ALTINBAS UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

CLASSIFICATION SYSTEM IDS ALERTS BY USING DATA MINING TECHNIQUE

M. Sc. Thesis

NOOR ABDULKHALEQ ALAZZAWI

ISTANBUL, 2017

CLASSIFICATION SYSTEM IDS ALERTS BY USING DATA MINING TECHNIQUE

by

Noor Abdulkhaleq Alazzawi

Master degree, Altinbas University, 2017

Submitted to the Graduate Faculty of

Science and Engineering in partial fulfillment

of the requirements for the degree of

Master of Electrical and Computer Engineering

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Adil Deniz Duru

____

Co-Supervisor

Asst. Prof. Yasa Ekşioğlu Özok

____

Supervisor

Examining Committee Members (first name belongs to the chairperson of the jury and the second name belongs to supervisor)

(Asst. Prof. Yasa Ekşioğlu Özok) ______________________

(Assist. Prof. Adil Deniz Duru) ______________________

(Assist. Prof. Dilek Göksel Duru) ______________________

(Asst. Prof. Çağatay AYDIN) ______________________

(Assist. Prof. Emrullah Fatih Yetkin) ______________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Çağatay AYDIN

____

Head of Department

Assoc. Prof. Oğuz BAYAT

____

Approval of [Institution] ____/____/____

Director

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

NOOR ABDULKHALEQ

ACKNOWLEDGEMENTS

The satisfaction that accompanies the successful completion of this thesis would be incomplete without mentioning the people who made this possible and whose constant guidance and encouragement were my driving force during the course of this work. I consider myself privileged to express gratitude and respect to all those who guided me through the course of this thesis. I am extremely grateful to my major professor, Dr.Yasa Ekşioğlu Özok for her effective guidance. Also, special thanks to Dr. Adil Duru for His constant support and mentoring and encouragement right from the nascent stages of this work through its evaluation, which has played a major part in the successful completion of this work.and for all committee members. In addition, I would like to thank my family and my friends for their support to me.

ABSTRACT

CLASSIFICATION SYSTEM IDS ALERTS BY USING DATA MINING

NOOR ABDULKHALEQ ALAZZAWI,

M.S, Electrical and Computer Engineering, Altınbaş University,

Supervisor: Assist. Prof. Yasa Ekşioğlu Özok

Co-Supervisor: Asst. Prof. Dr. Adil Deniz Duru

Date: December,2017

Currently, people are living in a world without borders, which means that nothing is beyond reach. The significant growth in technology has led to new threats in the era of computing. These risks are increasing and we should be dealing with them in a more efficient manner. Therefore, it has become necessary for researchers to focus on protecting networks and to work on the production of software for this purpose, namely 'an intrusion detection system' (IDs) .IDS can reveal various types of attacks and analyze events that arise in networks and computer systems to identify any protection problem. However, an IDS generates a considerable number of alerts each day most of which may be false alarms. Therefore, researchers have attempted to find ways to solve the problem of false alerts. One of these methods is data mining algorithms, which is a process of mining knowledge from huge datasets. Data mining may be suitable for dealing with this large number of alerts. This research presents a methodology involving an improved data mining technique to classify alarms as being a real attack or a false attack. This technique is used in designing the proposed classification system. An application has been designed using C# to test the dataset. The classification system is tested by conducting three experiments on the second, fourth, and fifth week of the DARPA 1999 dataset which extracted from a simulation of a military management network. Each experiment produces high accuracy to classify the alerts in order to facilitate the process of analyzing alerts to help security analysts to distinguish between true and false alerts. The first experiment is conducted on the second week with percentage of false alert (PFA) and percentage of true alerts (PTA) equaling 95%, and 5%, respectively. The second test was conducted on the fourth week and the PFA and PTA equaled 94.19% and 5.81%, respectively. The third experiment was conducted in the fifth week with the PFA and PTA equaling 93.768% and 6.232%, respectively. The proposed system achieved the best results when compared with the literature findings that had used the same dataset.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IDS	: Intrusion detection system.
I.D	: Intrusion detection
HIDS	: Host intrusion detection.
SB	: Signature Base detection
AD	: Anomaly Base detection
NIDS	: Network intrusion detection.
D.M.	: Data mining
A.R	: Association Rules.
TID	: Transaction ID.
IPd	: IP destination address.
IPs	: IP source address (IPs).
Pd	: Port destination address.
Ps	: Port source address.
Min.sup	: Minimum support.
FIS	: Frequent Itemsets
PFA	: Percentage of the false alerts.
PTA	: Percentage of the True alerts.
Tv	: Threat score of the IDS alert

1. INTRODUCTION

1.1 Overview

Recently, web applications and the Internet pervade our life and businesses; everyone uses the Internet for their benefit. In addition, organizations are using the Internet to expand their businesses through communication. Moreover, modern businesses cannot achieve progress effectively without the Internet. However, these great benefits of the Internet bring to us critical security issues where any attacks on a network via the Internet can negate these benefits in case the web application can be stolen. In particular, organizations can suffer from this severity where networks are heavily used. This expansion was accompanied by several threats to the network through the spread of many types of malware that reduce the efficiency of networks, especially in terms of the transmission of data over the network [1].

These days, information is becoming increasingly based on interactive processing and speed of access; therefore, much information is saved on computers. However, with the increase in the numbers of cheap computers and the weakness network security, the problem of unauthorized access from outside or inside an organization has exacerbated. Moreover, tampering with data can occur by providing an access path to data from any place on a network. With understanding of how the system works, an intruder becomes proficient at discovering any weakness and flaw in the system and exploits these to obtain the privileges that enable him to do anything to destroy the system [2].

Moreover, in this age of the Internet where the Internet reaches into each home along with the evolution of Internet usage, the Internet has become more suitable for people’s lives, such as consumer's need for Internet banking and Internet shopping. In addition, the military and government organizations depend on computers in their work. However, there are concerns about information stored on their operating systems being hacked [3].

Therefore, many techniques have appeared to reduce these risks, including user authentication via passwords and/or biometrics, software techniques, data protection such as decryption and encryption, and firewalls. However, these techniques were not sufficient to prevent intruders. This problem has prompted researchers to develop new techniques to explore and hinder such threats. Computer intrusions have become an increasingly serious problem in the past few years, so it has become necessary to focus on protecting networks and work on the production of applications such as intrusion detection systems [1].

1.2 Intrusion Detection

Intrusion Detection (I.D.) is the process of identifying people who use network systems without permission and who have legitimate access to a computer or network system but without privileges [4, 5].

The objective of I.D. is to observe network assets to reveal any misuse and anomalous conduct in the network. The term 'intrusion detection' was first introduced in 1998 by Jim Anderson after developing the Internet, which was accompanied by an end of the control on observation of threats. Anderson encouraged the concept of recording user activity with computer logs to protect information from external and internal access by unauthorized persons and to protect information from people who misuse their privileges. Additionally, an intrusion detection system can be considered to be a theft alert. To illustrate, a house can be protected from theft by using a lock system. In case anyone attempts to break the lock in order to enter the house, the theft alert will detect that the lock has been compromised and alert the house owner by sending an alarm [6].

IDS can be a hardware or a software structure that plays an essential role in detecting various types of attack. It is also responsible for monitoring and analyzing any emerging events on a computer and/or network system to determine all security issues. An IDS has three major security stages: (1) monitoring; (2) detection, and (3) responding to unauthorized activities [7, 8]. Fig .1 shows IDS stages [7].

In the first stage, the data are collected and converted into a common format, stored and then sent to the next stage. The second phase is detection, where the formatted data obtained from the first stage are analyzed to reveal any intrusion attempt which are considered to be malicious and are thus redirected to the final stage. The third step is a response of classification into two types the first of which is passive wherein the IDs discovers and records any intrusion attempts and abnormal behavior. However, it cannot release an alarm to the administrator. The second type is an active response which detects attacks and abnormal behavior and responds by raising an alarm to the administrator [7, 9].

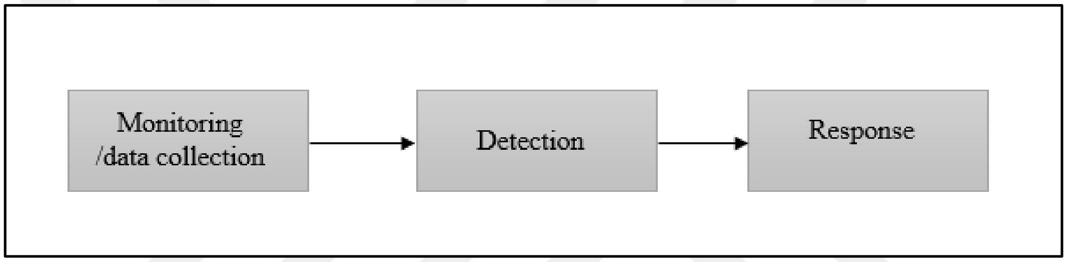


Figure 1. IDs stages [9]

An IDS do not prevent the intrusions from appearing; but, it detects them and making makes a report for operation. Moreover, it observes different sources of targeted activities, gathers and checks the data which is audited to find evidence for any exotic behaviors. When it discovers malicious suspicions, an alert is sent to the network administrator to give an ability for a rapid response attempts; such a response is cutting the network connection [10].

ID system is basic element of network security. It detects all intrusions which threaten the fundamental security objectives, availability, confidentiality and integrity [9].

There are some factors to motivate the intrusion detection system:

1- There are many systems that have security errors that make them vulnerable to threat. and it is difficult to detect these errors

2- Several organizations consider the risks that come from inside more harmful than the outside because internal risks are done by users have privileges.
3- Many new kinds of intrusions appear that, security methods need to be developed and improved to protect the systems against new attacks.

Overall, Intrusion detection system consisting of two discovery methods: Fig.2 explains Intrusions detection methods [4].

- Signature (Misuse) based (Sb) detection

- Anomaly based (Ab) detection

A Signature based has a database which contains signatures that have been generated from previous intrusions. Such a system is used to identify current malicious entities [4]. It works by sniffing all packets which pass through it, followed by a matching procedure for each packet it has sniffed with a signature in the database to determine whether the packet matches a signature in the database. The intrusion detection system will create alerts which are sent to an administrator of the network or they can be recorded for future inspection [11]. Misuse detection is denoted as a signature-based detection for generated alerts that depend on special attack signatures, these attacks specifically include activity or particular traffic which depend on known intervention activity. This generates an accurate result and fewer false alerts [12, 13].

