is when the attacker launch an attacks to more than one machine simultaneously in the live network. The single network attack on the other hand is a one-to-one attack between the attacker and a single host.

4. Source of Attack

In order to counter the attacks, network security unit has to be aware of the potential source of attacks. There are various locations that could be the attacker's homeland, from the single location to several different locations. Not to mention the targeted locations that could be a single destination or many different destinations. To determine the exact location of the distributed attack, the security unit has to analyze network data from several sites.

5. Environment

The attack environment is referring to the type of the network environment or network infrastructure where a particular attack emerges. The network environment could be in the computer networks, in a single host machine, in the peer-to-peer (P2P) environment and in the wireless network. By recognizing the environment for an attack to likely occurs, the security unit might has a clue to overcome the attack and preserved back the security onto the network.

For example, in the P2P environment, the connected computers act as peers on the Internet, there is no such thing as clients or servers. Each of distributed computers in the network doesn't have fixed IP address as they are not relying on the DNS system and therefore to trace the source on a particular attack could be a difficult task.

On the wireless network environment, the physical layer is less secure compared to in the fixed computer networks. In the wireless network, there are no traffic concentration points, where packets can be monitored. More over, the mobile nodes don't have a fixed infrastructure. All of these constraints on the attack environment have to be considered in order to overcome the attacks more efficiently.

6. Firewalls & Antivirus

To overcome the thread, network security has to be established onto the network that provides the security mechanism in the form of hardware and software like Firewalls and Antivirus. The Firewall is used to prevent external attacks from penetrating or gaining access to the network. Acts like a filter that monitors and controls the access points, the Firewalls is the best current solution in network security in encountering external attacks protecting the local network from the harmful intention on the Internet.

Meanwhile Antivirus software provides a secure environment to the computer system, free from viruses and worms. Virus is a self-generated malicious code that capable to take over the system and doing certain damage to the system. With the combination of both worlds, we managed to provide a safer network environment from the outsiders' attacks.

7. Problems with Firewalls and Antivirus

However, protection from the outsiders only, still is not enough. The inside attacks also have to be put under consideration. Firewalls for example, could be an efficient tool to comprehend attacks from the outside, but still gives a room to the insiders attack to occur.

The Firewall doesn't monitor the secure network on the inside, as most of the inside attacks are being committed among the authorized users themselves. Besides that, the outsider's attacks still are able to penetrate into the secure network through the security holes made for programmers, users, and administrators.

The Antivirus on the other hand requires additional updates on its database in order to recognized new viruses and worms. Therefore system admin has to perform the updating process regularly to ensure that the virus definition for the Antivirus software is always up-to-date. Having Firewalls standby and antivirus installed still cannot confirmed to be secure. To improve the security, instead of using Firewalls and Antivirus, the Intrusion Detection System (IDS) is introduced

8. The Traditional IDS Approach

Intrusion Detection System (IDS) is a combination between the software and hardware that is capable in monitoring and analyzing events occurring in the computer system or network. When there is a sign of intrusion, it will raise the alarm, therefore further immediate security actions can be taken to overcome the incoming attacks. The IDS is the right tool to handle the potential attacks from outside and also in the inside the secure network.

The traditional IDS approach is normally based on the signatures of known attacks. However, there are a few limitations regarding the traditional IDS approach that requires a new solution to improve the current IDS. As mentioned before, the traditional IDS approach used a signature database method that has to be revised manually for each new discovered attack, which is in the same case with the Antivirus. More over, it is difficult to deploy new created signatures onto the current IDS. This affects the efficiency of the IDS, as it is unable to detect new emerging cyber threads and has to rely on the administrators.

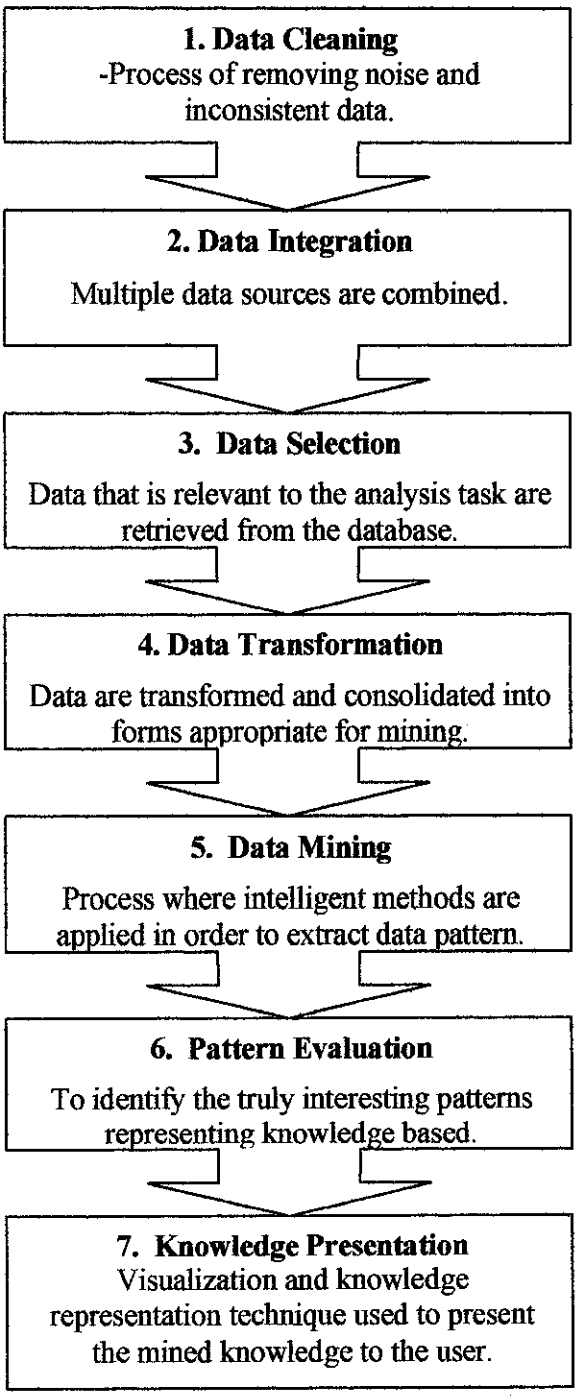
9. Data Mining

Data Mining is used to specify the kind of patterns to be found in the data. With Data Mining, ones can perform analysis and uncover important data pattern to extract the knowledge from a specified source of data. Data Mining mostly used in business for decisions making, forecasting market trends and simplifying the task in marketing strategy.

There are various types of data that can be mined. They are DBMS, Data Warehouses, Heterogeneous Database (DB), Spatial DB, Transactional DB, Multimedia DB, Advance DBS/Application, and much more. In order to classify the Data Mining System, one has to consider the database mixed, knowledge mixed, the kind of technique used, and application that want to be adapted.

Basically, Data Mining is a process of knowledge discovery in databases. The process of knowledge discovery is a process of extracting useful information from large databases.

The steps in the process of the knowledge discovery are as shown in the diagram below:



1. Data Cleaning
-Process of removing noise and inconsistent data.

2. Data Integration
Multiple data sources are combined.

3. Data Selection
Data that is relevant to the analysis task are retrieved from the database.

4. Data Transformation
Data are transformed and consolidated into forms appropriate for mining.

5. Data Mining
Process where intelligent methods are applied in order to extract data pattern.

6. Pattern Evaluation
To identify the truly interesting patterns representing knowledge based.

7. Knowledge Presentation
Visualization and knowledge representation technique used to present the mined knowledge to the user.

There are also lot of different techniques and approaches in order to implement Data Mining. They can be categorized into five different functionalities. But all of this techniques used are depending on the type of the usage or the system that the Data Mining has to be implemented to. The five functionalities are:

Association Analysis

The association analysis searches for interesting relationships among items in a given dataset. Association rule shows attribute value conditions that is frequently occurs together in the given set of data.

Classification & Prediction

Classification is a process of finding a set of models or function that describe or distinguish data classes or concepts. Prediction on the other hand is the process of predicting class of object whose class label is unknown.

Cluster Analysis

Cluster analysis is the process of grouping a set of physical or abstract objects into classes of similar objects. The cluster analysis analyze data object without consulting a known class label. Objects are clustered and grouped based on the principle of maximizing the intra-class similarity and minimizing the inter-class similarity.

Outlier Analysis

The outlier is a data object that does not comply with the general behavior or model of the data analyzed. The outlier analysis analyzes these outliers although most of the techniques in Data Mining discard outliers as noise or exceptions.

Evolution Analysis

The Evolution analysis describes and models regularities or trends for object whose behavior change overtime.

CHAPTER 3
METHODOLOGY / PROJECT WORK

4.0 The System Modules

The performance measurement tool is comprised of three main modules:

- **Confusion Matrix Analysis**
  The final results of DM techniques in IDS are in the form of confusion matrix. In this module the system will determine the False Positive and False Negative values based on the confusion matrix. The results from this module are passed on to the next module for calculation.

- **Detection Rates and False Alarm Rates Calculation**
  This module contains the calculation function which is used to calculate the Detection Rates and the False Alarm Rates. Then it will return the results to the ROC curve generator module.

- **ROC Generator**
  From the results of the previous calculation module, the ROC curve is generated. The ROC curve is the final result of the system which is in a visual and presentable form.

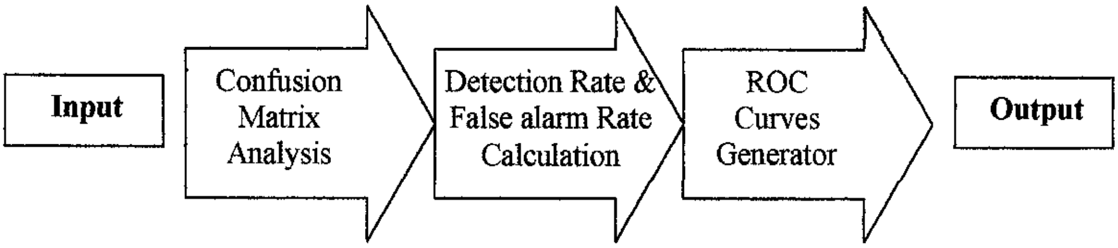

Input
Confusion Matrix Analysis
Detection Rate & False alarm Rate Calculation
ROC Curves Generator
Output

Figure 3: The system flow diagram

5.0 The Evolutionary Development

The evolutionary development approach is based on developing the initial implementation, exposing it to user comments and refining it through many versions until an adequate system can be developed. (Sommerville, Software Engineering, 7th Edition).

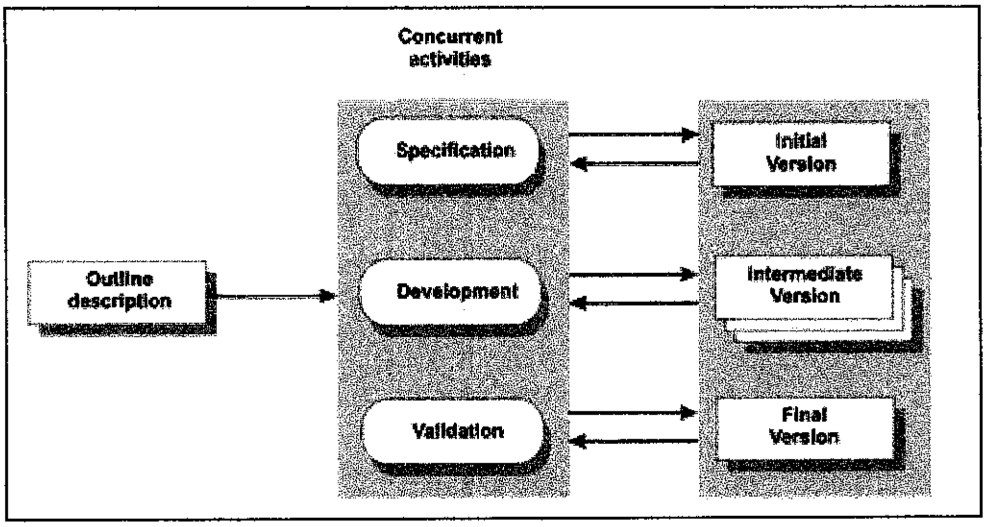


Figure 4: The evolutionary development process.

The development process involves three main concurrent activities; there are specification, development and verification and validation. These activities are interleaved on each other and open to rapid feedback across activities *(see Figure 4)*. There are two types of prototype development approach in the evolutionary development:

1) **Exploratory development** – The development starts with part of the system that is well understood. The prototypes development process requires the developer to work with the end-user from the beginning until the end of the development process. Their requirements contribute to the evolution of the system due to the new features added through out the process.

2) **Throwaway prototyping** – The development is focusing on understanding the user requirements in order to develop the user requirements definition for the system. Therefore the prototype developed is used for experimenting with the user requirements that are poorly understood.

5.1 The Pros and Cons

The evolutionary development is definitely has more advantages than the traditional waterfall approach because of its ability to deliver a final product fast and meet the immediate needs of the end-users.

More over the specification process in the evolutionary development approach is done incrementally, whereby the system's can be easily reconfigured when users develop better understanding of their problem. By doing this, the developed system has a great advantage in meeting the user expectations and satisfactions.

However, the evolutionary approach does have some problems if we look from the engineering and management perspective. There are two problems identified, firstly, the development process is not visible. Therefore it will be difficult for the project manager to keep track on the progress as the system is developed quickly. Besides that it will be not too cost-effective to produce documents for each every version released.

Another problem is that, the system might be poorly structured due to the continual changes. The continuous and rapid changing might corrupt the software structure. And because of this the software changes will become difficult and costly.

In overall, the evolutionary development is still the best approach to development especially for the medium-sized systems such as this project, whereby the system architecture is much simpler and much easier to manage and handle.

6.0 Rapid Application Development (RAD)

As for the system development process, I preferred to use the Rapid Application Development (RAD) approach due to the limited time constraints and tight schedule. In the RAD, I plan to use the evolutionary development prototype which is by developing an experimental system in order to applied definite features and to satisfy user needs. More over the prototype will be improved from time to time and will later on be released as a delivered system when it is ready.

The main idea behind RAD process is to produce useful software quickly. As mentioned earlier, RAD is consists of iteration processes that interleaved each other. There are the specification, design, development, and testing. In RAD, the system is not fully developed and being deploy entirely but from a series of increments. Whereby for each increment, new system's functionality is develop and added onto the system, hence improving the system in overall.

Although there are many approaches to RAD, the following are the common shared fundamental characteristic of a typical RAD:

1. The process of specification, design, and implementation are concurrent. Because of this, there is no detailed system specification, and design documentation. The system is normally developed using a programming environment which minimized the documentation process by generating them automatically.

2. The system is developed in a series of increments. There will be an involvement of the end-user in specifying and evaluating each increment. Through out the development process, the end-user or the stakeholder may propose changes to the system and a new requirement should be implemented onto the system.

3. The graphical user interfaces (GUI) are often developed using an iterative development application. In this project for example, we use the Microsoft Visual Basic 6.0 that using the Object Oriented Development (OOD) approach that allows the user interface designs to be quickly created simply by drawing, dragging and dropping icons.

7.0 RAD Implementation

According to the literature reviews and the previous research, I had decided to implement the Rapid Application Development (RAD) approach for this project. With that, I begin the project work with the planning process.

The planning process is about tasks planning and scheduling. In the Rapid Application Development (RAD) process, the prototype development is done concurrently, whereby the improved version of the prototype is released from time to time during the development process.

As mentioned earlier, there are three main concurrent activities involved in the process; there are the specification, development and verification. Therefore these three major tasks are actually done in different scopes depending on the stages in the prototype development process.

For instance, specification activity in the developing the initial prototype is focusing only on the main functions of the system, otherwise in the intermediate version prototype development stage, the specification activity is more focus on the duplicates functions of the system.

7.1 Project Planning

The planning process requires me to make a detailed schedule for all the activities or the project works that need to be done in the prototype's development process. These activities are neatly scheduled and organize in the form of Gantt chart as follows. There are eight main activities all together in developing the prototype (See Appendixes).

8.0 Incremental Development and Prototyping

Prototyping normally is referred as a process of developing an experimental system that is not intended to be used for deployment by the end-user. Basically a system prototype is developed as guideline to help the software developer and to give a clear picture for the end-user to understand what to be implement.

However in evolutionary prototyping, the term prototyping is actually the real system that is being developed through the incremental development process that is not discarded but evolves according to the user's requirement *(see Figure 5)*. The objective of incremental development prototyping is to develop a working system to the end-user by starting with the best understood user requirement or the one that has the highest priority. The lower-priority requirements are implemented on the user's demand and are normally are consists of duplicate functions.

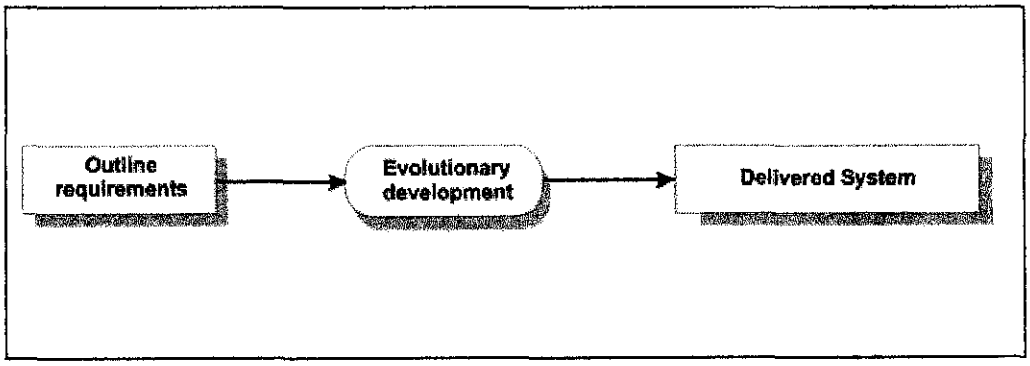


Figure 5: Evolutionary development prototyping.

There are two main advantages in applying the Rapid Application Development (RAD) as describes in the following:

1. Accelerated delivery of customer services – The customer or the end-user will get the value from the system early as the system can be delivered during the early increment in the development process. The early increment emphasis the high-priority functionality of the system and therefore the customer or the end-use be able to see the initial version of the product and specify changes that can be incorporated for the next released.

2. User engagement with the system – As in RAD development process requires user's intervention in order to provide feedback and comments; therefore it is good thing to have them around. This is because, besides be able to produce a system that suites their requirements, end-user involvement might also reflects their commitments in realization of the developed system.

9.0 The Process Design

Use case diagram and data flow diagram is used to model the process. Use case diagram depicts the functions provided by the system to external entity (actor) or in the other words, use case diagram concerns about the specification the function of the system that we are building. In this case, it is between the Performance Measuring Tool and the research officer. (Please refer to Appendix)

The Data flow diagram (DFD) depicts the flow of data through a system and the work of processing performed by that system. The conceptual level is used to show the flow of data in the Performance Measuring Tool. The system however is consists of three main process or functions that handles different tasks. Therefore the flow diagram for each component is also varies. (See Appendix)

10.0 The Interface Design

The design of the interface touches on the system functionality.

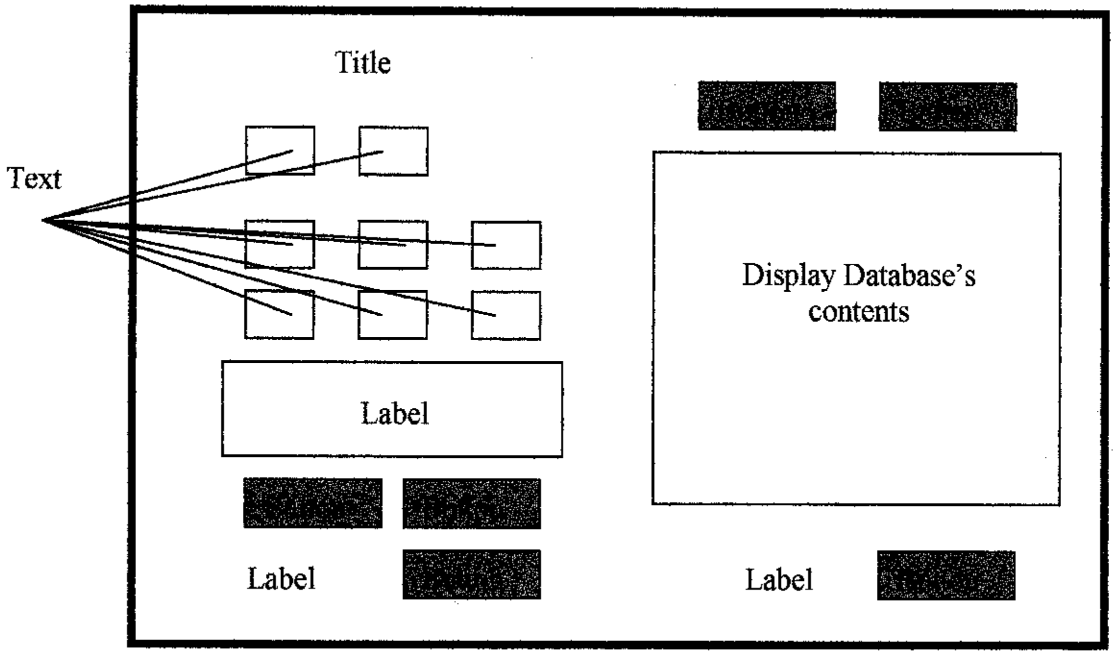


Figure 6: Confusion Matrix Analysis Interface

The initial form or the main interface of the Performance Measuring Tool is the Confusion Matrix Analysis Interface *(see Figure 6)*. The interface is consists of the confusion matrix analyzer that directly accepts the confusion matrix values. During the inputs insertion process, the system will run the inputs validation checking for each and every inputs entered by the user.

The inputs validation checking is a procedure that will ensure only the correct inputs are entered *(see Figure 7)*. By doing so, the system at the same time be able to guide user through out the analysis process. Whenever there is an invalids input during the key-in process, an error messages will appears immediately to warn user from proceeding and suggest them for entering the correct inputs.

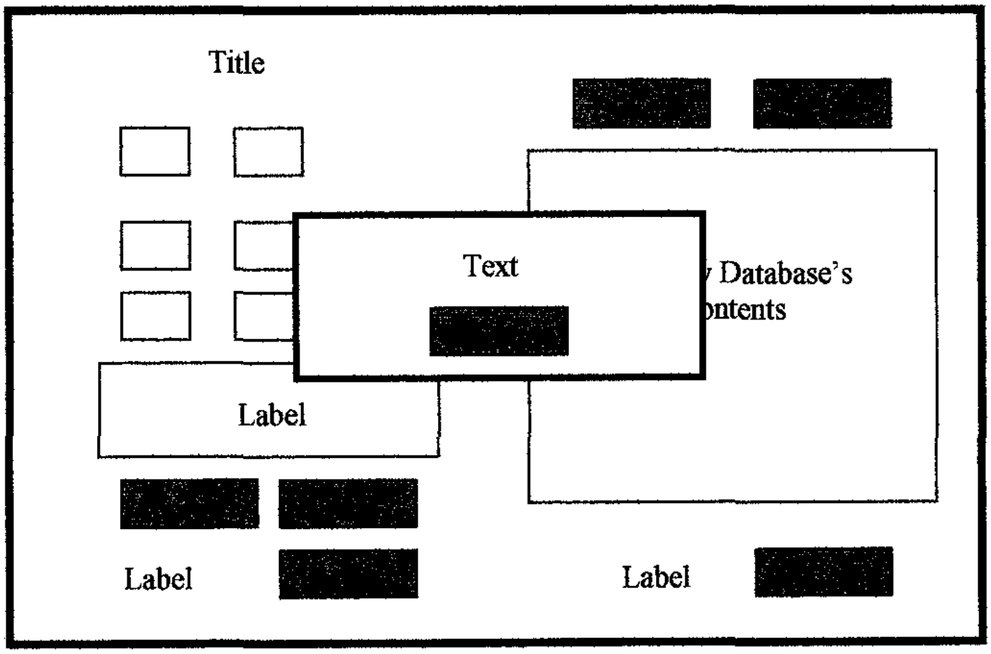


Figure 7: Input Validation Checking

Then the user will be given a freedom to view the results of the analysis process and stores them in the database. User can also modify the contents of the database and do some editing by using the Content Editor interface before moves on to the calculation process.

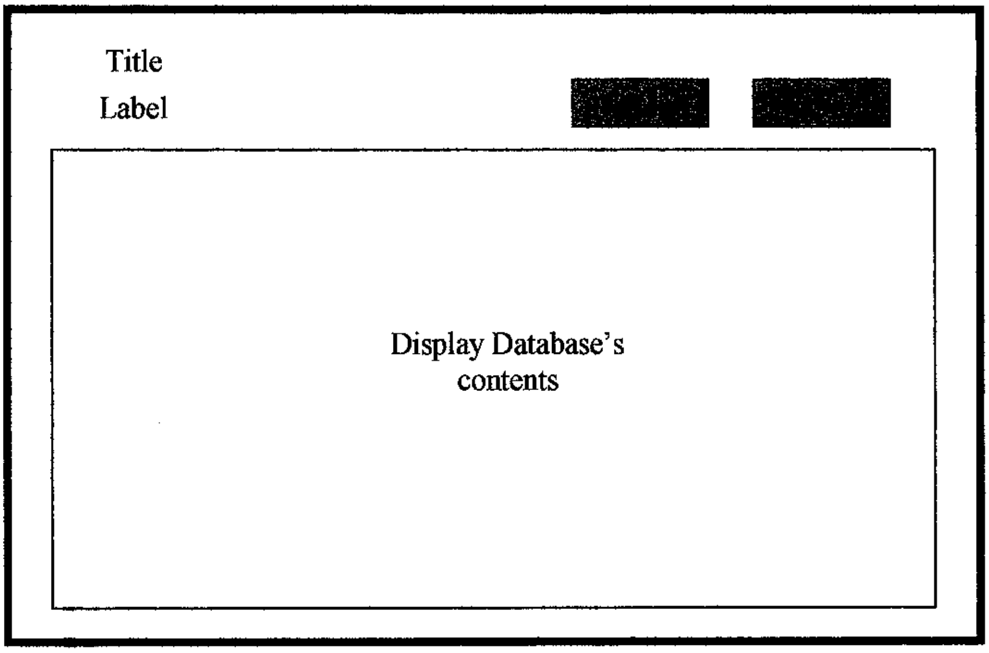


Figure 8: Calculation's Results Interface

The Content Editor interface will guides user in modifying the database contents. The use be able to seek for a specific row of data and do some changes onto the specified data without having to go through all the record entries in the database manually. The system provides user with database editting features such as add and remove record entry, save and update records, and the navigation control to browse records. *(See Figure 9)*

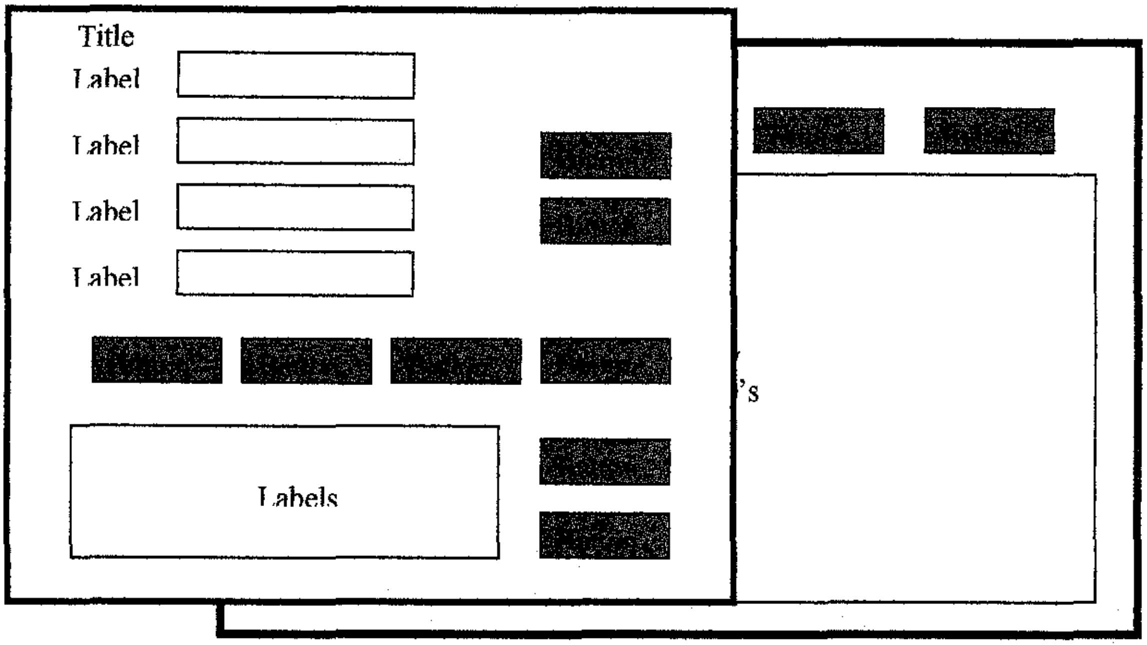


Figure 9: Content Editor Interface

The final process in the Performance Measuring Tools is to generate the ROC graph. The system will generate the ROC graph based on the Detection Rate and the False Alarm Rates data in the database. With a single press of a button, the graph will be automatically generated as a visual presentation of the data for the user to analyze *(See Figure 10)*.

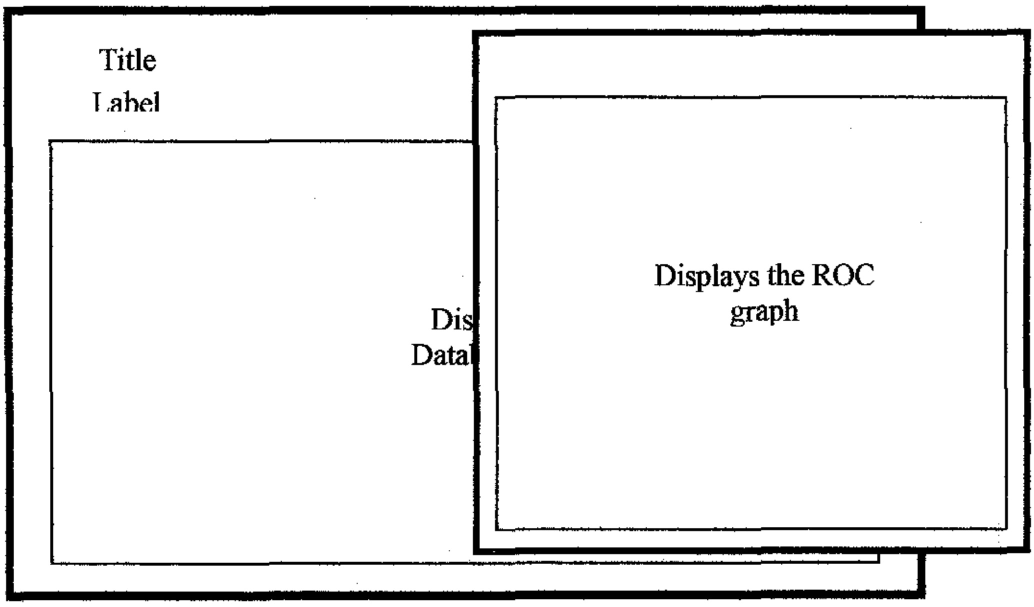


Figure 10: The ROC Graph Generator

11.0 Verification and Validation

Verification and validation of this system are a tantamount to the testing step in building a conventional information system. Verification intended to ensure that the system is right through a procedure, while validation involves testing the system to ensure it is the right system or meeting expectation.