An example of the misuse based system is the subject of the email which is free pictures and the name of the attached file is freepics.exe, and these specifications are known as malicious programs. In case the attackers change the file name from freepics.exe to freepics2.exe, the misuse discovery will not reveal this malicious program [14].

The anomaly based system has a baseline on how the system behaves normally [4], where it depends on audit data collected during a period of natural operation. Any defect that appears and is not identical to the baseline is considered to be an intrusion. An anomaly is sometimes called behavior-based detection because it is related to a user's behavior toward the system [15, 16]. It can detect unknown attacks as well as novel attacks depending on the audit data that do not depend on previous knowledge of detecting attacks, such as the misuse based system.

The example on the anomaly based system is when the administrator tries to enter to the network and writes the username many times because he/she forgets it. So, the IDs will consider this action like an intrusion because this action may come from an attacker who tries to penetrate the network.

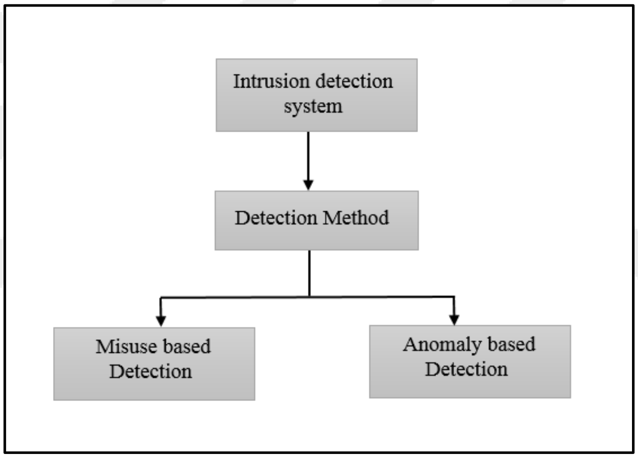


Figure 2. Intrusions detection methods [4]

Intrusion detection system is classified into three types of categories which depend on the kind of information that they monitor. And as shows in Fig. 3 these categories are:

- Network ID system (NIDS).

- Host ID system (HIDS).

- Hybrid ID system.

NIDS: A network is a connection between two or more computers to exchange data and share resources [14]. An NID system is used to observe and analysis the traffic of the network to protect the system from any risk [12]. It detects malicious activities such as 'port scans' and 'denial-of-service' attacks (DOS) [15]. In addition to monitoring network traffic attacks, an NIDS has many sensors to observe packet traffic. It checks the passage of packet after packet in near real time, or occasionally real time, in order to find infiltration patterns. It also analyzes any events in the network, including IP address protocol usage, etc. [4].

Network-based intrusion detection has some advantages:

i- it detects network attacks.

ii- it is very easy to deploy

iii- it reveal the failed attacks

HIDS: which needs small agents installed on separate systems for supervision. The small programs check the OS (operating system) and write data to a log file and send alerts. It detects any change in the OS and checks the checksum to ensure that system files do not change in a harmful manner [15]. In cases of any unauthorized activity or change in the OS, it will be detected immediately and alert the user [16]. HIDS has provided more relevant information than NIDS such that it provides an accurate description of network analysis, including reporting attacker activity such as which commands were used and what files were opened.

Host-based intrusion detection has some advantages:

i- it has the ability to detect the attacks that the NID system cannot reveal them.

ii- it does not need additional hardware.

iii- it observes the system activity.

A hybrid ID system combines a network IDs and host IDs. It is flexible and increases level security. Locations of IDS sensors are gathered and reported. A hybrid ID system works on a certain segment of a network or on the entire network [5].

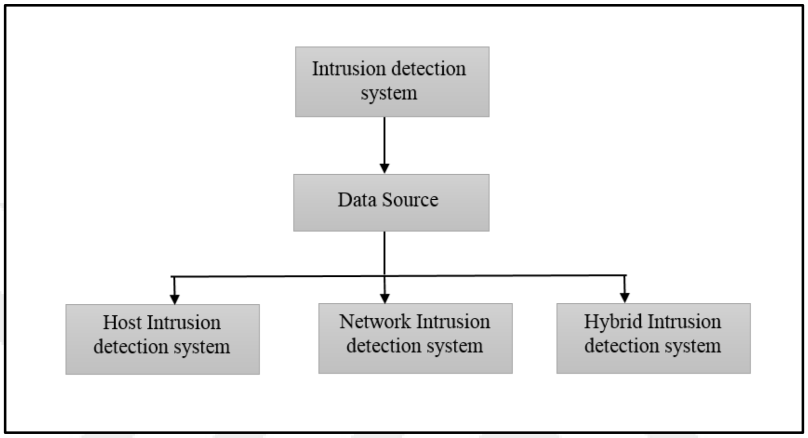

Intrusion detection system
Data Source
Host Intrusion detection system
Network Intrusion detection system
Hybrid Intrusion detection system

Figure 3. Classification of intrusion detection system [17]

1.3 Problem statement

IDs is the line of defense to protect a network from attacks. It checks packets to acquire evidence of intrusive behavior. An alert is routed when it detects an intrusive event. However, it gives an unmanageable number of alarms with 99% of them being false alerts [18-20]. The abundance of false alarms makes it difficult for the administrator to discover the true attacks and take immediate actions. Such alarms are not classified based on their degree of seriousness. Therefore, it important to find technique to solve this problem by classify the alerts into true alerts and false alerts to enable the network analyst to distinguish between them.

It is not unreasonable to avoid the unnecessary IDS alarms and false positives in different environments. Many approaches have been used to solve this problem and data mining techniques are the most important because they deal with massive amounts of data and intrusion detection systems produce many alerts. Therefore, data mining will be suitable to deal with a large number of alerts [21, 23]

1.4 literature survey

There are many researchers have tried to solve the problem of finding the false alerts from a huge number of IDS alerts. Some of these researchers are:

(Spathoulas and Katsikas, 2010) Proposed a post processing filter based on high alert frequency and neighboring alerts. The scheme of the proposed filter is based on two main hypothesizes. The first hypothesis indicates that the alarm is more potential to be true, in case it takes place in a higher repeat compared to the average repeat of the same alarms signature. The second hypothesis refers that there is a great difference in the number of alerts that have contiguous distributions from true positives to false positives. The proposed system contains three elements; Usual False Positives (UFP) element, Neighboring Related alarm (NRA) element, and   High Alarm Frequency (HAF) element.

The action mechanism of these elements is based on examining any alert enters. Each component is evaluated and given the degree of likelihood of a positive alert. After that, a calculation of the sum of total scores performed depending on this outcome which is determined by that the alarm regarding a positive true or false. The evaluation is accomplished by using the DARPA 1999 data (DARPA 1999 is extracted from a simulation of a military management network of 5 weeks). It has been shown that the approach has found the number of the false alerts as the average of 75% [24].

(Waitanjogy and Jiawei, 2010) Proposed a method to build an alert cluster that depends on support evidence and clustering technique. They computed the similarity of verified alarms by using distance among new alarm features. In order to compute new feature alert, it measures the distance between these alarms, for finding the new features. They needed more information to give them the support evidence; so that, they use vulnerability scanner. Also, they used Meta alert to measure the distance between the features. Fig. 4 show the Architecture of their alert cluster [25]. The new features selected are frequency, severity and relevance, each of these has a high or low degree. If two alerts have the same degree, the alarms have the same distance.

They applied this mechanism on DARPA 1999 on 3 days (Thursday in the fourth week and also Friday and Thursday in fifth week). Their approach focuses on finding the number of unnecessary alerts and improves quality of the alerts which directed to analyst. When they applied their method, they got the result 78 % as average for three days [25].

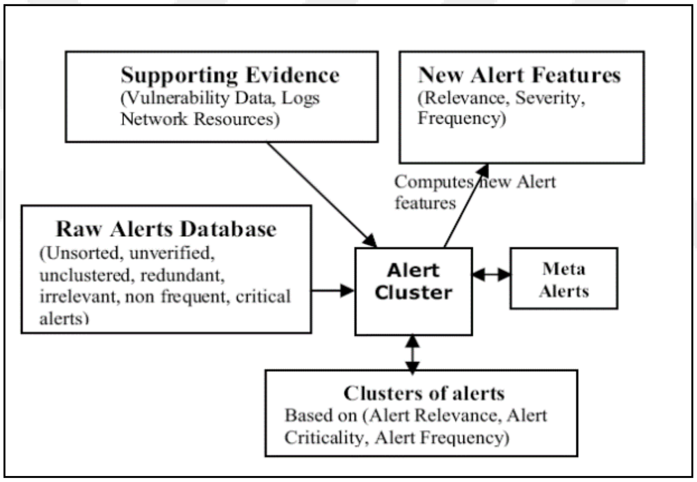


Figure 4. Architecture alert cluster [25]

(Perdisci and Giacinto and Roli, 2006) Proposed an alarm clustering system which depends on a new strategy where produced descriptions of attacks from alerts resulted from multiple IDS.

That system consisted of many modules such as AMI which is used to unite the formula of the alerts because the system uses three IDs. The second module is classification which labelled the alert message to attack class that depended on describing attacks. The last one is clustering /fusion module which it consisted of many sets. Each set related to particular attack class. Fig 5 shows the alert clustering system [26].

They used DARPA 1999 dataset on this system where they used 3IDS on 3 days (the same days which were tested in the previous study). They got result 64.6% as the average for the three days. The purpose of that system to summarize the attacks and find the number of false alerts [26].

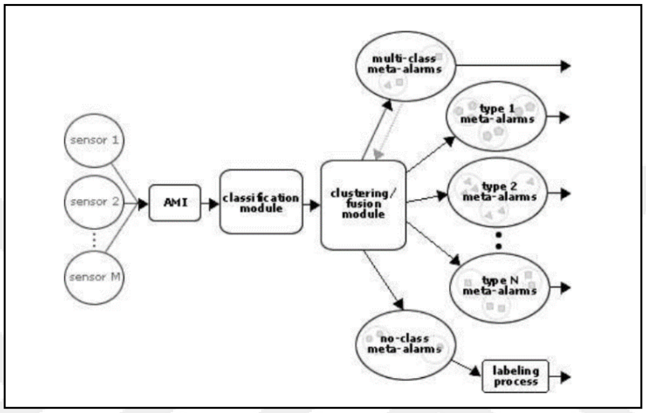


Figure 5. The alert clustering module [26]

1.5 Objectives

The basic goal of this study is to propose an enhanced data mining technique to use in our design for the proposed classification system. Our system classifies the alerts and provides a threat degree to each alarm. This study would be beneficial for the security analyst to distinguish the real attacks from the unreal attacks. The classification will reflect positively on network security and give the facilities to the analyst to achieve his work in the shortest time. This will increase the efficiency of network security and increase levels of accuracy.