Iterative testing and verification are done throughout the development of the system. This is so as to test the prototype functioning as in requirements by running different data according to different situation given.

12.0 Testing Procedure

The testing process is done each time a new function is implemented onto the system. Every time a new modification on the prototype system is made, a testing procedure is done.

The input and output results of a program usually fall into a number of different classes that have common characteristics such as positive numbers, negative numbers, 0s or 1s only, and menu selections.

Programs normally behave in a comparable way for all members of a class. Because of this equivalent behavior, these classes are sometimes called equivalence partitions or domains. One systematic approach to test case design is based on identifying all partitions for a systems or component.

12.1 Partition Testing

Partition testing can be used to design test cases for both systems and components. As shown in Figure 3, each equivalence partition is shown as an ellipse. Input equivalence partitions are sets of data where all of the set members should be processed in an equivalent way.

Output equivalence partitions are program outputs that have common characteristics, so they can be considered as a distinct class. You also identify partitions where the inputs are outside of other partitions that you have chosen.

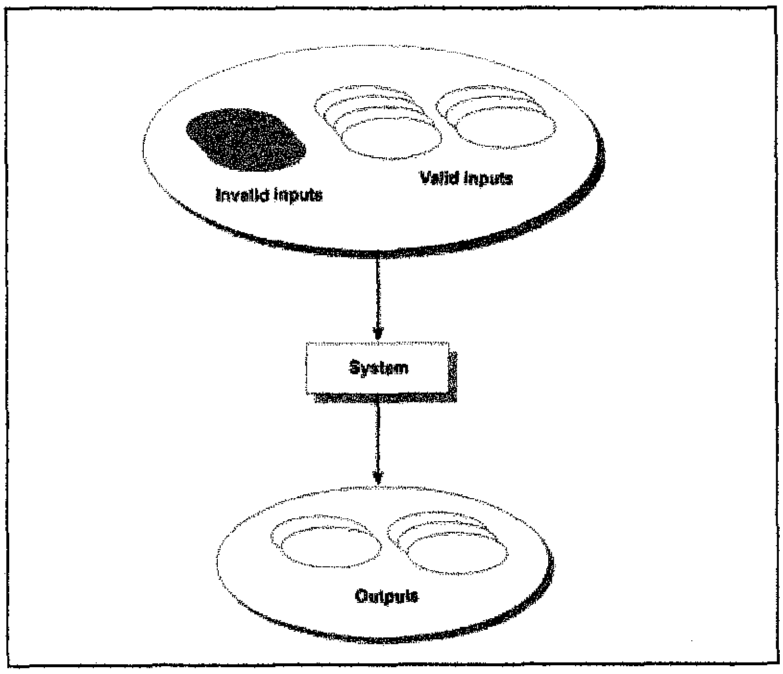


Figure 11: Equivalence partitioning.

These test whether the program handles invalid inputs correctly. Valid and invalid inputs also form equivalence partitions. The rationale of this is that the designers and the programmers tend to consider typical values of inputs when developing a system (Software Engineering, Sommerville).

13.0 Testing Process

The testing process is done each time a new function is implemented onto the system. Every time a new modification on the prototype system is made, a testing procedure is done.

The testing process main goals are to demonstrate to the developer and the customer that the developed software meets its requirements and to discover faults or defects in the software where the behavior of the software is incorrect.

There are a few types of test done onto the prototype system each time the system is altered. These tests are different from each other in terms of its purpose and objectives. The tests done are describes as in the following:

No.	Test	Description
1.	System Testing	Involves integrating two or more components that implement system functions or features and then testing this integrated system.
2.	Integration Testing	Is a part of the System Testing procedure where the developer has to access to the source code of the system and when the problem discovered, the involved component will be debugged.
3.	Release Testing	Is also a part of the System Testing procedure where a version of the system could be released to users. Basically in the system's development process there are three times where the prototype version is released; the initial version, the intermediate version and the final version.

4.	Performance Testing	Is a test designed to ensure that the system can process its intended load. The testing is concerns on the both, that are demonstrating that the system meets its requirements and discovering problems and defects in the system.
5.	Component / Unit Testing	Is the process of testing individual components in the system. In this case for the prototype system, three are three major components involved that are the Confusion Matrix Analysis Function, the Calculation of the Detection Rate and the False Alarm Rate Function and the ROC Generator Function.

14.0 Experimental Data

The data used for the system's inputs and referred outputs in the testing procedure is based on my previous experiments done during my eight months internship at MIMOS Berhad. During the internship, I had done experiments on one of the Data Mining technique used in IDS that is the Support Vector Machine (SVM) to be implemented in the intrusion detection of the IDS.

There are two types of experiments, the first experiment is the SMO classification on the normal/attack data and the second experiment is the SMO classification on the normal/mixed attack data. The objective of these experiments is to measure the effectiveness of the SVM techniques in detecting and classifying attack or intrusion in the normal/attack data and also in classifying five different types of attacks in the normal/mixed attack data.

In the experiments, I had used two different groups of datasets which are prepared according to the KDD process. There are two separated experiments that need to be conducted. Therefore it requires two groups of training data that has to be prepared:

- **Group A** -For Experiment 1 that is to classify normal data and various kinds of attack data.
- **Group B** -For Experiment 2 that is to classify five types of attack and normal attack: portsweep, ipsweep, smurf, neptune, and teardrop.

Group A Dataset

For group A, the datasets is consists of normal data and various types of attack data, which are randomly selected and transferred into separate tables. The amount of data for each table is carefully specified according to the percentage range from 10% to 90% of normal data and attack data. There are nine tables for each week (week1 to week 7) created for the 5000 data (*See Table 1*).

Set	Normal (%)	Attack (%)
1	10	90
2	20	80
3	30	70
4	40	60
5	50	50
6	60	40
7	70	30
8	80	20

Table 1: Sets containing different percentages of data.

Group B Dataset

For group B, the datasets is consists of normal data and combine with only five types of attacks (portsweep, ipsweep, smurf, neptune, and teardrop), which are randomly selected and transferred into separate tables. The amount of data for each table is carefully specified according to the percentage range from 10% to 90% of normal data and attack data. There are only nine tables created for the 5000 data from the combination of week 1 until week 7.

15.0 Testing Data

The dataset used for the testing procedure is based on Group A dataset which is the SMO evaluation on the 5000 random data on week 7, table of 40% normal data and 60% attack (Take note that the SMO is stands for Sequential Minimal Optimization and is the algorithm used in SVM to train data). The result of week 7 is shown in the Table 2 below:

Table	Normal	Attack	Detection Rate (%)	False Alarm (%)
week7_10n90a	500	4500	**99.93**	**0.20**
week7_20n80a	1000	4000	**99.98**	**1.00**
week7_30n70a	1500	3500	**99.97**	**0.00**
week7_40n60a	2000	3000	**99.97**	**0.50**
week7_50n50aa	2500	2500	**99.92**	**0.10**
week7_50n50ab	2500	2500	**99.96**	**0.10**
week7_60n40a	3000	2000	**99.95**	**0.33**
week7_70n30a	3500	1500	**100.00**	**0.23**
week7_80n20a	4000	1000	**99.90**	**0.18**
week7_90n10a	4500	500	**100.00**	**0.13**

Table 2: Experimental Results with Different Parameters (A) Detection Rate (B) False Alarm Rate for SVM (5000 Data) – Table of Week 7

The ROC Graph of Week 7 (5000 data):

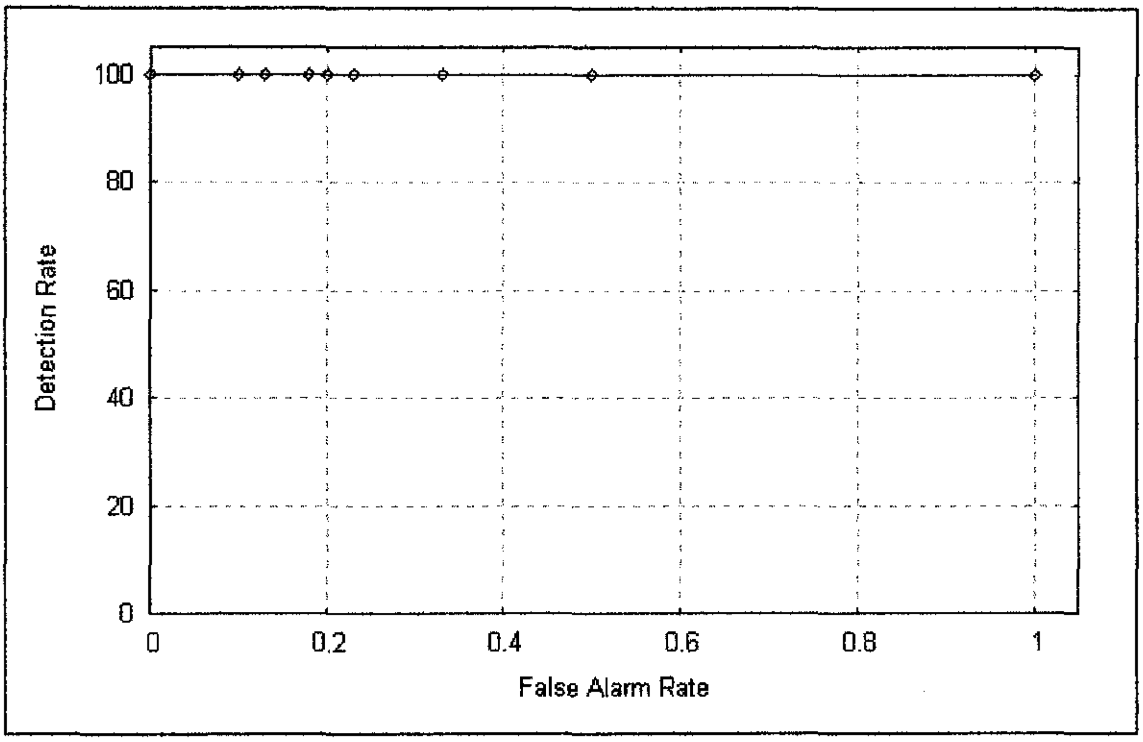


Figure 12: Experimental Results with Different Parameters (A) Detection Rate (B) False Alarm Rate for SVM (5000 Data) – ROC graph for Week 7

Description on the Detection Rate and False Alarm Rate Calculation for Week 7:

The calculation on the Detection Rate and the False Alarm Rate is done according to the confusion matrix gained from the SMO results. The SMO evaluates 5000 data which is randomly fetched from the DARPA 1998 IDS Evaluation datasets. Each table of 5000 random data is consists of fixed amount of random attack data and normal data, which is determined according to the percentage.

There are ten tables all together, consists of 10%, 20%, 30%, 40%, 50% (Table A), 50% (Table B), 60%, 70%, 80%, and 90% of random attack data and the rest of the 5000 data is the normal data *(see Appendixes).*

16.0 HARDWARE & TOOLS

1. Project Management & Modeling

As for managing the project and creating system's documentation for modeling purpose in the development process, I used Microsoft Visio.

2. Analysis

Gnuplot 4.0 is used for statistical analysis and presenting results of studies.

3. Development

Microsoft Visual basic 6.0 is chosen to be the main programming language for the system's front end. Microsoft Access 2000 will be the back end for the prototype's repository.

4. Documentation

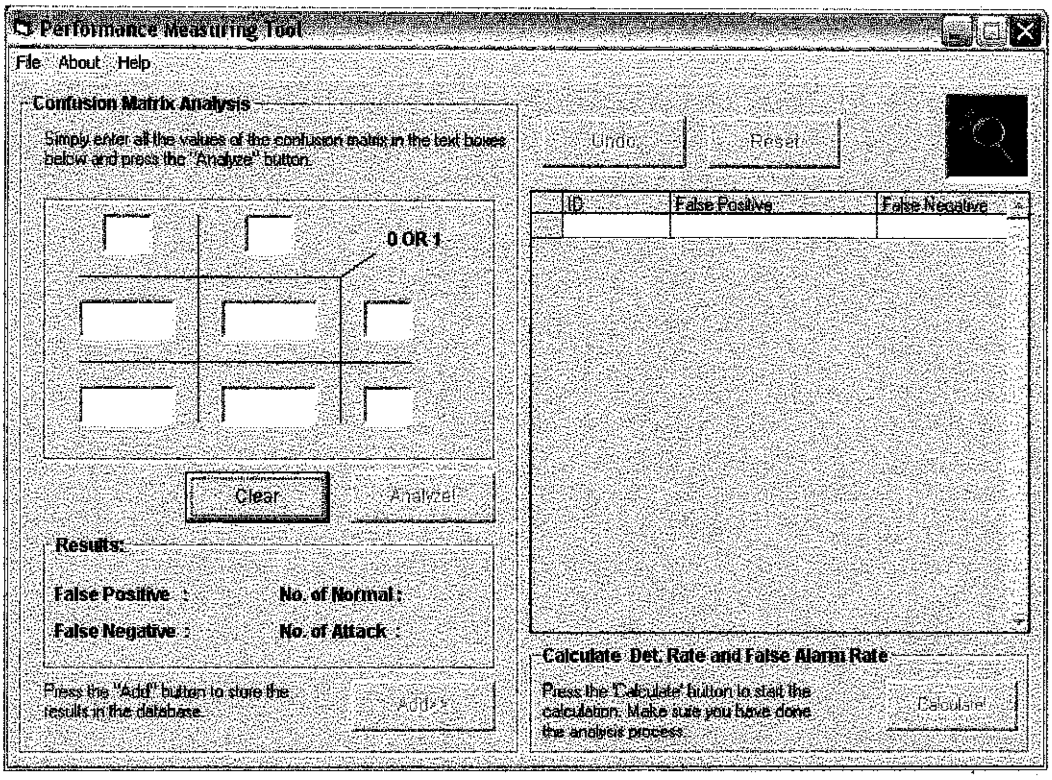
Microsoft Word is used for composing most of the project documentation.

Below are the hardware specification and tools needed for development of the developed system:

Hardware Specification	Software Used
- Pentium III 600 MHz - 256 MB - 10 GB Hard Disk Drive - CD-ROM drive - Floppy disk drive - Mouse - Keyboard - Monitor - VGA driver 8MB	- Visual Basic .Net / 6.0 - Gnuplot 4.0 - Microsoft Access 2000

CHAPTER 4
RESULTS AND DISCUSSIONS

17.0 The Prototype System



Simply enter all the values of the confusion matrix in the text boxes below and press the "Analyze" button.

Press the "Add" button to store the results in the database.

Press the 'Calculate' button to start the calculation. Make sure you have done the analysis process.