1.6 Data Mining (D.M.)

D.M is a process of obtaining knowledge from the big database. But in the beginning, it is important to understand the available data and then predict new data [27].

Many people have used term *knowledge discovery from data* (KDD), which has many iterative sequences of steps. The first step is 'data cleaning' (removal of inconsistent data and noise), followed by 'data integration' (combining multiple data sources). The third level is 'data selection' (wherein data relating to the task of analysis is recovered from the Data Base (DB)). Next is the step of 'data transformation' where the data are consolidated into forms suitable for extraction by performing aggregation operations or summaries. The fifth step is 'D.M' (intelligent methods used according to extracted data styles). This is followed by' pattern evaluation' (identifying interesting styles which represent knowledge depending on some exciting measures). Finally, the 'knowledge is presented' (using techniques for knowledge representation which are used to display the obtained knowledge to the user) [28].

Many terms have little different or similar meanings to D.M, like knowledge extraction and knowledge mining from data. In addition, there are various D.M techniques like classification, which analyses training data (data have a class label) and creates models for each class depending on features of the data. Moreover, this analysis provides a good understanding of the big data and it helps decision makers to make decisions, such as whether a bank will give loans to customers. Clustering is a type of analysis that determines the clusters in massive data, where each cluster consists of a collection of data with similar features. Distance functions can express similarities and provide a high quality of resemblance or low degrees, such as the clustering of houses with respect to their geographical location and flood areas. Association rules discover the correlations or association relations among a set of items and these relations are expressed as rules depending on frequent items. Additionally, association rules are used in transaction data analysis in marketing [28].

With its appearance, the data mining software, which extracts important information from data, has become popular among educational institutions, telecommunication companies, banks and business organizations. Such information can help to facilitate decisions. Many organizations prefer to use data mining in order to find relationships in a large database and/or to discover patterns. Data mining plays an important role in finding patterns in data that highlight customer needs and purchase behaviour. This vital information helps organisations to develop their business performance and to enhance target sales, customer management and marketing [29].

In practice, there are two objectives of data mining: prediction and description [30]. *Prediction* means using fields or values in DB to predict new values of other features, such as values of classification techniques. *Description* distinguishes data properties in a database such that it is a process to find any patterns and relationships in a dataset. These patterns are found in forms rules such as cluster and association rules.

1.6.1 Data mining requirements

For the affectivity of data mining techniques, many data mining requirements should be compatible with proposed technique [31].

1- First requirement is scalability: which means that it has the Possibility to deal with the growing massive number of data, and this is very interesting in case of dealing with data in form alerts emitted from the IDS. The scalability is necessary to be provided.

2- The second requirement is multiple attribute types: the alarms of IDS have different data types like categorical attribute (IP address and ports) and text attribute (classification) and so on. Therefore, it is necessary to use DM technique to deal with the differences in data.

3- Another requirement which is easy to use: the proposed method must be simple to use and to get the correct results because the security analyst may not be specialized in data mining. Also, he needs to know a little about data mining.

4- Finally, the Noise tolerance**:** there are a lot of illegitimate alarms and noise came from logs file and it is ambiguous to be interpreted. Therefore, the DM technique should deal with illegitimate alerts.

In these days, there is a need to D.M techniques to detect and find intrusions because no one knows the amount of data that will get it and how we display it? Also, what the type of the data which they will see it in field intrusion detection? [32]. So, the researchers have used data mining technique because it deals with a large number of data and since the intrusion detection system that produce massive alerts. Therefore, data mining will be suitable to deal with a large number of alerts. Furthermore, the data mining technology has solved the problem of IDs about how to isolate the abnormal activities and the usual activities from a vast number of row attributes, and also how to create strong intrusion rules after the row network data have been collected [33].

The D.M. Applications are used to satisfy the needs of the professional person, researchers. The first people who applied D.M methods in the IDs fields were Savatore and Lee from Colombia University [33].

Data mining has many techniques: classification, clustering and association rules. This thesis highlights the association rules in the proposed classification system.

1.6.2 Association Rules (A.R)

Data mining is used in many fields like marketing, fraud detection, intrusion detection system and medicine etc. and a lot of algorithms is used in these fields, and one of the popular algorithms is association rule [30].

Association rule based on discovery association and correlations among items in the big dataset. The discovery of correlation relationship hidden among a lot of transaction records helps the officials in decision-making processes like in the field of marketing. For example, Data mining experts can see if the customers bought object R, how likely the customers also bought object S on the same trip to store. So, the rule suggests there is a strict relationship between the sale of R and sale of S because many customers bought the two products together [25, 26] the relationship is represented in association rule by R ⟹ S where R and S are groups of items [30].

In more details ,let $N=\{i_1,i_2,\ldots,i_N\}$ set of items ,$D =\{t_1,t_2, .., t_K)$ be set of all transactions where t is subset of items like t ⊆ N. Every t in D have transaction ID (sptial number) called (TID)), suppose that R group of items. Also, t will contain R if R ⊆ t, and A.R include the form R ⟹ S, where R ⊆ N, S ⊆ N and R ∩ S = ∅. The rule R ⟹ S holds within group D with support (sup) is the ratio of transactions containing both R and S in the whole set [28,36,37] In addition to, the confidence is a measure of certainty between the item sets.

Association rules (AR) have two steps:

i- Find every frequent item sets (FIS) is the basic step.

ii- The next step is to create reliable rules from the item sets (IS) that obtained them in the (i) step.

The term item set means a group of items. In case of item set contains 1 item, it will be called 1-itemsets, and if include M items, it will be called M-item sets. The number of times appearance of item set in each transaction in dataset called support value. While the frequent item set denotes to any item set have the support value is higher than or equal to the minimum sup (min.sup) that the user define it [38].

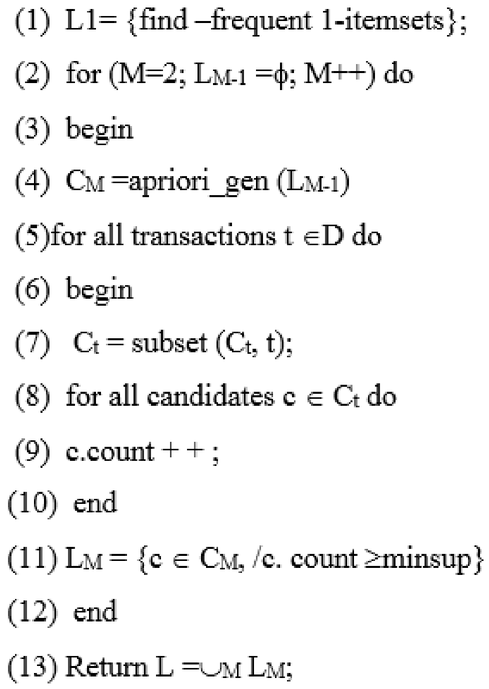
1.6.2.1 Apriori algorithm

Apriori is algorithm proposed by R. Srikant and R.Agrawal in 1994 for discovery of FIS .The fact of name apriori algorithm came from the prior knowledge of frequent itemset where it is used with an iterative approach called a level-wise search. To explain, M-itemset is used to generate (M+1)-itemset where level1 (L_1) which contains frequent first itemset. Then L1 is used to explore level2

(L_2) which includes frequent2 itemsets then (L2) is used to explore (L_3), and go on until it cannot find M-FIS. Exploration of each of L_M needs a full scan of database [28, 39,40]

To decrease the search space and improve work of the level wise generation of itemsets that match the condition min.sup. A significant property named the prior property is used (all nonempty subsets of FIS should also be repeated). To more details, if there is any itemset that is unfrequent which means it does not meet the min.sup. Moreover, its superset should not be generated [40].

Apriori (   )

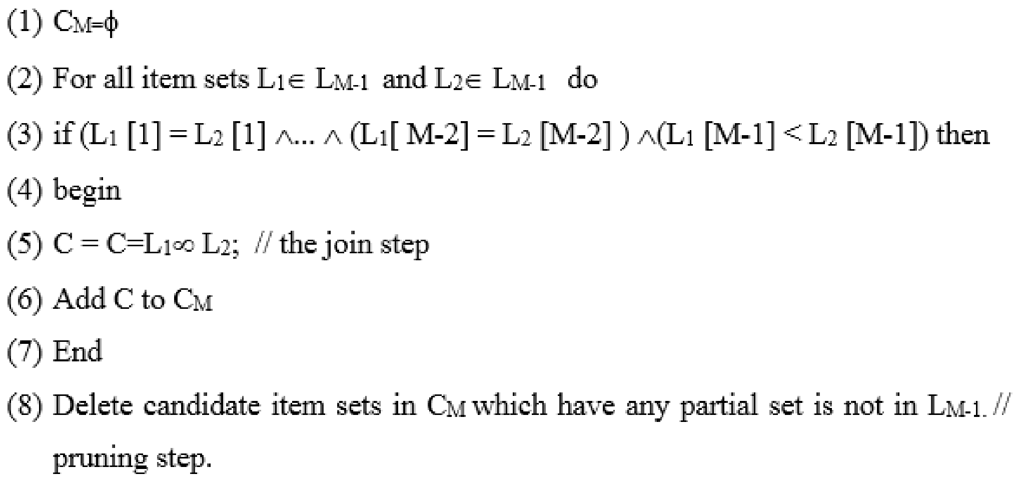


```
L1= {find –frequent 1-itemsets};
for (M=2; $L_{M-1}$ =$\phi$; M++) do
begin
    $C_M$ =apriori_gen ($L_{M-1}$)
    for all transactions t $\in$D do
    begin
        $C_t$ = subset ($C_t$, t);
        for all candidates c $\in$ $C_t$ do
            c.count + + ;
    end
    $L_M$ = {c $\in$ $C_M$, /c. count $\geq$minsup}
end
Return L =$\cup_M$ $L_M$;
```

According to steps of the Apriori algorithm [28], the first phase of the algorithm in the DB is calculating the number of appearance of each item and finding the most frequent item set. Each phase will generate the C_M candidate that contains sets of the items from Level (M-1) L_{M-1} by using apriori-gen. After that, the algorithm scans the DB to count the support of every candidate in $C_{M.}$

The function (Apriori-gen) takes the variable L_{M-1}, and the level of frequent (M-1) item sets. And returns a superset of all M FIS.