Figure 13: Main Interface

The main interface of the prototype system is the Confusion Matrix Analysis interface *(See Figure 13).* At the system's start up, the main buttons are inactive or disabled. These buttons are activated after user has complete filing up all the required data in the textboxes which is the integer value of the confusion matrix.

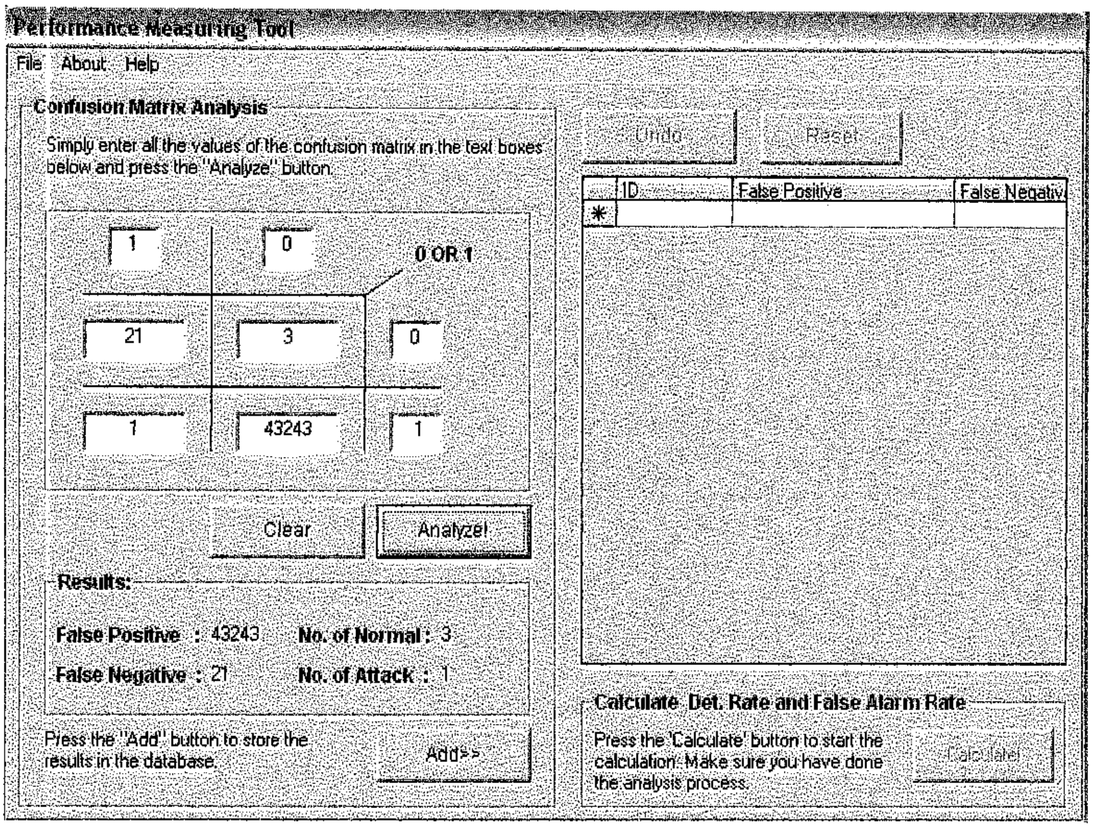
18.0 The Confusion Matrix Analysis

The confusion matrix analysis is basically a process of determining the False Positive (FP), True Positive (TP), False Negative (FN) and True Negative (TN). Depending on the field of studies or usage, the values of FP, FN, TP, and TN are varies. For example, as for this system, the confusion matrix analysis accepts the value of confusion matrix which is derived from the data mining technique.

Therefore, according to the ROC theory, the False Positive (FP) value will be the normal packets which are recognized or labeled as attack; the False Negative (FN) value in the other hand is actually a group of attack that is labeled as normal. The True Positive (TP) value is the attack packets that is correctly labeled as attack and same goes to the True Negative (TN) value which is consists of normal packet labeled as normal *(See Table1).*

The result of the confusion matrix analysis will be used later on in the calculation of the detection rate and the false alarm rate. Therefore the accuracy and precision of results in the analysis process is very crucial as the calculation process is fully depending on the analysis results.

The confusion matrix gained from the WEKA Knowledge Analysis Tool can be used to determine the Detection Rate and the False Alarm Rate. What user has to do now is just directly enter all the values of the confusion matrix and the system will determine the False Positive, False Negative, Normal and Attack value, when user press the Analyze button. Then the analyzed results will be displayed in the 'Results' column *(See Figure 14).*



Simply enter all the values of the confusion matrix in the text boxes below and press the "Analyze" button.

1	0	
21	3	0
1	43243	1

False Positive : 43243 No. of Normal : 3
False Negative : 21 No. of Attack : 1

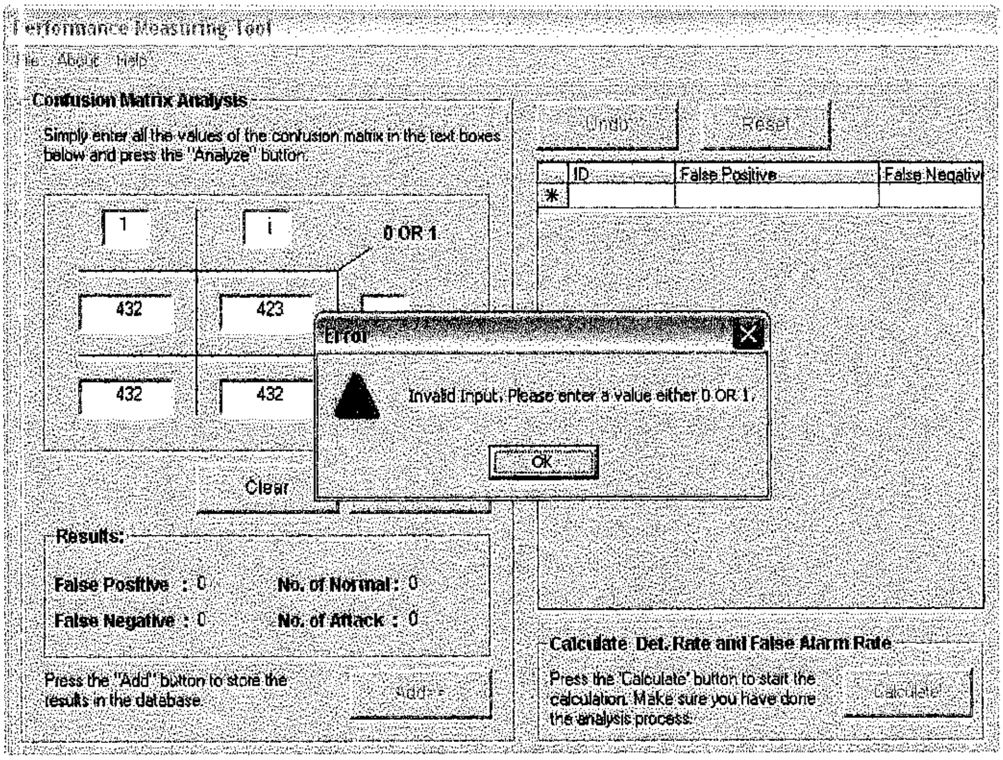
Press the "Add" button to store the results in the database.

Press the 'Calculate' button to start the calculation. Make sure you have done the analysis process.

Figure 14: Confusion Matrix Analysis

However during the inputs process, the system will also perform the inputs validation checking procedure to ensure that the data entered is correct, before the analysis could be made.

When the system encounters errors during the user intervention, error messages appears depending on the invalid conditions detected. User has to reenter the specified data in order to proceed with the task *(See Figure 15)*.



Simply enter all the values of the confusion matrix in the text boxes below and press the "Analyze" button.

1	i
432	423
432	432

Invalid Input. Please enter a value either 0 OR 1.

False Positive : 0 No. of Normal : 0
False Negative : 0 No. of Attack : 0

Press the "Add" button to store the results in the database.

Press the 'Calculate' button to start the calculation. Make sure you have done the analysis process.

Figure 15: Input Validation Checking

Once the analysis has been done, the Add button is activated which allows user to store the analysis result into the system's database.

18.1 Input Validation Checking

According to the system flow diagram, user will enter directly all the confusion matrix values and the system will automatically validate all the inputs entered before move on to the analysis process. The validation process is being implemented by settings some constraints. These constraints are the limits or the invalid conditions for the system to filter during the user intervention.

The system's validation process is called the Input Validation Checking process. These inputs validation checking is based on a few conditions as follows:

- **Empty textbox or Null entry** - The system will response when user is not completely filled in all the inputs in the required textboxes.

- **Reset condition** – The system will check whether it is a reset condition or not before activates the analyze button.

- **Inputs overflow** – The systems will inputs within an acceptable range based on certain threshold defined.

- **Inputs Data Types** – The system will only accept inputs with the appropriate data type that is the integer numbers.

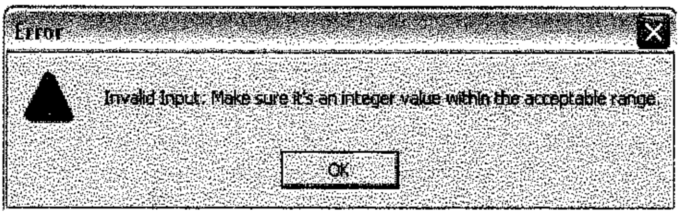
- **Only 1s and 0s condition** – The system will allowed only 0s or 1s integer numbers with the correct combination of 0s and 1s entered.

18.2 Types of Error Messages

Each time the system detects an invalid entry from the user, it will display a particular error message to notify user the current error that occurs and waits for user's respond.

Below are the descriptions about types of error messages and the reasons for it to occur:

1) Error Message 1

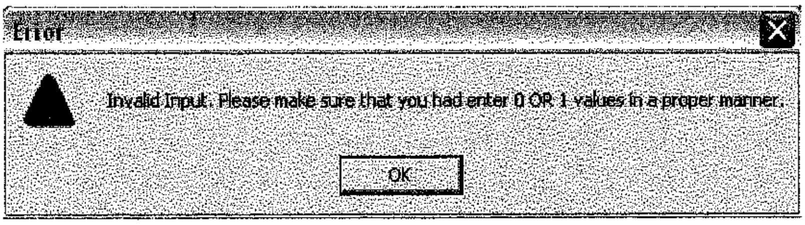


Invalid Input. Make sure it's an integer value within the acceptable range.

Description:

This error message will appear when user enters a value other than integer numbers or a null entry. The integer number entered must also within the acceptable range that is not more than 1,000,000.

2) Error Message 2



Invalid Input. Please make sure that you had enter 0 OR 1 values in a proper manner.

Description:

When this error message occurs, it means that there is an incorrect input regarding the 0s and 1s values detected. The 0 value should be enter side by side with the 1 value or vice versa.

There are only 4 valid combinations of 0s and 1s values out of 16 combinations ($2^4 = 16$ possibilities). The four valid combinations are shown as in the figure below:

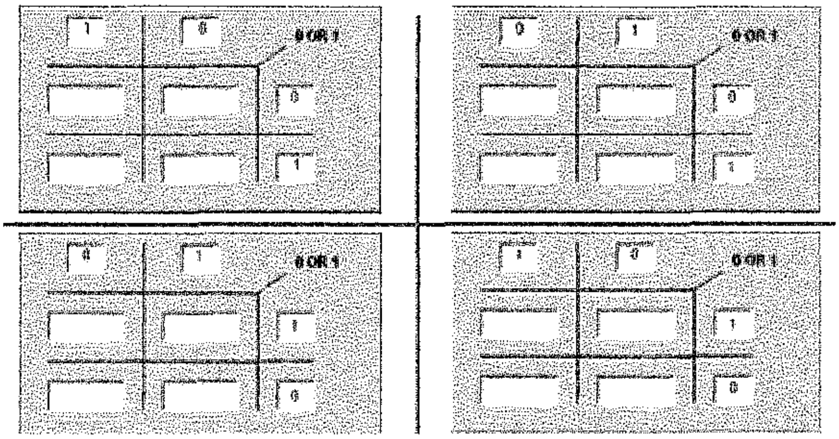
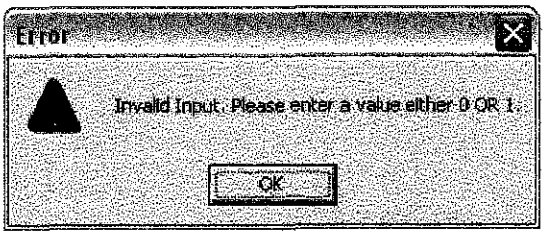


Figure 16: The four valid combinations of 0s and 1s.

3) Error Message 3



Invalid Input. Please enter a value either 0 OR 1.

Description:

This error message will appear when user enters other values or integer numbers rather than 0 or 1 value.

19.0 The Detection Rate & False Alarm Rate Calculator

The calculation of the Detection Rate and the False Alarm Rate is based on the Receiver Operating Characteristics formula or the ROC

19.1 The Performance Measure

The Receiver Operating Characteristics (ROC)

The ROC is stands for Receiver Operating Characteristics (ROC) curve that is used in the analysis to measure performance of method used in the IDS. To develop the ROC curves, one has to consider the True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). In my understanding, all of these fractions can be assumed as in the following table:

	Attack	**Normal**
Positive **- Assumed / labeled as attacks**	**True Positive** Assumed / labeled as attacks and are attacks	**False Positive** Assumed / labeled as attack but normal
Negative **-Assumed / labeled as normal**	**False Negative** Assumed / labeled as normal but are attacks	**True Negative** Assumed / labeled as normal and are normal

Table 3: Relation of True Positive, False Positive, False Negative and True Negative.

Whereby,

$$\boxed{\textbf{TP (True Positive)} + \textbf{FP (False Positive)} = \textbf{1}}$$

The performance of the IDS is measured by two indicators the detection rate and the false alarm rate. The detection rate is the ratio between the number of correctly detected attacks and the total number of attacks.