Priori-gen (L_{M-1});



```
$C_M$=ϕ
For all item sets $L_1 \in L_{M-1}$ and $L_2 \in L_{M-1}$ do
if ($L_1$ [1] = $L_2$ [1] ∧... ∧ ($L_1$[ M-2] = $L_2$ [M-2] ) ∧($L_1$ [M-1] < $L_2$ [M-1]) then
begin
C = C=$L_1 \infty L_2$; // the join step
Add C to $C_M$
End
Delete candidate item sets in $C_M$ which have any partial set is not in $L_{M-1}$. //
pruning step.
```

Pruning step is applied in two stages, Join stage: in this stage, $L_{M-1}$is joined with $L_{M-1}$to find L_M, a set of the candidate(C) of 'M-item sets' that is created by joining (L_{M-1}) with itself. This set of (C) is refer as C_M. Let considered (L_1) and (L_2) be item sets in (L_{M-1}). The notation (L_i [j]) denoted to the (j^{th}) item in (L_i) (e.g., (L_1 [M-2]) means the second of the last item in (L_1)) [28].

By the convention, apriori assumes that items within a transaction are stored in lexicographical. The join ($L_{M-1} \infty L_{M-1}$) is performed, where members of (L_{M-1}) are joinable of their first (M-2) items that are in common (The new item set contains the items in those two large item sets in order). Thus after the joining step, C_M becomes a subset of $C_M \supseteq L_M$ .That neans, members (L1) and (L2) of (L_{M-1}) are joined if (L_1 [l] = $L_{.2.}$ [1] ^ ($L_{.1}$] [2] = $L_{2.}$[2] ^... ^($L_{.1}$ [M-2] = $L_{2.}$ [M-2]) ^($L_{.1}$ [M-1] < $L_{2.}$ [M-1]) .The condition $L_{1.}$ [M-1] < $L_{2.}$ [M-1] give ensures that no duplicates are created. The resulting item set is appeared by joining (L_1) and (L_2) contains $L_{.1}$ [1] $L_{.1}$ [2] ... L_1 [k-1] L_2 [k-l].

Prune step is the second step: any candidate C_M generated from level L_{M-1} that contains item sets may be frequent or not. So, the database is scanned to determine super value of each item set in candidate C_M to determine L_M. Moreover, there is a condition that all item sets in candidates must be frequent (supper value more than or equal to min.sup), and in case is checked each candidate in each time that led to massive computation and huge steps. Therefore, apriori property is used to reduce size the C_M where it checks each (M-1) item sets if is not frequent. In case of it will not be subset of frequent M-item sets and remove from candidate C_M, it will not put in level (M). For example, (AB is 2 subset of 3-itemset ABD but AB or AD or BE not in level 2 so ABD will remove from C_3) the steps below illustrate pruning 's work [38].

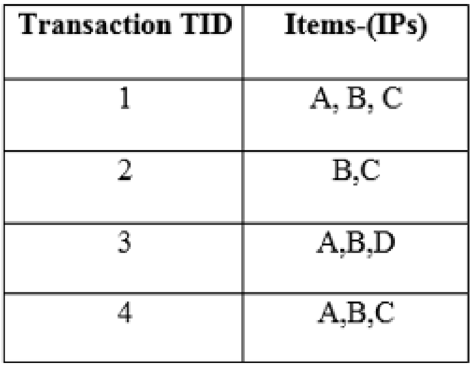
```
For all candidates $c \in C_M$
For all (M-1)-item sets $d \subseteq c$ and $c \in C_M$ do
If $d \notin L_{M-1}$ then
 Delete c from $C_M$
end if
end for
end for
```

Example on apriori algorithm [27], the items abbreviations of the database has been given in Table 1, and the transaction database has been explained in Table 2.

Table 1. Items abbreviations of database [27]

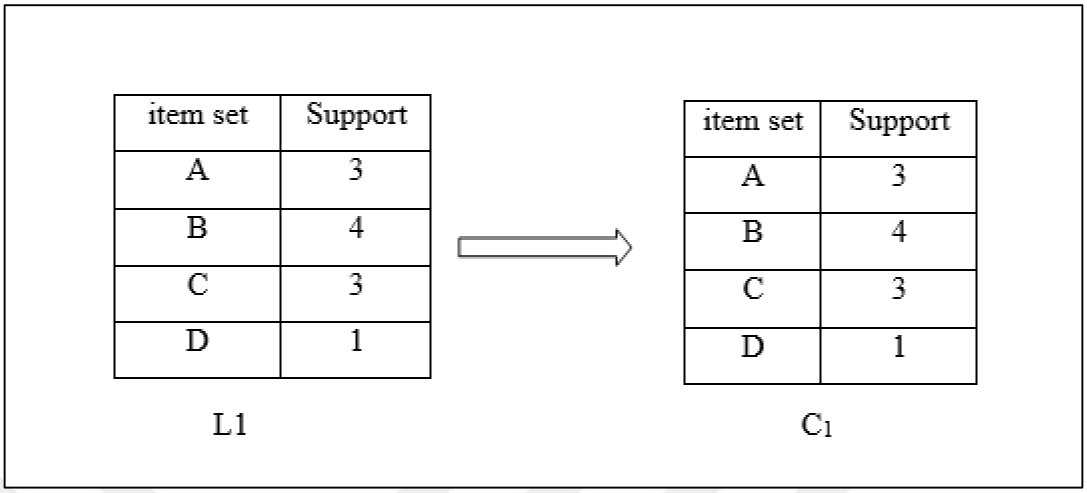
Item	IPs Names
A	IP1
B	IP2
C	IP3
D	IP4

Table 2. Transaction Database [27]



Transaction TID	Items-(IPs)
1	A, B, C
2	B,C
3	A,B,D
4	A,B,C

The algorithm scans all transactions to calculate the number of each item which appears in each transaction. To generate 1-itemset in the first candidate C1, it Compares each support for 1-itemset with min.sup =1 and removes each itemset not frequent. Here there does not remove any itemsets because each itemset is frequent. The next step is to generate $L_{1.}$ Fig 6 explain generating level 1.



item set	Support
A	3
B	4
C	3
D	1

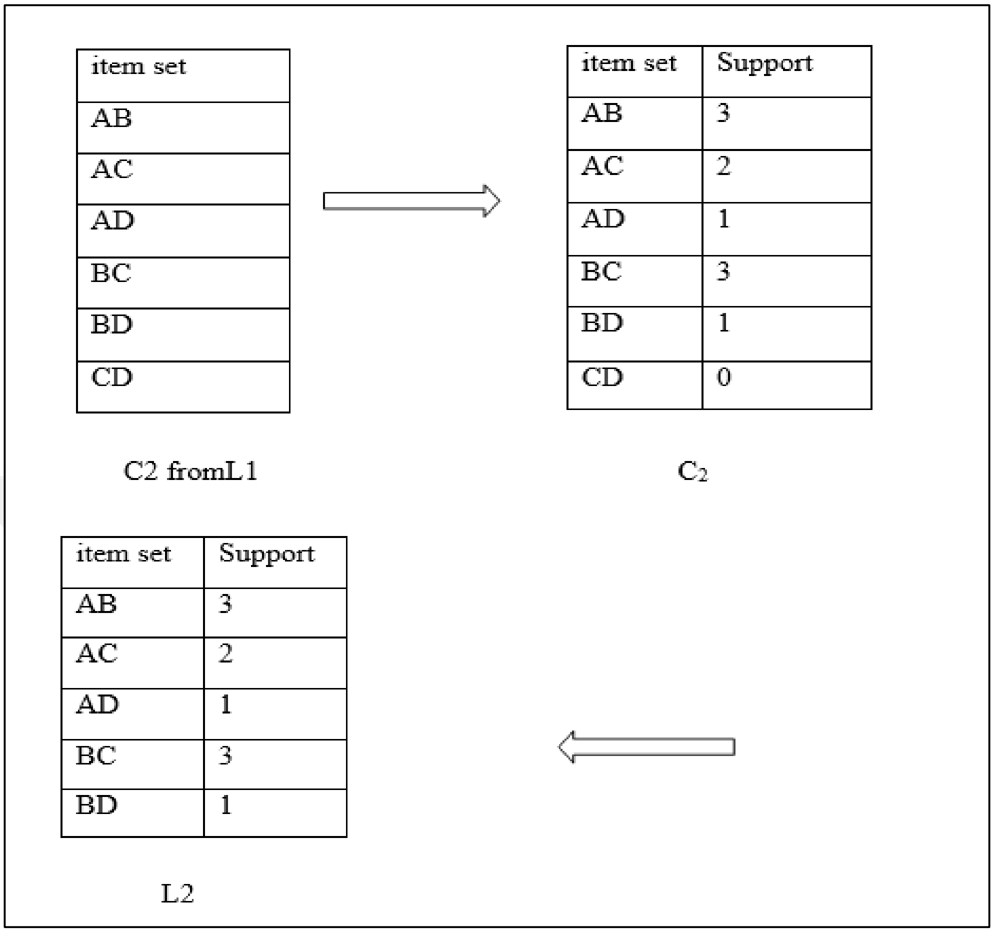
L1

item set	Support
A	3
B	4
C	3
D	1

C_1

Figure 6. Generating level 1 [27]

L1 is joined with itself to generate the candidate C2 that contains 2-item set, and the transaction in database is scanned and calculated the sup. For each 2-itemsets in C_2. Again, the algorithm compares each support for 2-itemset in C_2 with min-sup and removes each itemset that is not frequent. To generate $L_{2,}$ the CD is removed because it is not frequent. Fig 7 explain generating level2.



item set
AB
AC
AD
BC
BD
CD

C2 fromL1

item set	Support
AB	3
AC	2
AD	1
BC	3
BD	1
CD	0

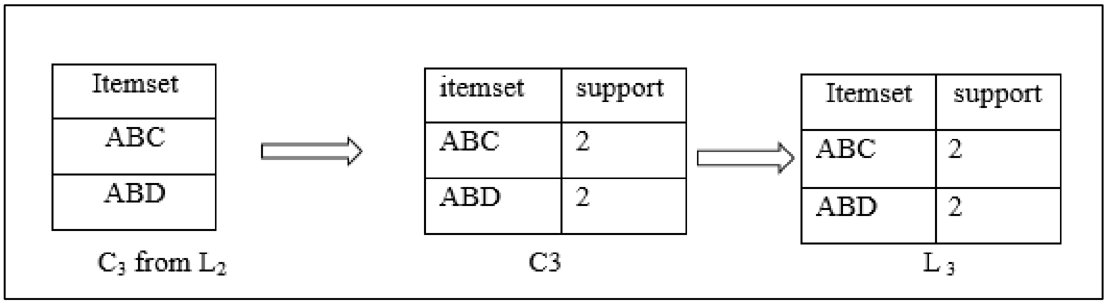
C_2

item set	Support
AB	3
AC	2
AD	1
BC	3
BD	1

L2

Figure 7. Generating level 2 [27]

To generate 3-itemsets. The Apriori algorithm uses joining L_2 with itself to generate candidate C_3 which contains 3-itemsets with the condition the first item in 2-itemsets is similar; however, it is different in the last one. For example, AB is not joined with BC; but, is joined with AC to generate ABC and joined with AD to generate ACD. Firstly from join steps, C_3 = {ABC, ABD, ACD, BCD} is achieved. But, according to apriori property which requires all subsets of item sets be frequent; otherwise, the item sets are removed. Therefore, the algorithm removes these item sets ACD, BCD from C_3. Again transactions in the database are scanned, and support is calculated for each candidate item sets in C_3.The algorithm Compares each support for 3-item set in candidate C_3 with min.sup and removes each 3-itemset that are not frequent after that L_3 is generated. Fig 8 shows generating level3.



Itemset
ABC
ABD

C_3 from L_2

itemset	support
ABC	2
ABD	2

C3

Itemset	support
ABC	2
ABD	2

L_3

Figure 8. Generating level 3 [27]

To generate 4-itemsets, apriori algorithm joins L_3 with itself to generate candidate C_4 where C_4 contains 4-itemsets. From joining step, it must be C_4= {ABCD}. But according to apriori property, which need all sub group of 4-itemsets must be frequent or the itemsets is removed. Therefore, the algorithm, in this case, removed the 4-itemsets ABCD from C_4 because the subsets ACD and BCD are not in L3.

1.6.2.2 Rules Generating

When the FIS are found in transactions, it is easy to create the powerful rules from these sets (the Rules that meet minimum confidence threshold (min-conf), and minimum support threshold (min-sup) are called Powerful rules) [38]. The strong rule can be achieved by using equation (1).

$$\text{Confidence } (R \dashrightarrow S) = P(R|S) = \left(\frac{\text{support } (R \cup S)}{\text{support } (R)}\right) \qquad (1)$$

Support (R ∪ S) means the number of transactions contain the (R ∪ S) item sets . While support R refers to the number of many transactions include the R item set.

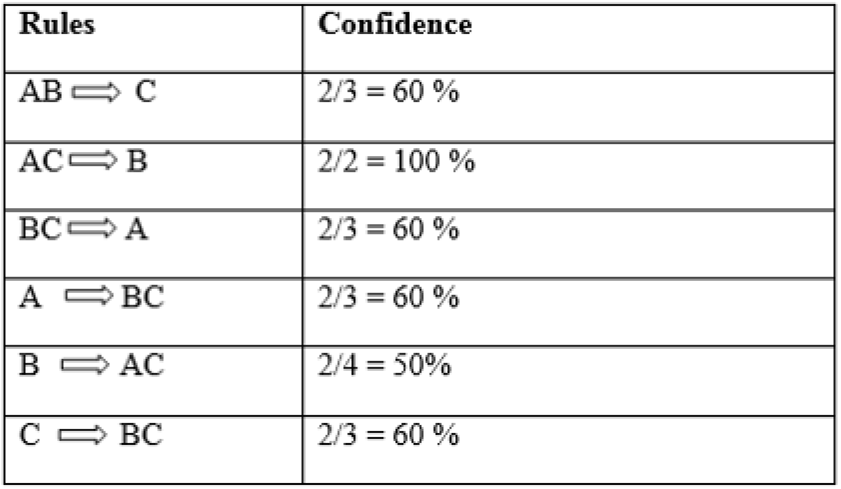
Depending on the equation (1), the rules can be created as following [28]:

- For every frequent item set I, generate all not empty subset of I.

- For each partial sets not empty P of I, the output of the rule is {P⇨ (I-P)} in case of $\left(\frac{\text{sup (I)}}{\text{sup (P)}}\right)$ $\geq$ minimum confidence.

For example, the frequent item set is I = {ABC} , the partial sets of I are [AB] , [AC] , [BC] , [A], [B] ,[C] and the results are given in Tables 3.

Table 3. Generating rules [27]



Rules	Confidence
AB ⇨ C	2/3 = 60 %
AC⇨ B	2/2 = 100 %
BC⇨ A	2/3 = 60 %
A ⇨ BC	2/3 = 60 %
B ⇨ AC	2/4 = 50%
C ⇨ BC	2/3 = 60 %

According to the Table (1.3), and in case of min-conf = 60 % which is defined by the user. The five rules consider the strong rules because of its confidence larger than or equal min-conf. But the fifth rule (B ⇨ AC) is not reliable rule because of its confidence less than 60.So, it is cancelled.

2. METHODS

2.1 Introduction

This chapter introduces the proposed framework for this thesis namely classification system IDs by using data mining. The proposed system aims to classify alerts. The networks have been widely used in various fields. Although the benefits they have brought by them, they also bring the dangers. For example, a lot of kinds of malicious programs which affect the work of networks when transmitting data. So, working researchers to develop techniques that reduce this risks.

Intrusion detection systems are discovered to give the protection to networks and computers systems. These techniques send alerts, which help the analysts to check each alert.

2.2 Methodology

The proposed system contains of two modules. The preprocessing module is the first module where the best results can be obtained by removing any unnecessary features and duplicate alerts. After application of the first module, the remaining alarms will be entered into the second module, which is the classification module. The architecture of the classification system is presented in Fig 9. This proposed system is illustrated in detail in the following sections.

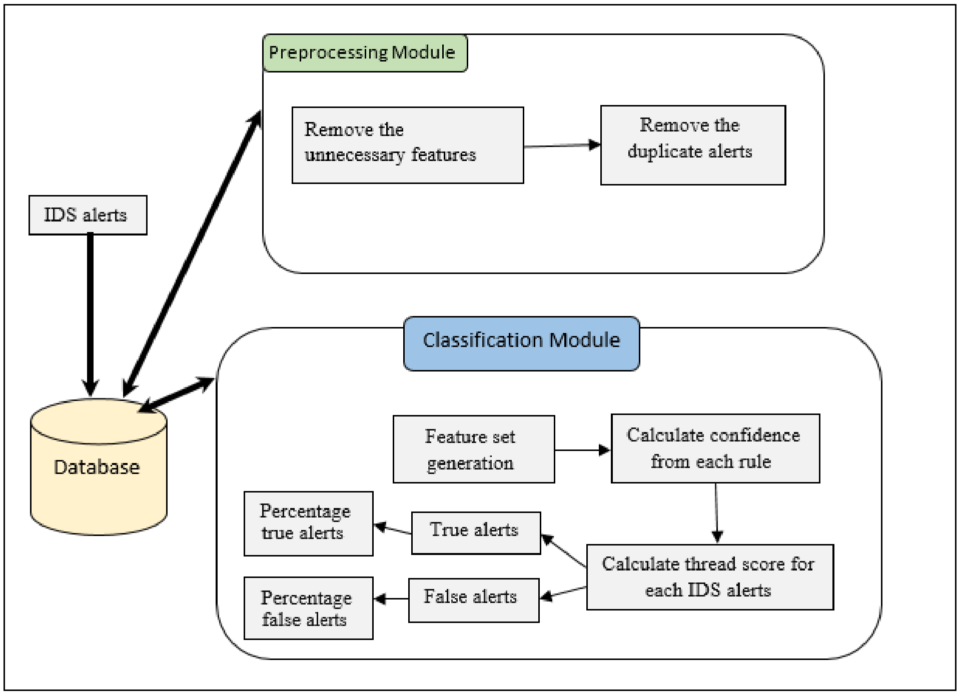


Figure 9. The proposed classification system architecture "classification system IDs by using data mining"

2.2.1 Preprocessing module

The preprocessing module has two components: one to remove unnecessary features and the other to remove duplicated alerts. The preprocessing module is explained in Fig 10.

A. Removing unnecessary features: In this component, unnecessary features which are not used by all researchers are removed. This is done manually. The total number of alerts is equal to 23 features , and the features that have been used are:

- IP destination address (IPd) .
- IP source address (IPs).
- Port destination address (Pd).
- Port source address (Ps).

- Classification feature as a transaction.
- Priority.

Where IPd, IPs, Pd, Ps, and classification are used to generate the featuresets, while priority is used to calculate the thread score IDs alerts.

B. Removing duplicated alerts: In this component, all duplicated alerts are removed. If the all features have the same data, it will remove the duplicate alerts and remain in one row (alerts). Table 4 Example to explain duplicate alerts. while Table 5 shows the remaining alerts after removing the duplicate alerts from Table 1.

Table 4. The duplicate alerts .

IPs	**IPd**	**Ps**	**Pd**	**TID**	**priority**
IPs1	IPd2	Ps1	Pd2	X	P2
IPs1	IPd2	Ps1	Pd2	X	P2
IPs2	IPd2	Ps3	Pd1	X	P1
IPs2	IPd2	Ps2	Pd1	Y	P1

Table 5. The rest alerts after removing the duplicate alerts

IPs	**IPd**	**Ps**	**Pd**	**TID**	**priority**
IPs1	IPd2	Ps1	Pd2	X	P2
IPs2	IPd2	Ps3	Pd1	X	P1
IPs2	IPd2	Ps2	Pd1	Y	P1

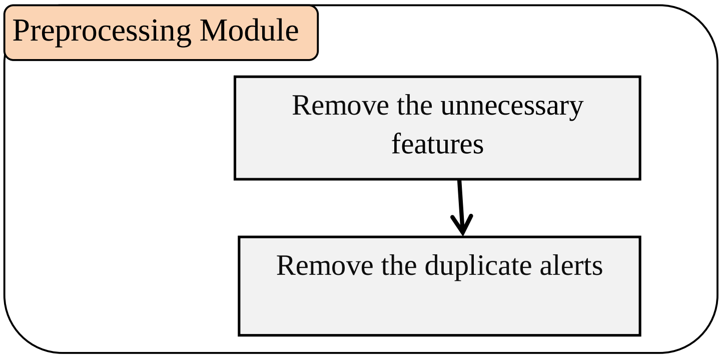


Figure 10. Preprocessing module

2.2.2 Classification module

The classification module contains several components with various functions that lead to classifying the alarms. The details for each component are explained in the following sections. The elements of the classification module are explained in Fig 11.