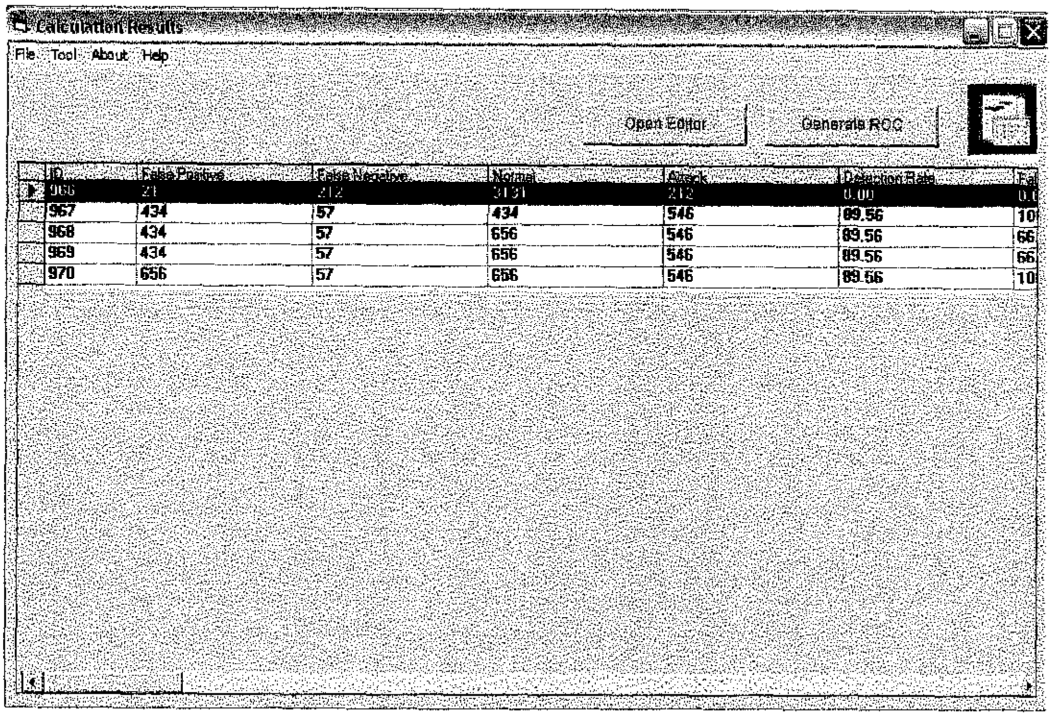
According to the theory, the formula should be:

$$\boxed{\textbf{Detection Rate} = \textbf{1} - \frac{\textbf{Number of False Negatives}}{\textbf{Total number of Attack Connections}}}$$

The false alarm rate in the other hand is the ratio between the number of normal connections that are incorrectly misclassified as attacks and the total number of normal connections. The formula will be:

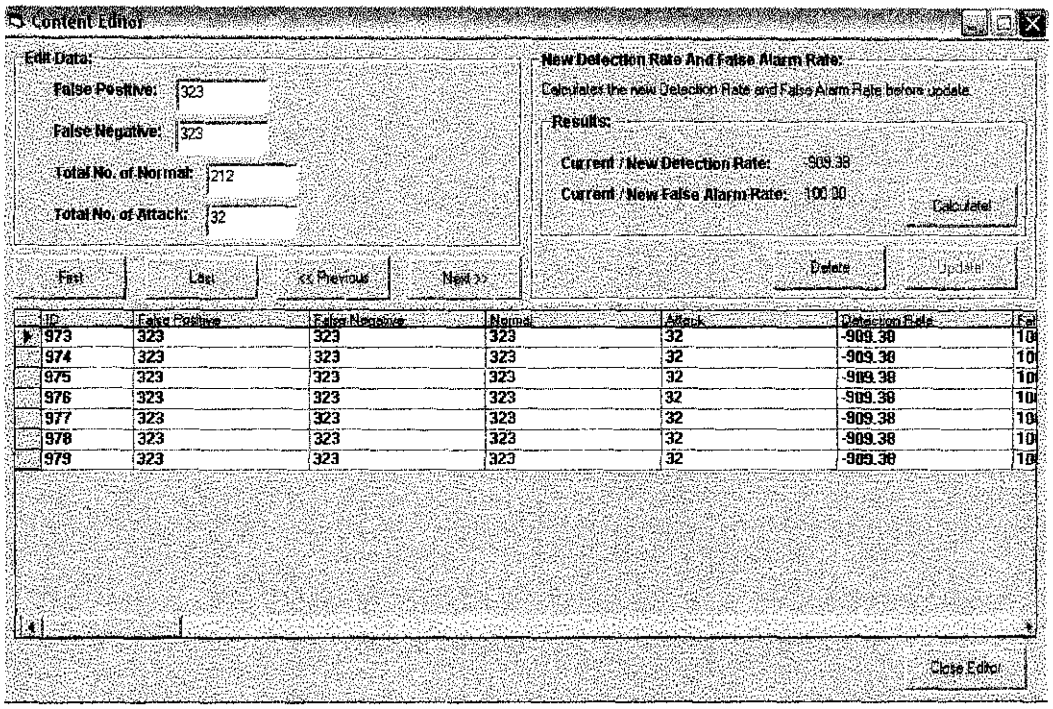
$$\boxed{\textbf{False Alarm Rate} = \frac{\textbf{Number of False Positives}}{\textbf{Total Number of Normal Connections}}}$$

In the prototype system, the result from the analysis process which is stored in the database is used to determine the Detection Rate and the False Alarm Rate. The results are displayed in the Calculation's Results interface. Here, user is able to edit the data by using the Content Editor Interface. To open the editor, user has to press the Open Editor Button on the top left of interface *(See Figure 17).*



	ID	False Positive	False Negative	Normal	Attack	Detection Rate	Fal
▶	966	21	212	3131	212	0.00	0.0
	967	434	57	434	546	89.56	10
	968	434	57	656	546	89.56	66
	969	434	57	656	546	89.56	66
	970	656	57	656	546	89.56	10

Figure 17: Calculation Results Interface



Edit Data:

False Positive: 323

False Negative: 323

Total No. of Normal: 212

Total No. of Attack: 32

New Detection Rate And False Alarm Rate:

Calculates the new Detection Rate and False Alarm Rate before update.

Results:

Current / New Detection Rate: -909.38

Current / New False Alarm Rate: 100.00

	ID	False Positive	False Negative	Normal	Attack	Detection Rate	Fal
▶	973	323	323	323	32	-909.38	10
	974	323	323	323	32	-909.38	10
	975	323	323	323	32	-909.38	10
	976	323	323	323	32	-909.38	10
	977	323	323	323	32	-909.38	10
	978	323	323	323	32	-909.38	10
	979	323	323	323	32	-909.38	10

Figure 18: Content Editor Interface

20.0 The ROC Graph Generator

The ROC generator is used to retrieve the value of the confirmed Detection Rate and False alarm Rate in the system's database and used it to generate the ROC graph. User is able to generate the ROC graph by pressing on 'Generate ROC' button on the Calculation's Results interface.

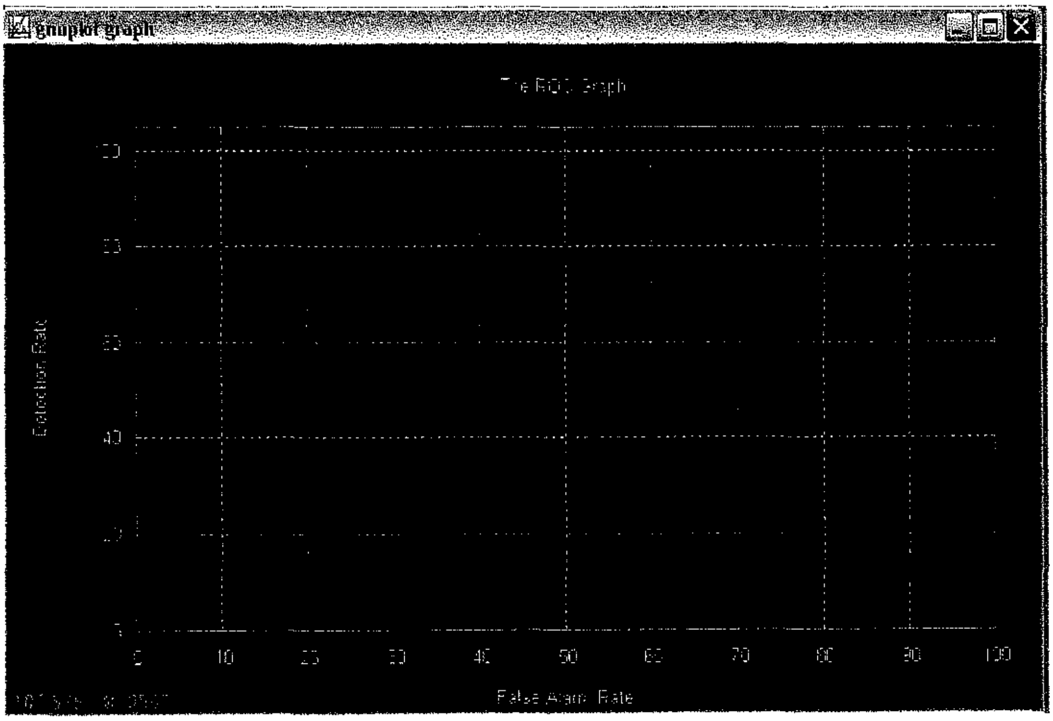


Figure 19: The Results - ROC Graph

20.1 Writing the ROC Graph Generator Script

According to the plan in the beginning, I had planned to generate the ROC graph by using scripts from the Database Toolbox Function in MATLAB. However the scripts can only be generated by using the MATLAB software and therefore in order for the system to generate the graph, it requires the installation of the MATLAB software. In this case, the system will no longer become a standalone system as it has to rely on other software in order to perform certain task.

With this, I had decided to write the ROC graph generator script using the Gnuplot 4.0. The Gnuplot software is much stable and light as it requires only a small portion of hard disk space. More over the Gnuplot is open source software that can be legally copied and used without having to confront with any copyright issues as stated in the General GNU Public License.

However the Gnuplot software only reads the text file format (*.txt) and the comma-separated value format (*.csv) file format. By using the Gnuplot I don't have to worry about establishing connection between the system database and the software as the software supports the used of text files and the csv files. The idea is to convert back all the data required into the csv file format to be fetched by Gnuplot to generate the graph.

The only problem here is the system's database itself is in the form of Microsoft Access Database file that could not be associated with Gnuplot. Therefore I had to create another function that capable to fetch the particular data in the system's database and transfer them into the merged csv file format.

To make it up-to-date, the function is made active each time the user click on the 'Generate ROC' button. Therefore, the problem is solved. Below is the flow chart of generating the ROC graph that might help in explaining the process:

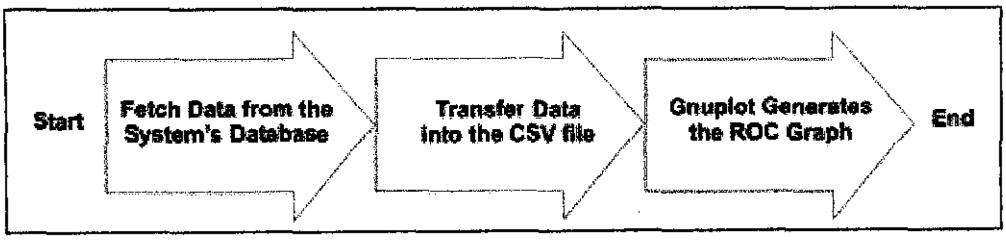

Start
Fetch Data from the System's Database
Transfer Data into the CSV file
Gnuplot Generates the ROC Graph
End

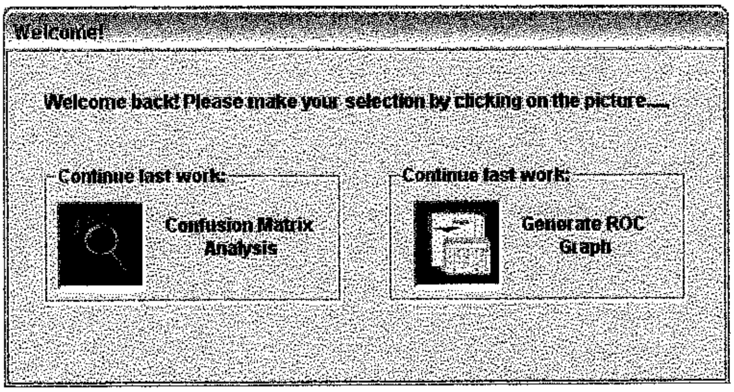
Figure 20: The ROC Graph Generator Process.

21.0 The Welcome Screen Interface Wizard

Although the system has fulfilled most of the main user requirements, the current system still has some limitation. The system don't have the save function which can capture records from the system's database and saved it into a new Microsoft Access Database File (*.mdb). This is because; the development of such function requires a complex programming and therefore requires more time.

According to the schedule, the time estimated for development process of the prototype system is about two months only. Therefore the development of such function is impossible. However I have created a new way of saving the user's work. What user has to do is just left the current work and quit the program.

The system in the other hand will capture the last record or work that the user had done. To continue last work, what user has to do is simply restart the program and the user will be greeted with a welcome screen that will guide user to continue their work *(See Figure 21).*



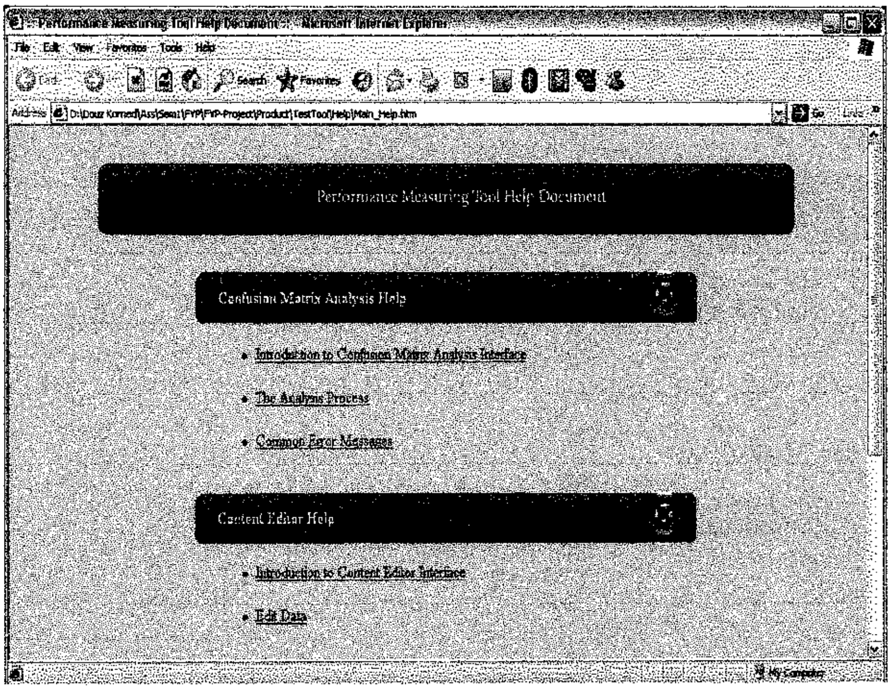
Welcome back! Please make your selection by clicking on the picture....

Figure 21: The Welcome Screen Interface Wizard

This function is called the wizard function, whereby the function will detect whether there is a remaining data in the system's database or not. If there is one or more records detected in the system's database, the system will automatically guide user to the welcome screen wizard and waits for user respond or else it will directs user to the system's main interface.

22.0 Help Document / User Manual

In order to improve the system in term of the ease of interactivity or the user-friendliness between the system and the user, I had develop a complete yet brief help documents in the form of HTML format which user can easily reached from the main menu in the system.


:: Performance Measuring Tool Help Document :: - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\Douz Komed\Ass\Sem1\FYP\FYP-Project\Product\TestTool\Help\Main_Help.htm
Go Links
My Computer

Performance Measuring Tool Help Document

Confusion Matrix Analysis Help

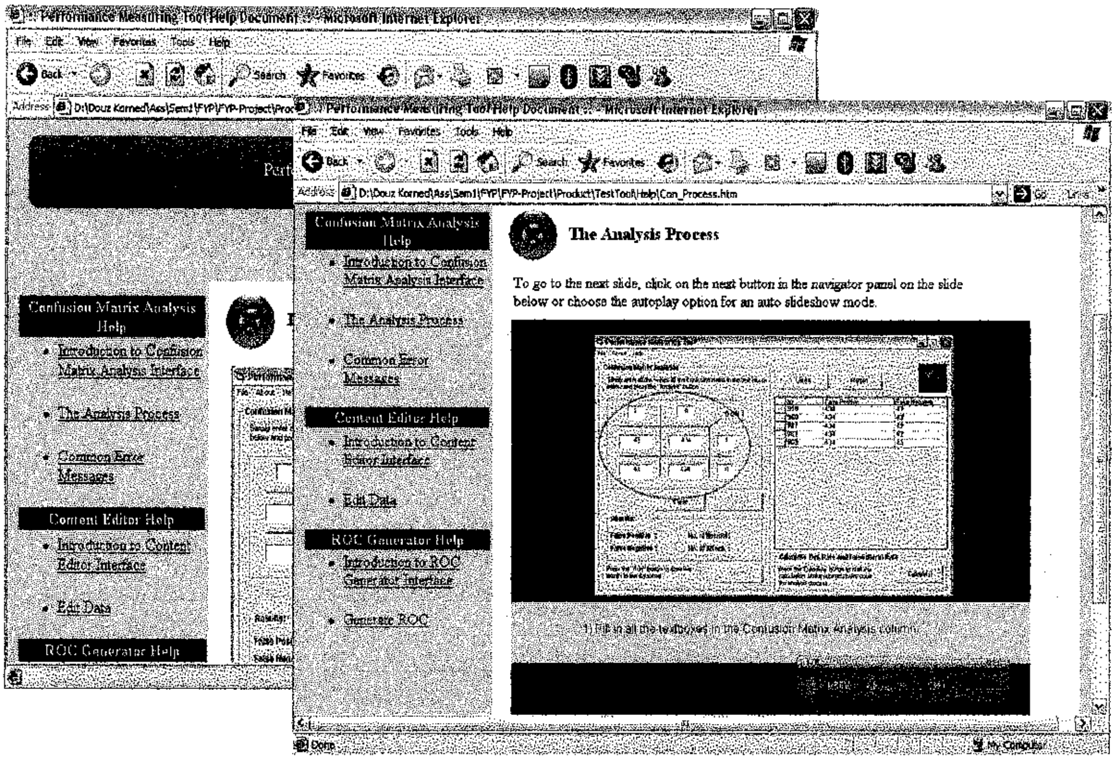
- Introduction to Confusion Matrix Analysis Interface
- The Analysis Process
- Common Error Messages

Content Editor Help

- Introduction to Content Editor Interface
- Edit Data

Figure 22: Help Document front page

The help documentation is consists of a step by step tutorial slides instead of lengthy texts, covering from the Confusion Analysis process until the creating of the ROC graph. User is able to view a complete process and learn fast from the visual presentation of the tutorial slides *(See Figure 22 & 23).*


Performance Measuring Tool Help Document :: - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address D:\Douz Korned\Ass\Sem1\FYP\FYP-Project\Product\TestTool\Help\Con_Process.htm
Go Links
Confusion Matrix Analysis Help
Introduction to Confusion Matrix Analysis Interface
The Analysis Process
Common Error Messages
Content Editor Help
Introduction to Content Editor Interface
Edit Data
ROC Generator Help
Introduction to ROC Generator Interface
Generate ROC
The Analysis Process
Done
My Computer

To go to the next slide, click on the next button in the navigator panel on the slide below or choose the autoplay option for an auto slideshow mode.

1) Fill in all the textboxes in the Confusion Matrix Analysis column.

Figure 23: Help Documentation and the tutorial slides

23.0 FUTURE EXPECTATION

For future expansion of the system, I had discovered a few modifications that could be made to improve the current system. Firstly the welcome interface wizard function will be replaced with open and saving file function. With this, the user will be able to safe file in the form of Microsoft Access Database File (*.mdb) and open the same format of files.

Besides that, the system can also be equipped with the import / export functions that enables user to import or export the Microsoft Access Database Files (*.mdb). Another special feature that could be embedded onto the system is the Report Generator function. The function will allow user to generate a report that will neatly presents the final results that is the ROC curve and its database table.

Last but not least, the confusion matrix analysis on the current system only dealing with a specific type of confusion matrix which is the four by four (4 x 4) matrix, the future system will be able to handle various kinds of other confusion matrix in the analysis process.

CHAPTER 5
CONCLUSION

With the development of the performance measurement tool for data mining techniques, it might reduce some of the workloads required for the implementation of the Data Mining techniques in IDS. More over, the system is not just only simplify the manual process but also preserved the reliability of the IDS developed. The system ensures the accuracy in performance measure in order to determine the suitable Data Mining technique that can be implemented in the IDS. Although the system had covered the crucial stage (Interpretation and Evaluation) in KDD process, the accuracy of the results produced are totally depending on the inputs. Therefore there is still a room for future expansion on developing other reliable systems towards improving the current process.

REFERENCES

1. Hierarchical Clustering, Andrew W. Moore, Associate Professor, School of Computer Science, Carnegie Mellon University, www.cs.cmu.edu/~awm, awm@cs.cmu.edu

2. Anomaly Detection Using S Language Framework; clustering & Visualization of Intrusive attacks on Computer Systems, Khaled Labib and V. Rao Vemuri

3. K-Means; a new generalized K-Means clustering algorithm, Yui-Ming Cheung, www.computerscienceweb.com

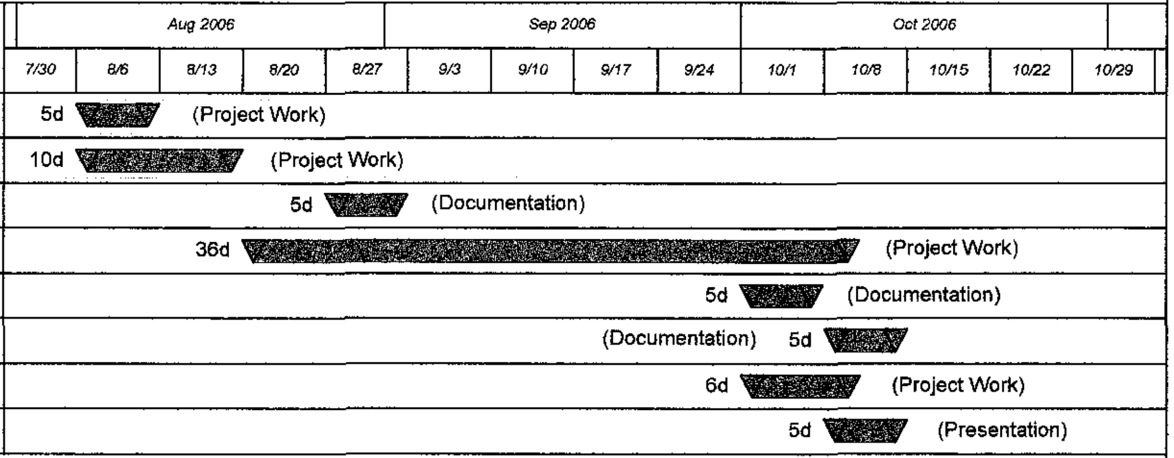
4. Learning Intrusion Detection Supervised or Unsupervised, Parel Laskov, Patick Dussel, Christina Shafe and Konrad Rieky

5. Unsupervised Anomaly Detection in Network Intusion Detection Using Clusters, Kingsly Leung, Christopher Leckie, caleckie@cs.mu.oz.au

APPENDIXES

Performance Measuring Tool for DM Techniques in IDS Timeline

ID	*Task Name*	*Start*	*Finish*
[illegible]	Problem Identification (Requirement Analysis)	8/7/2006	8/11/2006
[illegible]	Planning (Specification)	8/7/2006	8/18/2006
[illegible]	Progress report writing (1) & Submission	8/28/2006	9/1/2006
[illegible]	Prototype Development	8/21/2006	10/9/2006
[illegible]	Progress report writing (2) & Submission	10/2/2006	10/6/2006
[illegible]	Dissertation report writing	10/9/2006	10/13/2006
[illegible]	Deployment (Verification)	10/2/2006	10/9/2006
[illegible]	Oral Presentation	10/9/2006	10/13/2006



Project: Performance Measuring Tool for DM Techniques in IDS Timeline
Date: Thu, 9/14/2006

Task	1	▽▭▽	Milestone	0	◆	External Tasks	1
Progress	1	▨	External Milestone	0	◈		

SMO Results: Confusion Matrix
(Classify by: Classification)

Week 7

Week 7- 10% Normal 90% Attack
Table: week7_10n90a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      4996          99.92  %
Incorrectly Classified Instances       4           0.08  %
Kappa statistic                    0.9955
Mean absolute error                   0.0008
Root mean squared error                 0.0283
Relative absolute error              0.4441 %
Root relative squared error           9.4281 %
Total Number of Instances           5000

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  Class
  0.994    0        0.998    0.994    0.996    0
  1        0.006    0.999    1        1        1

=== Confusion Matrix ===

    a    b   <-- classified as
  497    3 |   a = 0
    1 4499 |   b = 1
```

Week 7- 20% Normal 80% Attack
Table: week7_20n80a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      4989          99.78  %
Incorrectly Classified Instances      11           0.22  %
Kappa statistic                    0.9931
Mean absolute error                   0.0022
Root mean squared error                 0.0469
Relative absolute error              0.6873 %
Root relative squared error          11.726  %
Total Number of Instances           5000

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  Class
  0.998    0.001    1        0.998    0.999    1
```

```
 0.999   0.003    0.99     0.999    0.995   0

=== Confusion Matrix ===

   a    b   <-- classified as
3990   10 |   a = 1
   1  999 |   b = 0
```

Week 7- 30% Normal 70% Attack
Table: week7_30n70a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4999              99.98   %
Incorrectly Classified Instances         1               0.02   %
Kappa statistic                      0.9995
Mean absolute error                     0.0002
Root mean squared error                   0.0141
Relative absolute error                 0.0476 %
Root relative squared error               3.0861 %
Total Number of Instances               5000

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
 0.999     0          1          0.999    1          0
 1         0.001      1          1        1          1

=== Confusion Matrix ===

   a    b   <-- classified as
 1499    1 |   a = 0
    0 3500 |   b = 1
```

Week 7- 40% Normal 60% Attack
Table: week7_40n60a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4989              99.78   %
Incorrectly Classified Instances        11               0.22   %
Kappa statistic                      0.9954
Mean absolute error                     0.0022
Root mean squared error                   0.0469
Relative absolute error                 0.4583 %
Root relative squared error               9.5743 %
Total Number of Instances               5000

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
```

```
 0.997    0.001     1       0.997    0.998   1
 1        0.003     0.995   1        0.997   0

=== Confusion Matrix ===

   a    b   <-- classified as
 2990   10 |   a = 1
    1 1999 |   b = 0
```

Week 7- 50% Normal 50% Attack (b)
Table: week7_50n50ab

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4997               99.94   %
Incorrectly Classified Instances         3                0.06   %
Kappa statistic                          0.9988
Mean absolute error                      0.0006
Root mean squared error                  0.0245
Relative absolute error                  0.12    %
Root relative squared error              4.899   %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  1         0.001     0.999     1         0.999    0
  0.999     0         1         0.999     0.999    1

=== Confusion Matrix ===

   a    b   <-- classified as
 2499    1 |   a = 0
    2 2498 |   b = 1
```

Week 7- 60% Normal 40% Attack
Table: week7_60n40a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4989               99.78   %
Incorrectly Classified Instances        11                0.22   %
Kappa statistic                          0.9954
Mean absolute error                      0.0022
Root mean squared error                  0.0469
Relative absolute error                  0.4583 %
Root relative squared error              9.5743 %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===
TP Rate  FP Rate  Precision  Recall  F-Measure  Class
 1        0.005     0.997     1        0.998   0
 0.995    0         0.999     0.995    0.997   1

=== Confusion Matrix ===

   a    b   <-- classified as
 2999    1 |   a = 0
   10 1990 |   b = 1
```

Week 7- 70% Normal 30% Attack
Table: week7_70n30a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4992              99.84  %
Incorrectly Classified Instances         8               0.16  %
Kappa statistic                      0.9962
Mean absolute error                     0.0016
Root mean squared error                   0.04
Relative absolute error                 0.3809 %
Root relative squared error              8.7287 %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  Class
 1        0.005     0.998     1        0.999   0
 0.995    0         1         0.995    0.997   1

=== Confusion Matrix ===

   a    b   <-- classified as
 3500    0 |   a = 0
    8 1492 |   b = 1
```

Week 7- 80% Normal 20% Attack
Table: week7_80n20a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4992              99.84  %
Incorrectly Classified Instances         8               0.16  %
Kappa statistic                      0.995
Mean absolute error                     0.0016
Root mean squared error                   0.04
Relative absolute error                 0.4999 %
Root relative squared error             10      %
Total Number of Instances             5000
```

```
=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.993     0         0.999     0.993     0.996     1
  1         0.007     0.998     1         0.999     0

=== Confusion Matrix ===

    a    b   <-- classified as
  993    7 |   a = 1
    1 3999 |   b = 0
```

Week 7- 90% Normal 10% Attack
Table: week7_90n10a

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4994               99.88   %
Incorrectly Classified Instances         6                0.12   %
Kappa statistic                          0.9933
Mean absolute error                      0.0012
Root mean squared error                  0.0346
Relative absolute error                  0.6661 %
Root relative squared error             11.547  %
Total Number of Instances             5000

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  1         0.012     0.999     1         0.999     0
  0.988     0         1         0.988     0.994     1