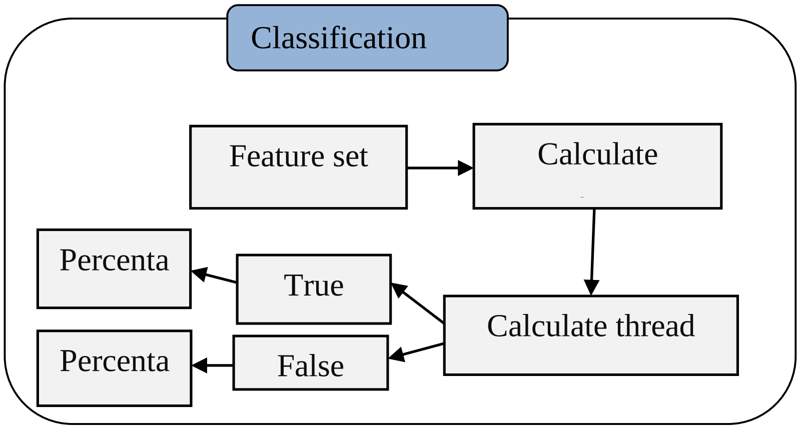


Figure 11. Classification module

2.2.2.1 Features set generation component

The features set generation component is depend on enhancing the Apriori algorithm, and the main function of this component is generating the features set.

This is steps the enhancement of Aproiori Algorithm:

1- Each selected feature is stored with TID separately.

2-Calculate the number of the TIDs for each selected feature, the number of TID is considered support value.

3-Join each (M-1) featureset with itself to generate M-featureset with condition intersect their common TIDs and another condition is to similar each first features in (M-1) featureset and differ in the last one.

4- Check each support value with min-sup (which is value the user defined it) if the support is greater than or equal to min-sup that mean the featureset is frequent and will used to generate the next featureset else it will remove.

Example of Enhanced Apriori algorithm with details of how it work:

First of all, each TID is stored separately with the selected feature and the process is performed with the selected feature serially. Table 6 example to explain the connection IPs (Feature 1) with classification (TID); Table 7 example to explain the connection IPd (Feature 2) with classification (TID); Table 8 example explain the connection port.s (Feature 3) with classification (TID); and Table 9 example to explain the connection IPd (Feature 4) with classification .

Table 6. IP.s (Feature 1) connects with classification (TID)

Feature (IP.s)	TID (classification)
IPs1	Misc activity
IPs2	Attempted Information Leak
IPs3	Misc activity
IPs4	Attempted User Privilege Gain

Table 7. IP.d (Feature 2) connects with classification (TID)

Feature (IP.d)	TID(classification)
IPd1	Attempted Information Leak
IPd2	Misc activity
IPd3	Misc activity
IPd4	Attempted User Privilege Gain

Table 8. port.s (Feature 3) connects with classification (TID)

Feature (port.s)	TID (classification)
Ps1	Attempted Information Leak
Ps2	Potentially Bad Traffic
Ps3	Attempted User Privilege Gain
Ps4	Misc activity

Table 9. Port.d Feature (4) connects with classification (TID)

Feature (port.d)	**TID(classification)**
Pd1	Attempted Information Leak
Pd2	Potentially Bad Traffic
Pd3	Attempted User Privilege Gain
Pd4	Misc activity

Table 10 example to illustrate TID abbreviations of the database. In this step, a 1-featureset is generated in the process of configuring a new database containing the three main columns: Feature, TIDs, and the last column is a support which refers to the frequency of TIDs in each feature. This step prevents repeated scanning of the transaction database. The value of minimum support is specified by the user and called min.sup.

The IP source (IPs) feature is selected in Table 11 to explain the technique of the improved Apriori algorithm and generate a rules algorithm with the value of min.sup equal to 1.

Table 10. The TID abbreviations of database

TID	**TID name**
X	Misc activity
Y	Attempted Information Leak
Z	Potentially Bad Traffic
W	Attempted User Privilege Gain

Table 11. 1-featureset

Featureset (IPs)	TID (classification)	Support
IPs1	X, Y, Z	3
IP2	Y, W	2
IPs3	X, Z	3
IPs4	X, Y, Z, W	4

To explain, the support for IPs1 is equal to 3 because IPs1 appeared in the TIDs (X, Y, and Z). Moreover, the support for IPs2 is equal to 2 because IPs2 appeared in the TIDs (Y and W). IPs3 appeared in the TIDs (X, Z), and IPs4 appeared in the TIDs (X, Y, Z, and W).

The generation of M-Feature sets is based on the following two steps: the basic step is to join the two (M-1) feature sets which are similar to the first feature and vary it in the last one. The second step is to intersect their common TIDs in cases of any frequent featuresets (its support is higher than or equal to the min.sup.), it will be used to generate the next featuresets. On the contrary, it removes and does not enter in the generation process. Therefore, there is no need to scan each subset in (M-1) featuresets to obtain L_M, as in the Apriori algorithm. The input of this step is a 1-Featureset and the min.sup threshold, which is here equal to 1. The output from this step is 2-Featuresets, which are stored in a large feature table. 2-Featuresets are illustrated in Table 12.

Table12. 2-Featuresets

Featureset (IP.s)	**TID (classification)**	**Suppose**
IPs1, IPs2	Y	1
IPs1, IPs3	X,Z	2
IPs1, IPs4	X, Y, Z	3
IPs2, IPs4	Y,W	2
IPs3, IPs4	X, Z	2

Table 12 refers to the joining between features (items) IPs1 with feature (item) IPs2. Moreover, feature IPs1 is joined with feature IPs3 and so on for all frequent 1- features (its support value is higher than or equal to min.sup) until all 2-featuresets (2-itemsets) are generated. Furthermore, it must check the condition as to whether the two 1-features are joined with themselves and have some TIDs in common. In this example, feature IPs2 is not joined with IPs3 because they do not have TIDs in common. IP2 has TIDs (Y, W) and IP3 has IDs (X, Z). Therefore, IP2 and IP3 are ignored. On the other hand, IPs1 and IPs2 have TID (Y) in common. In addition, IPs1 and IPs3 have TIDs (X, Z) in common, IPs1 and IP4 have TIDs (X, Y, Z) in common, IPs2 and IPs4 have TIDs (Y, W) in common, and IP3 and IP4 have TIDs (X, Z) in common.

From generating the 2-Featuresets, the 3-Featuresets are generated where the 2-featuresets are taken and joined with each other. An example of how a 3-Featureset is obtained from a 2-Featureset is illustrated in Table 13.

Table 13. 3-Featuresets

Featureset (IP.s)	**TID (classification)**	**support**
IPs1, IPs2, IPs4	Y	1
IPs1, IPs3, IPs4	X,Z	2

In this table, because the 2-Featuresets (IPs1, IPs2) and 2-Featuresets (IPs1, IPs4) are common in TID (Y) and they are similar in the first feature and vary in the last one, the 3-Featuresets (IPs1, IPs2, and IPs4) are generated by combining them. Furthermore, the 2-Featuresets (IPs1, IPs3) and 2-Featuresets (IPs1, IPs4) have TIDs (X,Z) in common and they are similar in the first feature and vary in the last one. Therefore, the 3-Featuresets (IPs1, IPs2, and IPs4) are generated by combining them.

The 2-featuresets (IPs1, IPs2) and 2-featuresets (IPs1, IPs3) do not have some TIDs in common; therefore, the 3-Featursets (IPs1, IPs2, IPs3) are ignored. According to the first step in this technique, we join the (M-1) -featuresets with each other, which are similar to the first feature and vary in the last feature. The 2-Featuresets (IPs2, IPs4) are ignored because there is no similarity in the first feature, which is the same reason for the 2-Featursets (IPs3, IPs4).

To generate 4-Featuresets, the 3-Featursets are joined with each other. However, here the second feature in each 2-featureset is not equal. For example, (IPs1, **IPs2**, IPs4) joined to (IPs1, **IPs3**, IPs4) because **IPs2** does not equal **IPs3** and this condition makes joining between the Featuresets. Moreover, there are no TIDs in common between the Featuresets, so the 4-Featureset is empty.

2.2.2.2 Calculation of confidence for each rule component

This component is based on the featuresets as inputs to this component in order to generate rules and to calculate the confidence for each alert according to the rule generating algorithm.

The rules that are generated from each featureset are created where Eq.(1) is used to extract the confidence values.

For example:

Confidence (IPs1 → IPs2) = support (IPs1 ∪ IPs2) / support (IPs) = 1/3 × 100 = 33%.

Therefore, the confidence between the two features was not strong.

Support (IPs1 ∪ IPs2) refers to the number of common transactions in each IPs1 and IPs2 in the database. Support (IPs1) refers to the number of transactions in IPs1 in the database.

In this component, the threat score of the feature is calculated by using Eq. (2), which was proposed for this purpose.

$$\mathrm{TF} = \left(\frac{\sum_{\mathrm{i}=1}^{\mathrm{F}} \mathrm{confd(A)}}{F}\right) \tag{2}$$

Where

TF refers to the threat score feature;

Confd., refers to the convergence degree (confidence degree) that is obtained from Eq. (1); and

F, refers to the frequent number of features.

2.2.2.3 Calculation of threat score for each IDS alert

This component depends on the threat score of the average of the four features and values of the priority feature, where each feature is calculated separately. Eq. (3) is proposed to compute the threat score for each IDS alerts in the classification module.

$$Tv = (TF1+TF2+TF3+TF4/4) + PV) /2 \qquad (3)$$

Where

TF1, TF2, TF3, TF4 refer to the threat score of the four features;

PV refers to the value of priority value; and

Tv refers to the threat score of the IDs alert.

Feature priority is used because it has a vital role in determining the level of threat of IDs alerts.

2.2.2.4 True Alerts

In this component, the threat score of the IDs alerts is classified by identifying the threshold value (Tv), which is equal to 50. In cases of the Tv score exceeding 50, the alarm will be a real alert. Another task for this component is to collect the number of true alarms.