=== Confusion Matrix ===

    a    b   <-- classified as
 4500    0 |   a = 0
    6  494 |   b = 1
```

ECTION C: DETAIL REPORT

TUDENT'S NAME & NO: MUHAMMAD FIRDAUS BIN ROSLAN 2128

/EEK NO: 20

'BJECTIVE (S) OF THE ACTIVITIES:

- To evaluate the 5000 of random data of the DARPA 1998 Evaluation Dataset at one time using the SVM model algorithm, the SMO function in the WEKA program.
- Calculate the Detection Rate and the False Alarm Rate based on the results.

CONTENTS:

After a one week of the Hari Raya break, I continue with the SMO evaluation on the 5000 random data on week 7, table of 40% normal data and 60% attack. The results of week 7 is as shown below:

Table	Normal	Attack	Detection Rate (%)	False Alarm (%)
week7_10n90a	500	4500	**99.93**	**0.20**
week7_20n80a	1000	4000	**99.98**	**1.00**
week7_30n70a	1500	3500	**99.97**	**0.00**
week7_40n60a	2000	3000	**99.97**	**0.50**
week7_50n50aa	2500	2500	**99.92**	**0.10**
week7_50n50ab	2500	2500	**99.96**	**0.10**
week7_60n40a	3000	2000	**99.95**	**0.33**
week7_70n30a	3500	1500	**100.00**	**0.23**
week7_80n20a	4000	1000	**99.90**	**0.18**
week7_90n10a	4500	500	**100.00**	**0.13**

Detection Rate and False Alarm Rate Calculation for Week 7:

The calculation on the Detection Rate and the False Alarm Rate is done according to the confusion matrix gained from the SMO results. The SMO evaluates 5000 data, which is randomly fetched from the DARPA 1998 IDS Evaluation datasets. Each table of 5000 random data is consists of fixed amount of random attack data and normal data, which is determined according to the percentage.

There are ten tables all together consist of 10%, 20%, 30%, 40%, 50% (Table A), 50% (Table B), 60%, 70%, 80%, 90% of random attack data and the rest of the 5000 data is the normal data.

Formula used to calculate the Detection Rate and the False Alarm Rate:

$$\textbf{Detection Rate} = 1 - \frac{\textbf{Amount of Attack Recognized as Normal (NA)}}{\textbf{Total Amount Attack}} \textbf{ X } \textbf{100\%}$$

$$\textbf{False Alarm Rate} = \frac{\textbf{Normal Data Recognized as Attack (AN)}}{\textbf{Total Amount of Normal}} \textbf{ X } \textbf{100\%}$$

<table>
<tr><th colspan="2">Table: week7_10n90a</th></tr>
<tr><td>NA=3
Detection Rate = 1- $\frac{3}{4500}$ = 0.9993
= 0.9993 x 100%
= **99.93%**</td><td>AN=1
False Alarm Rate = $\frac{1}{500}$ = 0.002
= 0.002 x 100%
= **0.20%**</td></tr>
<tr><th colspan="2">Table: week7_20n80a</th></tr>
<tr><td>NA=1
Detection Rate = 1- $\frac{1}{4000}$ = 0.9998
= 0.9998 x 100%
= **99.98%**</td><td>AN=10
False Alarm Rate = $\frac{10}{1000}$ = 0.01
= 0.01 x 100%
= **1.00%**</td></tr>
<tr><th colspan="2">Table: week7_30n70a</th></tr>
<tr><td>NA=1
Detection Rate = 1- $\frac{1}{3500}$ = 0.9997
= 99.97 x 100%
= **99.97%**</td><td>AN=0
False Alarm Rate = $\frac{0}{1500}$ = 0.00
= 0.00 x 100%
= **0.00%**</td></tr>
</table>

Table: week7_40n60a	
NA=1 **Detection Rate** = 1- $\frac{1}{3000}$ = 0.9997 = 0.9997 x 100% **= 99.97%**	AN=10 **False Alarm Rate** = $\frac{10}{2000}$ = 0.005 = 0.005 x 100% **= 0.5%**
Table: week7_50n50aa (Table A)	
NA=2 **Detection Rate** = 1- $\frac{2}{2500}$ = 0.9992 = 0.9992 x 100% **= 99.92%**	AN=2 **False Alarm Rate** = $\frac{2}{2500}$ = 0.001 = 0.001 x 100% **= 0.10%**
Table: week7_50n50ab (Table B)	
NA=1 **Detection Rate** = 1- $\frac{1}{2500}$ = 0.9996 =0. 9996 x 100% **= 99.96%**	AN=2 **False Alarm Rate** = $\frac{2}{2500}$ = 0.001 = 0.001 x 100% **= 0.10%**
Table: week7_60n40a	
NA=1 **Detection Rate** = 1- $\frac{1}{2000}$ = 0.9995 = 0.9995 x 100% **= 99.95%**	AN=10 **False Alarm Rate** = $\frac{10}{3000}$ = 0.0033 = 0.0033 x 100% **= 0.33%**
Table: week7_70n30a	
NA=0 **Detection Rate** = 1- $\frac{0}{1500}$ = 1.00 = 1.00 x 100% **= 100%**	AN=8 **False Alarm Rate** = $\frac{8}{3500}$ = 0.0023 = 0.0023 x 100% **= 0.23%**

Table: week7_80n20a	
NA=1 **Detection Rate** = $1- \frac{1}{1000} = 0.999$ = 0.999 x 100% = **99.90%**	AN=7 **False Alarm Rate** = $\frac{7}{4000} = 0.0018$ = 0.0018 x 100% = **0.18%**

Table: week7_90n10a	
NA=0 **Detection Rate** = $1- \frac{0}{500} = 1.00$ = 1.00 x 100% = **100%**	AN=6 **False Alarm Rate** = $\frac{6}{4500} = 0.0013$ = 0.0013 x 100% = **0.13%**

VERIFIED BY (PLANT SUPERVISOR),
INITIAL:

ECTION C: DETAIL REPORT

TUDENT'S NAME & NO: MUHAMMAD FIRDAUS BIN ROSLAN 2128

VEEK NO: 20

BJECTIVE (S) OF THE ACTIVITIES:

- To evaluate the 5000 of random data of the DARPA 1998 Evaluation Dataset at one time using the SVM model algorithm, the SMO function in the WEKA program.
- Calculate the Detection Rate and the False Alarm Rate based on the results.

CONTENTS:

After a one week of the Hari Raya break, I continue with the SMO evaluation on the 5000 random data on week 7, table of 40% normal data and 60% attack. The results of week 7 is as shown below:

Table	Normal	Attack	Detection Rate (%)	False Alarm (%)
week7_10n90a	500	4500	**99.93**	**0.20**
week7_20n80a	1000	4000	**99.98**	**1.00**
week7_30n70a	1500	3500	**99.97**	**0.00**
week7_40n60a	2000	3000	**99.97**	**0.50**
week7_50n50aa	2500	2500	**99.92**	**0.10**
week7_50n50ab	2500	2500	**99.96**	**0.10**
week7_60n40a	3000	2000	**99.95**	**0.33**
week7_70n30a	3500	1500	**100.00**	**0.23**
week7_80n20a	4000	1000	**99.90**	**0.18**
week7_90n10a	4500	500	**100.00**	**0.13**

Detection Rate and False Alarm Rate Calculation for Week 7:

The calculation on the Detection Rate and the False Alarm Rate is done according to the confusion matrix gained from the SMO results. The SMO evaluates 5000 data, which is randomly fetched from the DARPA 1998 IDS Evaluation datasets. Each table of 5000 random data is consists of fixed amount of random attack data and normal data, which is determined according to the percentage.

There are ten tables all together consist of 10%, 20%, 30%, 40%, 50% (Table A), 50% (Table B), 60%, 70%, 80%, 90% of random attack data and the rest of the 5000 data is the normal data.

Formula used to calculate the Detection Rate and the False Alarm Rate:

$$\textbf{Detection Rate} = 1 - \frac{\underline{\textbf{Amount of Attack Recognized as Normal (NA)}}}{\textbf{Total Amount Attack}} \ \textbf{X} \ \textbf{100\%}$$

$$\textbf{False Alarm Rate} = \frac{\underline{\textbf{Normal Data Recognized as Attack (AN)}}}{\textbf{Total Amount of Normal}} \ \textbf{X} \ \textbf{100\%}$$

Table: week7_10n90a	
NA=3 **Detection Rate** = 1- $\frac{3}{4500}$ = 0.9993 = 0.9993 x 100% = **99.93%**	**AN**=1 **False Alarm Rate** = $\frac{1}{500}$ = 0.002 = 0.002 x 100% = **0.20%**

Table: week7_20n80a	
NA=1 **Detection Rate** = 1- $\frac{1}{4000}$ = 0.9998 = 0.9998 x 100% = **99.98%**	**AN**=10 **False Alarm Rate** = $\frac{10}{1000}$ = 0.01 = 0.01 x 100% = **1.00%**

Table: week7_30n70a	
NA=1 **Detection Rate** = 1- $\frac{1}{3500}$ = 0.9997 = 99.97 x 100% = **99.97%**	**AN**=0 **False Alarm Rate** = $\frac{0}{1500}$ = 0.00 = 0.00 x 100% = **0.00%**

Table: week7_40n60a	
NA=1 **Detection Rate** = $1-\frac{1}{3000}$ = 0.9997 = 0.9997 x 100% = **99.97%**	AN=10 **False Alarm Rate** = $\frac{10}{2000}$ = 0.005 = 0.005 x 100% = **0.5%**

Table: week7_50n50aa (Table A)	
NA=2 **Detection Rate** = $1-\frac{2}{2500}$ = 0.9992 = 0.9992 x 100% = **99.92%**	AN=2 **False Alarm Rate** = $\frac{2}{2500}$ = 0.001 = 0.001 x 100% = **0.10%**

Table: week7_50n50ab (Table B)	
NA=1 **Detection Rate** = $1-\frac{1}{2500}$ = 0.9996 =0. 9996 x 100% = **99.96%**	AN=2 **False Alarm Rate** = $\frac{2}{2500}$ = 0.001 = 0.001 x 100% = **0.10%**

Table: week7_60n40a	
NA=1 **Detection Rate** = $1-\frac{1}{2000}$ = 0.9995 = 0.9995 x 100% = **99.95%**	AN=10 **False Alarm Rate** = $\frac{10}{3000}$ = 0.0033 = 0.0033 x 100% = **0.33%**

Table: week7_70n30a	
NA=0 **Detection Rate** = $1-\frac{0}{1500}$ = 1.00 = 1.00 x 100% = **100%**	AN=8 **False Alarm Rate** = $\frac{8}{3500}$ = 0.0023 = 0.0023 x 100% = **0.23%**

Table: week7_80n20a	
NA=1 **Detection Rate** $= 1 - \frac{1}{1000} = 0.999$ $= 0.999 \times 100\%$ **= 99.90%**	AN=7 **False Alarm Rate** $= \frac{7}{4000} = 0.0018$ $= 0.0018 \times 100\%$ **= 0.18%**
Table: week7_90n10a	
NA=0 **Detection Rate** $= 1 - \frac{0}{500} = 1.00$ $= 1.00 \times 100\%$ **= 100%**	AN=6 **False Alarm Rate** $= \frac{6}{4500} = 0.0013$ $= 0.0013 \times 100\%$ **= 0.13%**

VERIFIED BY (PLANT SUPERVISOR),
INITIAL:

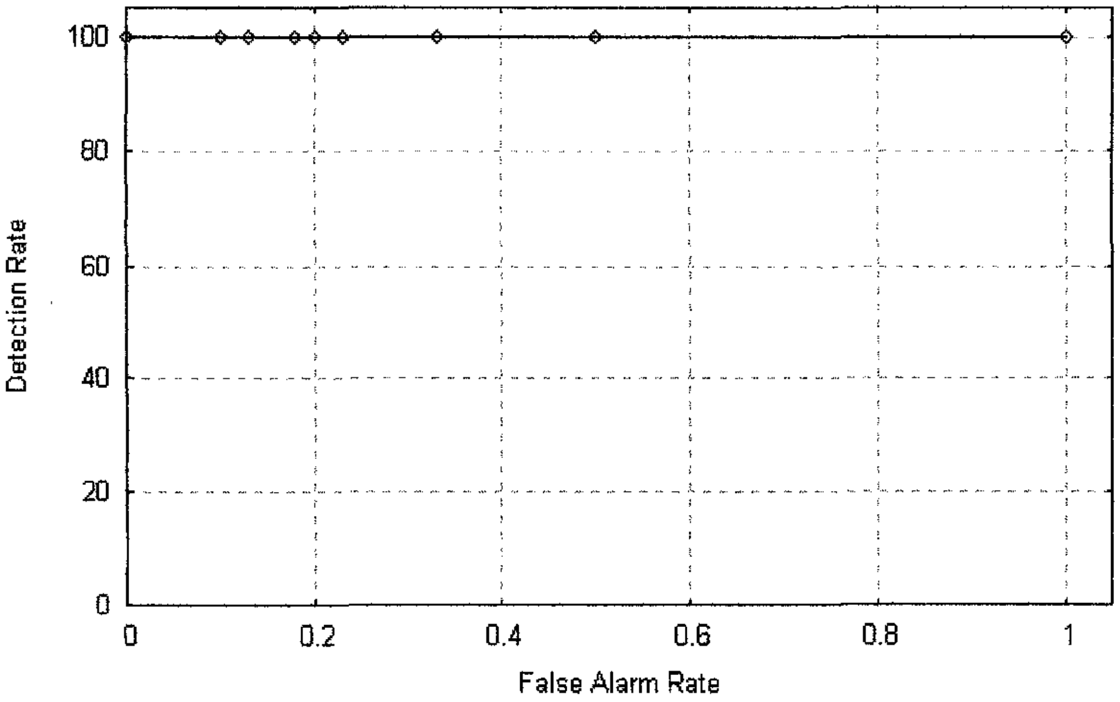
ATTACHMENT B:
EXPERIMENT 1 (5000 DATA)

EXPERIMENTAL RESULTS WITH DIFFERENT PARAMETERS (A) DETECTION RATE (B) FALSE ALARM RATE FOR SVM (5000 DATA)

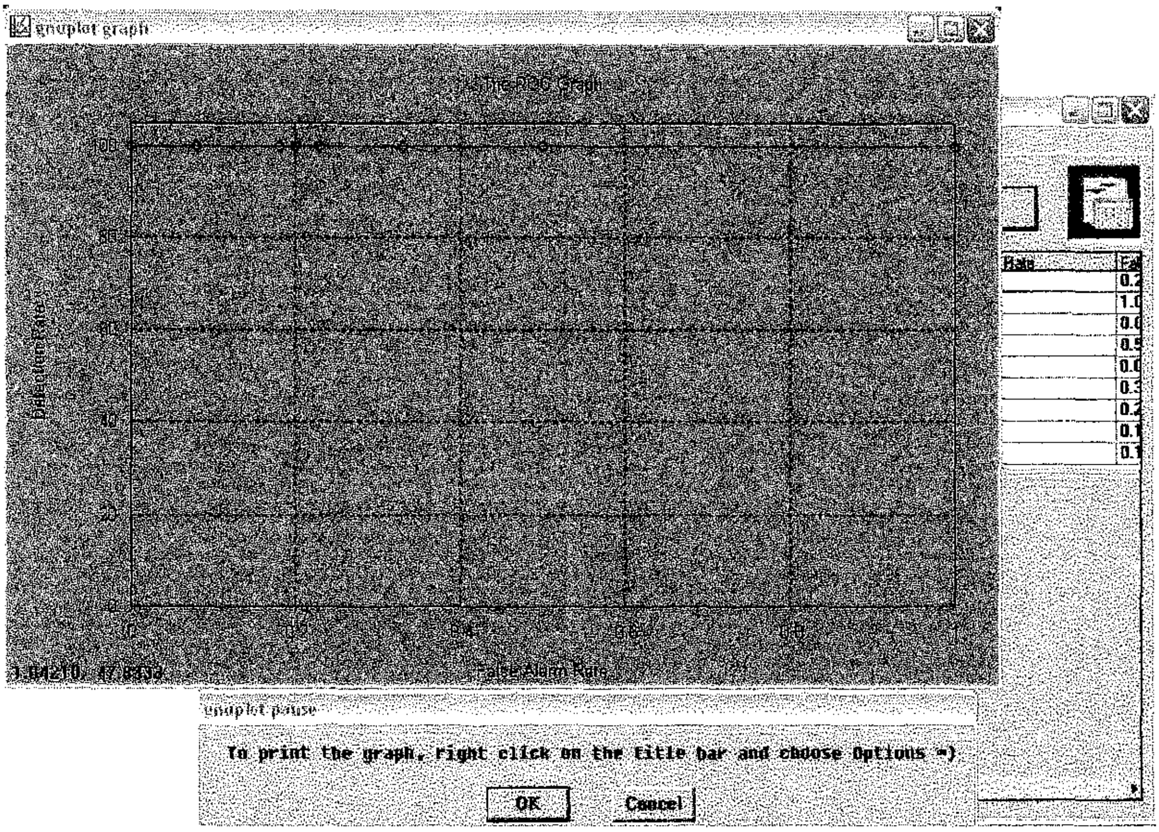
Week 7 (5000)

Table	Normal	Attack	Detection Rate (%)	False Alarm (%)
week7_10n90a	500	4500	**99.93**	**0.20**
week7_20n80a	1000	4000	**99.98**	**1.00**
week7_30n70a	1500	3500	**99.97**	**0.00**
week7_40n60a	2000	3000	**99.97**	**0.50**
week7_50n50aa	2500	2500	**99.92**	**0.10**
week7_50n50ab	2500	2500	**99.96**	**0.10**
week7_60n40a	3000	2000	**99.95**	**0.33**
week7_70n30a	3500	1500	**100.00**	**0.23**
week7_80n20a	4000	1000	**99.90**	**0.18**
week7_90n10a	4500	500	**100.00**	**0.13**

ROC: Week 7 (5000)

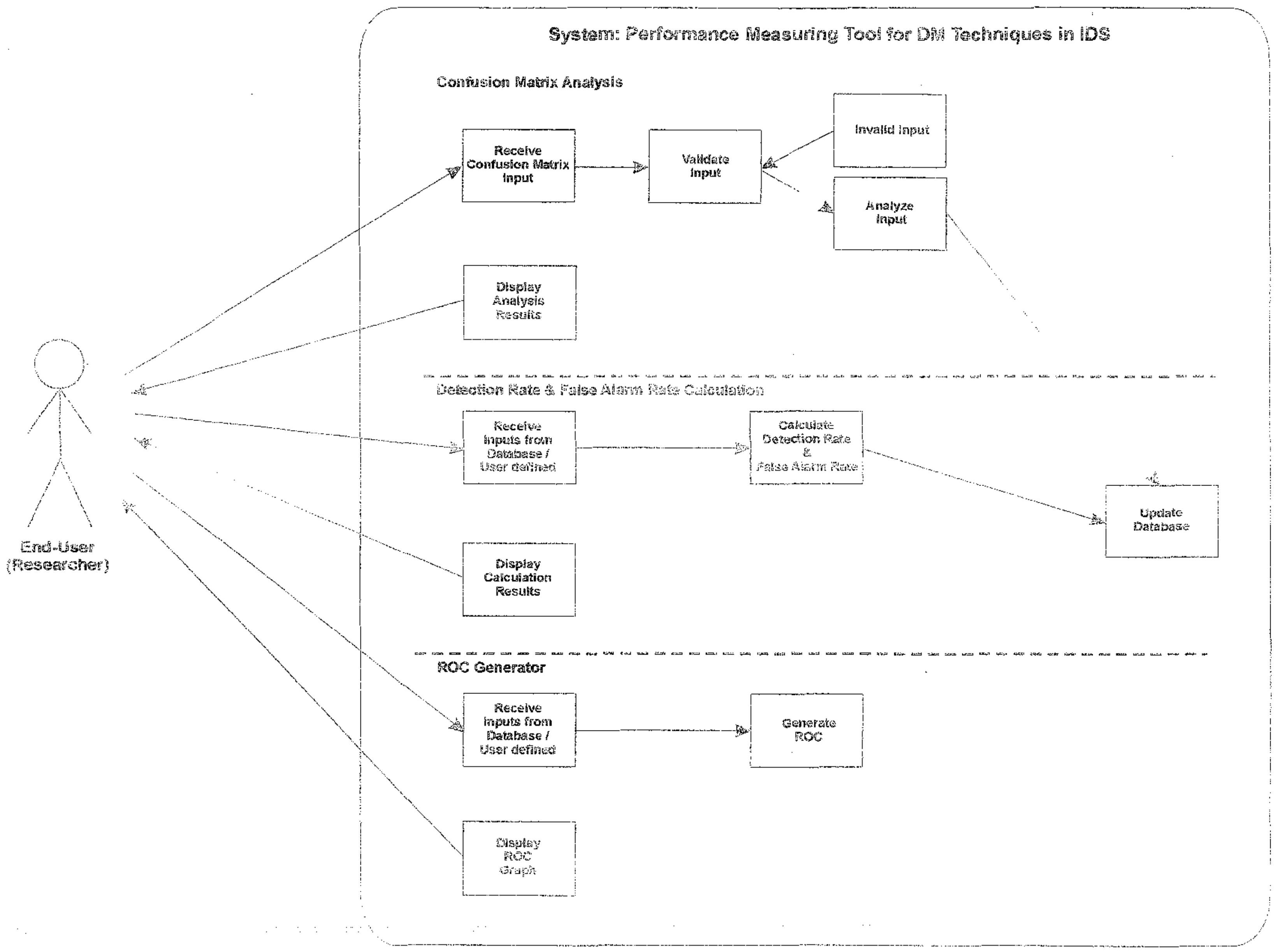


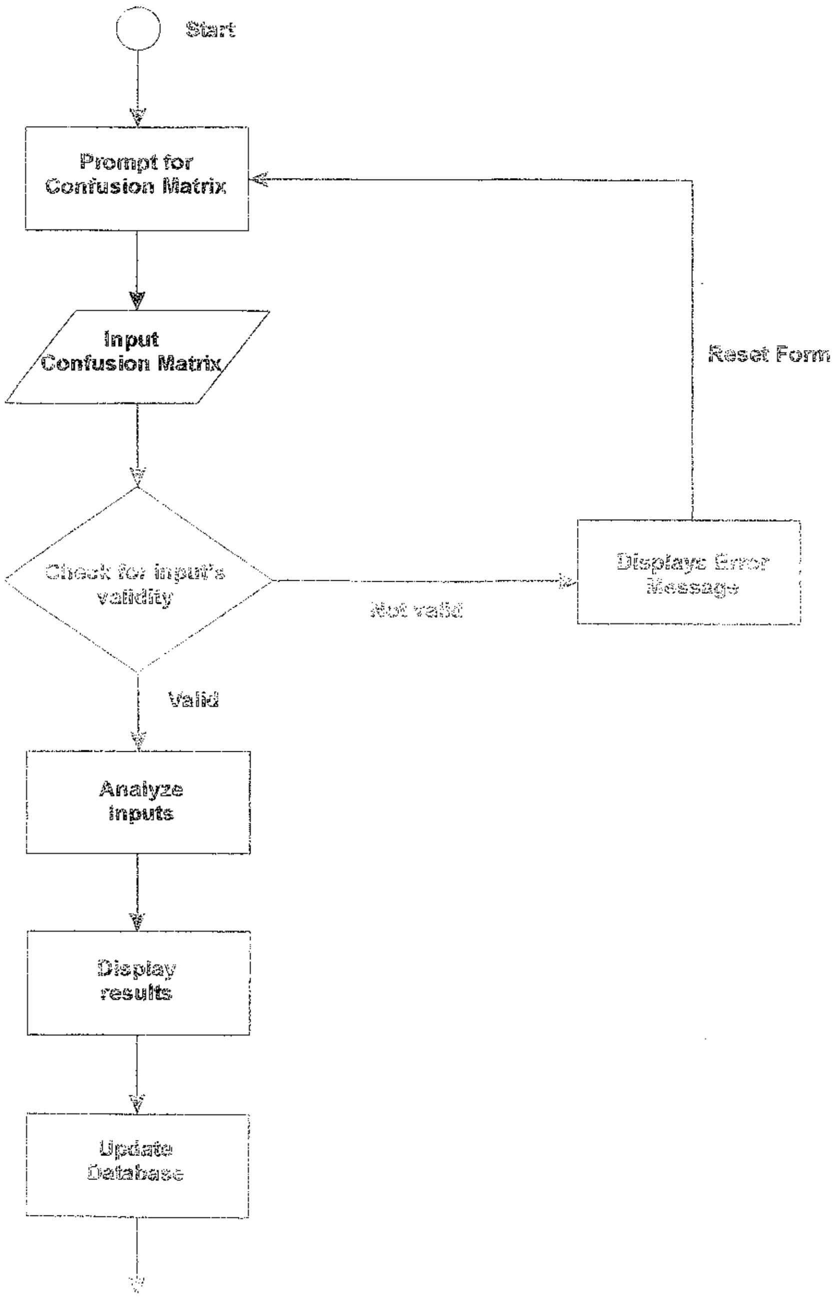
The Prototype's Test Result:

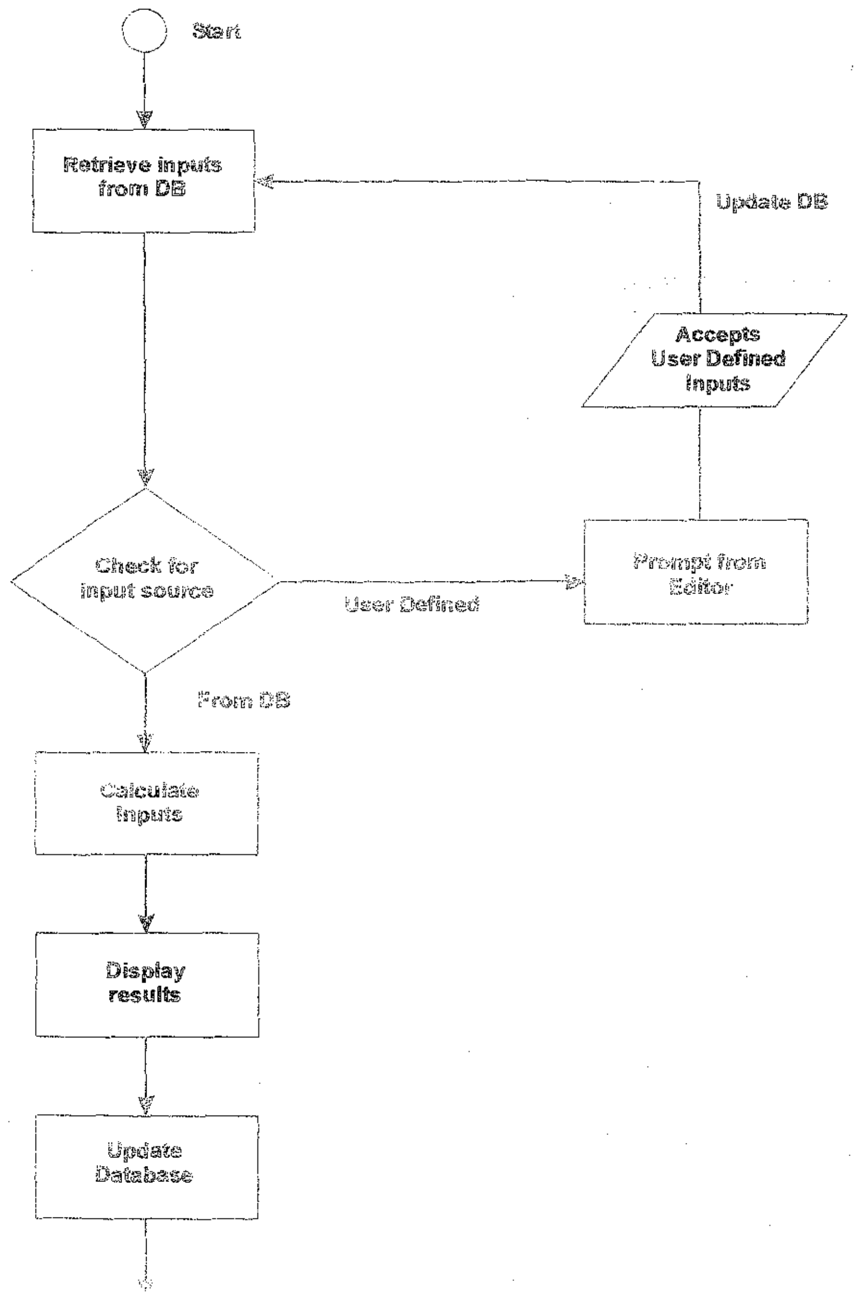


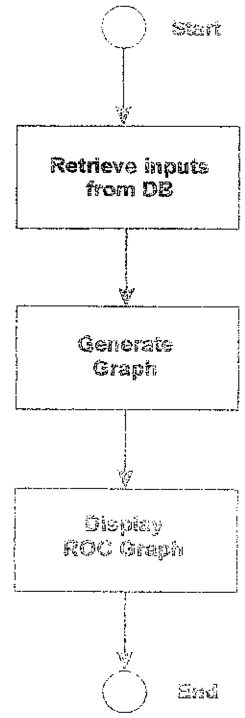
To print the graph, right click on the title bar and choose Options =)

Figure1: The test results of Week 7 SMO Confusion Matrix Analysis of 5000 data.

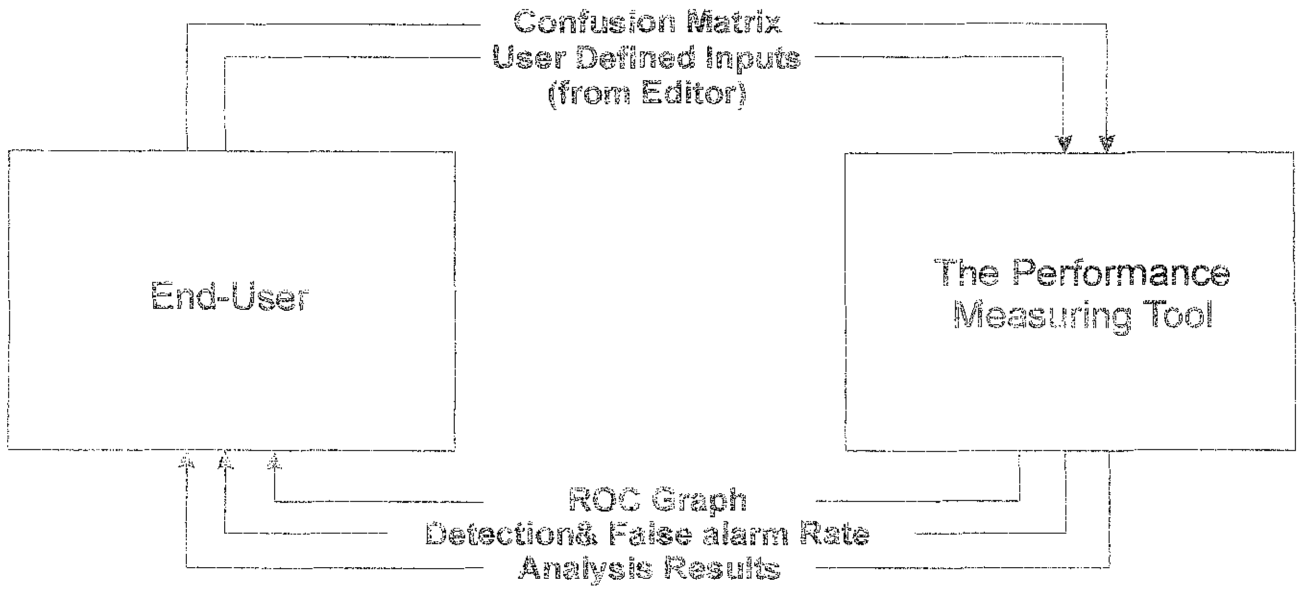

System: Performance Measuring Tool for DM Techniques in IDS
Confusion Matrix Analysis
Receive Confusion Matrix Input
Validate Input
Invalid Input
Analyze Input
Display Analysis Results
Detection Rate & False Alarm Rate Calculation
Receive Inputs from Database / User defined
Calculate Detection Rate & False Alarm Rate
Update Database
Display Calculation Results
ROC Generator
Receive Inputs from Database / User defined
Generate ROC
Display ROC Graph
End-User (Researcher)


Start
Prompt for Confusion Matrix
Input Confusion Matrix
Check for input's validity
Not valid
Displays Error Message
Reset Form
Valid
Analyze inputs
Display results
Update Database


Start
Retrieve inputs from DB
Check for input source
User Defined
Prompt from Editor
Accepts User Defined Inputs
Update DB
From DB
Calculate Inputs
Display results
Update Database


Start
Retrieve inputs from DB
Generate Graph
Display ROC Graph
End

Context Diagram


Confusion Matrix
User Defined Inputs
(from Editor)
End-User
The Performance
Measuring Tool
ROC Graph
Detection& False alarm Rate
Analysis Results