The threshold value which is selected to classify the alerts was equal to 50 because it gave the best results when used in true and false components, and another values like 30, 70 are tested for this reason and they did not give the satisfying results.

2.2.2.4.1 False alerts

In this component, the threat score of the IDs alerts is classified by identifying the threshold value, which is equal to 50. If the score of the Tv is less than 50, the alert will be designated a false alert. This means that no attack has occurred and it starts collecting the number of false alarms.

2.2.2.5 Percentage of true alerts

The task of this component is to take the final number of real alerts, which is collected from the previous component and divide it into the total number of alerts (No.alerts) to calculate (PTA).

$$\text{PTA} = \left(\frac{\text{number true alerts}}{\text{total number of alerts}}\right) \tag{4}$$

Example 1: In case the total number of alerts is equal to (1063) alerts and the number of the true alerts (which is computed from the true alerts component) is equal to 113, the percentage of true alerts according to Eq. (4) will be equal to 113/1063, which is 10.6%.

2.2.2.5.1 Percentage false alerts

In this component, the final number of calculated false alarms is divided by the total number of alerts (No.alerts) to calculate (PFA).

$$\text{PFA} = \left(\frac{\text{number false alerts}}{\text{total number of alerts}}\right) \tag{5}$$

Example 2: In case the total number of alerts is equal to (1063) alerts and the number of the false alerts (obtained from the false alerts component) is equal to 113, the percentage of false alerts (PFA) according to Equation (5) will become 950/1063, which is 89.3%. Fig 12 shows the details the classification according to the threshold value.

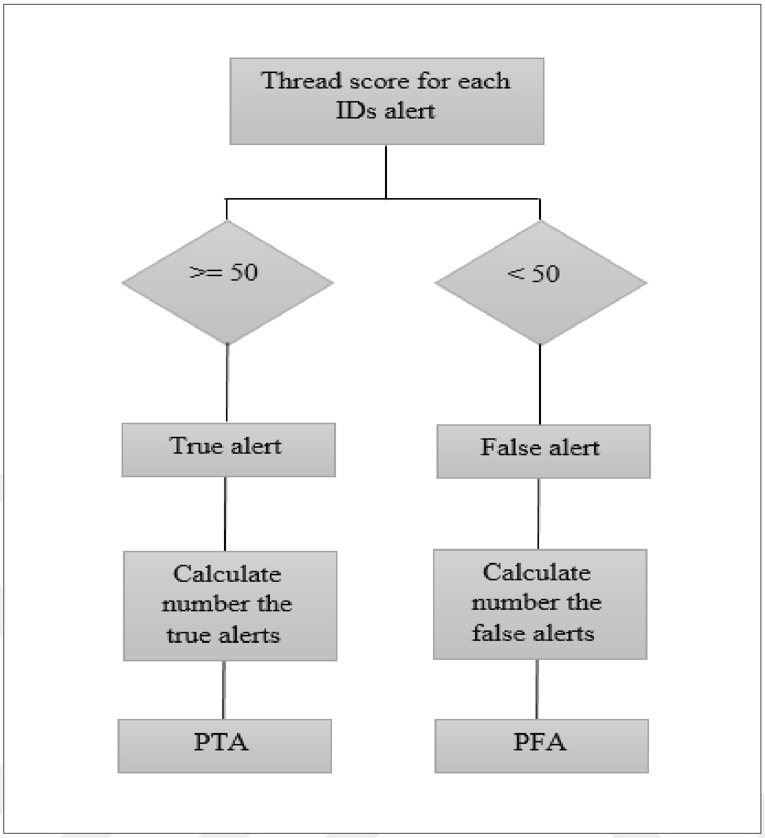

Thread score for each IDs alert
>= 50
< 50
True alert
False alert
Calculate number the true alerts
Calculate number the false alerts
PTA
PFA

Figure 12. The classification according to the threshold value

3. RESULTS AND DISSCUSIONS

3.1 Introduction

This chapter concentrates on discussing the results obtained from the proposed classification system. After the threat score IDs alerts (Tv) are acquired from the third component, the Tv’s are classified into true and false alerts by using the false and true component. In this component, the threshold value is determined by the user and selected as 50. If the value of Tv. is equal to or higher than 50, the alert is considered a true alert (attack). However, if the Tv is less than 50, the alert is considered a false alert (i.e., not an attack). Therefore, it becomes easy to calculate the percentage of each by calculating the percentage of the true alert components in order to calculate the PTA as well as the percentage of false alerts component in order to calculate the PFA. Percentage of false alerts (PFA) refers to the percentage of unnecessary alerts or false positives.

3.2 Datasets

To check the validity of the classification system, the proposed system is evaluated based on the Darpa 1999 data set.

DARPA 1999 dataset is generated by MIT Lincoln group to produce a 'DARPA' _sponsored Comparison of various intrusion detection system [41]. Dataset Darpa 1999 is considered as a reference in appreciating the performance of IDS. Moreover, the dataset is extracted from a simulation of a military management network for 5 weeks of its network traffic traces which have of different attack and common user activities.

3.3 Hardware and Software specifications

This system is programmed by using C# language and according to test the performance of the designed system, these hardware specifications were utilized:

The CPU for the computer was CPU Intel ® Core™ i3-32174 @ 1.8 GHz. Also, the size of the memory equal to 4.00GB, Finally the Hard Drive was 500 GB Hitachi HDS721616PLA380.

The following software specifications were utilized to check the performance of the proposed system as follow:

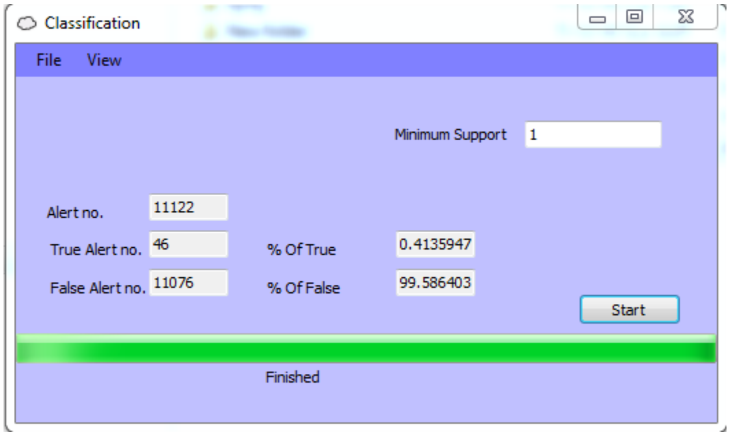
The operating system which is used to test the classification system is windows 7 professional and the Compiler is C# language, while the data set which is used to test the system effectively is Darpa 1999 data set which work on Microsoft office 2013

3.4 Implement the classification system on dataset

The proposed classification system has been tested based on the same number of weeks on which the researchers depended in previous works. They used the second, fourth and fifth weeks to test their system. Moreover, the proposed system was tested by conducting the three experiments on it. The first experiment was conducted in the second week. The second experiment was conducted in the fourth week and the third experiment in the fifth week.

3.4.1 The First Experiment

The first experiment was performed in five days from the second week of the Darpa 1999 data set (Friday, Monday, Thursday, Tuesday and Wednesday).



Minimum Support 1

Alert no. 11122

True Alert no. 46

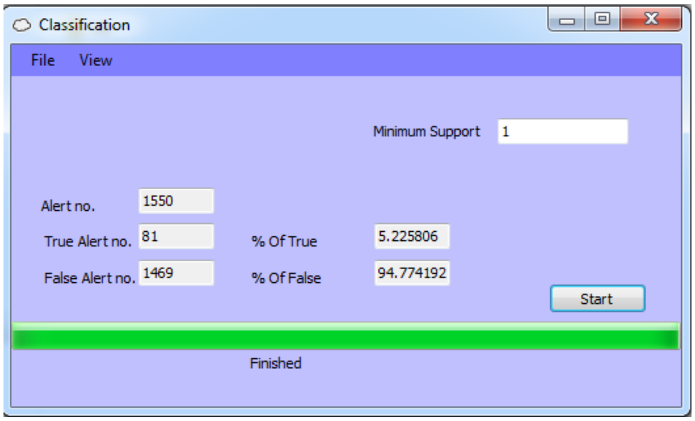
False Alert no. 11076

% Of True 0.4135947

% Of False 99.586403

Finished

Figure 13. Second week - Friday



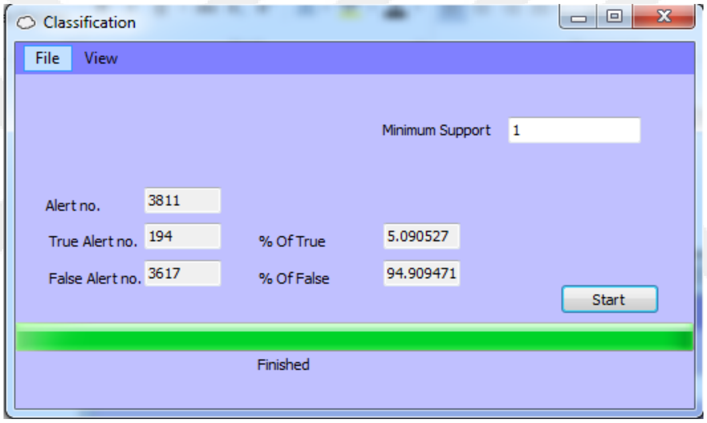
Minimum Support 1

Alert no. 1550

True Alert no. 81 % Of True 5.225806

False Alert no. 1469 % Of False 94.774192

Figure 14. Second week - Monday



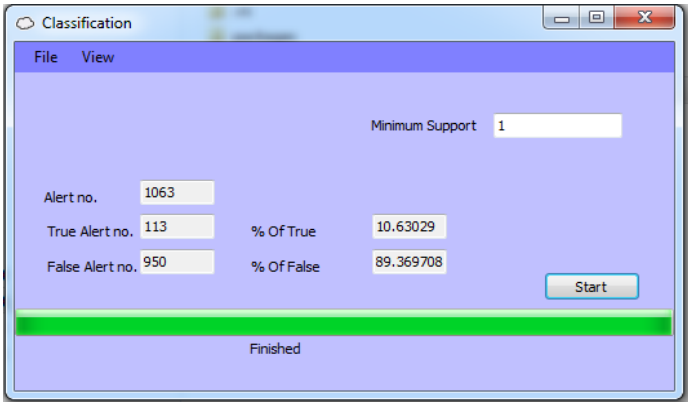
Minimum Support 1

Alert no. 3811

True Alert no. 194 % Of True 5.090527

False Alert no. 3617 % Of False 94.909471

Figure 15. Second week –Thursday



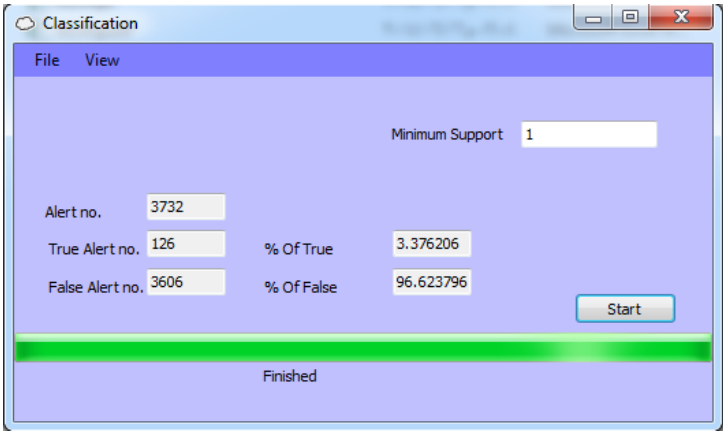
Minimum Support 1

Alert no. 1063

True Alert no. 113 % Of True 10.63029

False Alert no. 950 % Of False 89.369708

Figure 16. Second week-Tuesday



Minimum Support 1

Alert no. 3732

True Alert no. 126 % Of True 3.376206

False Alert no. 3606 % Of False 96.623796

Finished

Figure 17. Second week -Wednesday

Table 14. Results of the first experiment

Day	Total alerts before preprocessing	No. alerts	True Alerts no.	False Alerts no.	% of true	% of false
Friday	11122	11122	46	11076	0.41 %	99.58 %
Monday	1550	1550	81	1469	5.22 %	94.77 %
Thursday	3811	3811	194	3617	5.09 %	94.9 %
Tuesday	1063	1062	113	950	10.63%	89.36%
Wednesday	3732	3732	126	3606	3.37%	96.62 %

Table 14 demonstrates the results of the first experiment according to the implementation of the proposed system on the second week.

The symbols in Table (3.1) mean:

No. alerts in the table means the total number of alarms from an IDS in each day after preprocessing modules, True alerts no. means the number of real alerts (attacks) . While false alert no. means the number of false alerts, % of true means the percentage of true alerts .Finally, % of false means the percentage of false alerts.

According to Table 14, it became clear how the total number of alerts was classified into true alerts (attacks) and false alerts (not attacks). Moreover, the percentage of correct and wrong alerts were obtained where the average of the percentage of false alarms (PFA) was equal to 95% ((99.5 + 94.7 + 94.9 + 89.3 + 96.6)/5). This means that 95% of the total alerts in five days were unreal alerts while the average of the percentage of the true alerts was equal to 5%. Fig 15 shows the contradiction between the percentage of false alerts (PFA) and the percentage of true alerts (PTA) for the second week.

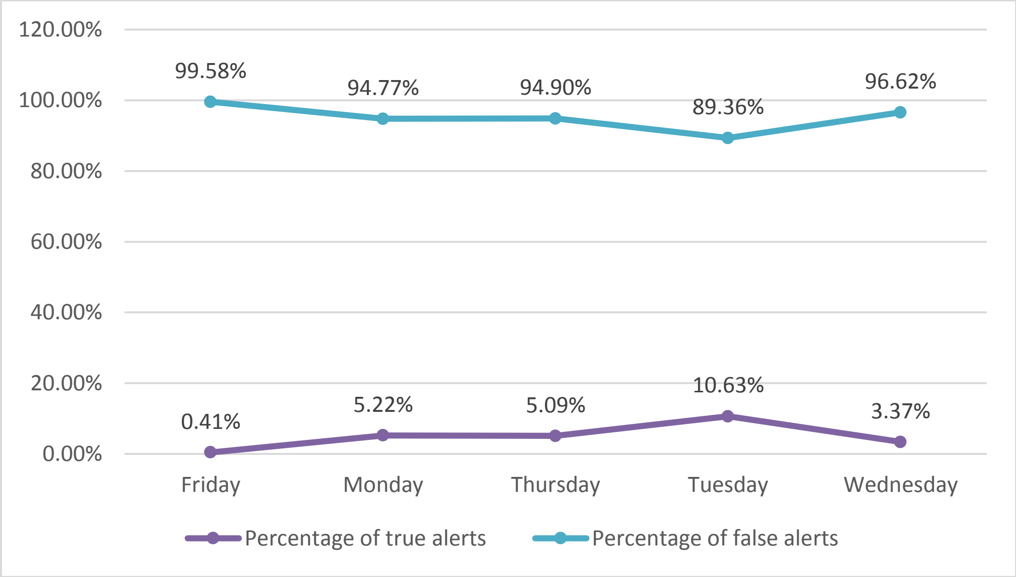
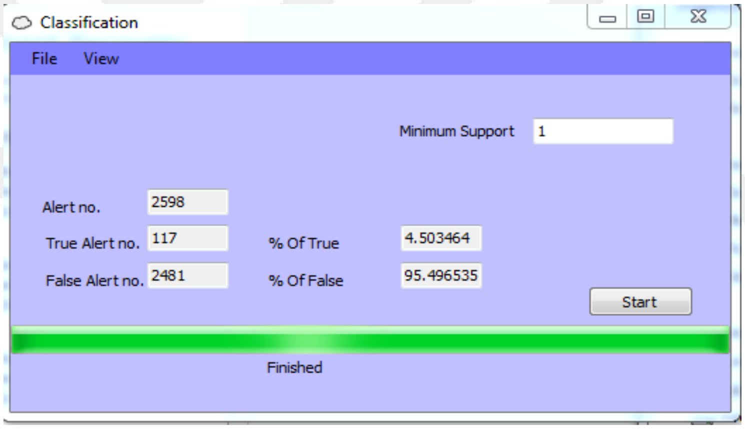


Figure 18. The contradiction between the PFA and PTA

According to Fig 18 on Tuesday, the percentage of the true alerts was the highest rate at 10.60% when compared with the other days, which means that it had the maximum number of attacks. Moreover, the percentage of false alerts was the lowest value at 89.30% when compared to the other days. In contrast, the percentage of true alerts on Friday was the lowest rate at 0.40%. This refers to the fact that it had the minimum number of attacks when compared to the other days. In addition, the percentage of false alerts had the highest value at 99.50% when compared to the other days.

3.4.2 The Second Experiment

The second experiment was performed in five days from the fourth week of the Darpa 1999 data set (Friday, Monday, Thursday, Tuesday and Wednesday).



Minimum Support 1

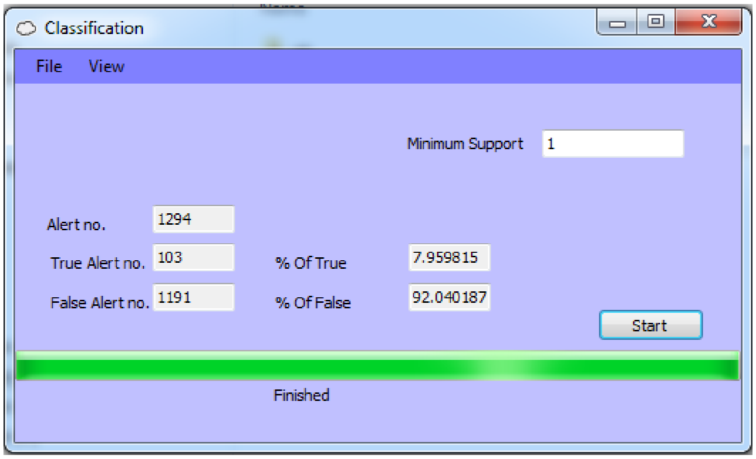
Alert no. 2598

True Alert no. 117 % Of True 4.503464

False Alert no. 2481 % Of False 95.496535

Finished

Figure 19. Fourth week –Friday



Minimum Support 1

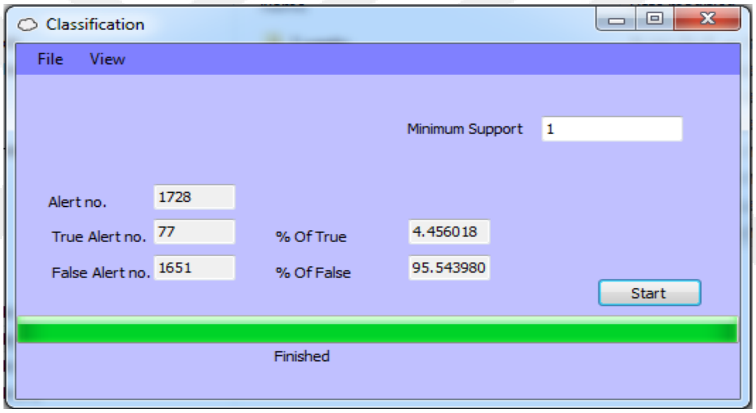
Alert no. 1294

True Alert no. 103 % Of True 7.959815

False Alert no. 1191 % Of False 92.040187

Finished

Figure 20. Fourth week –Monday



Minimum Support 1

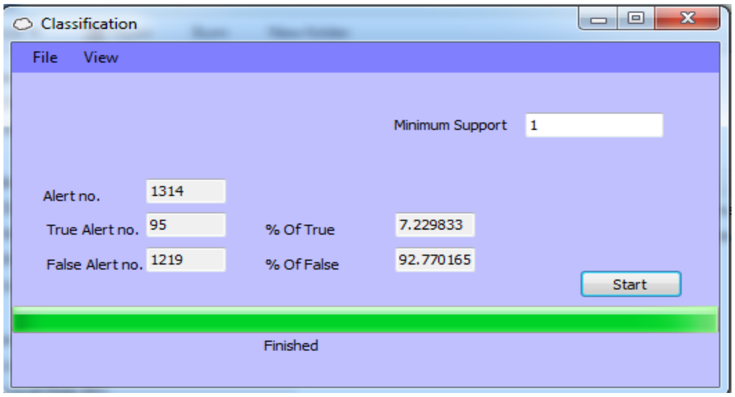
Alert no. 1728

True Alert no. 77 % Of True 4.456018

False Alert no. 1651 % Of False 95.543980

Finished

Figure 21. Fourth week –Tuesday



Minimum Support 1

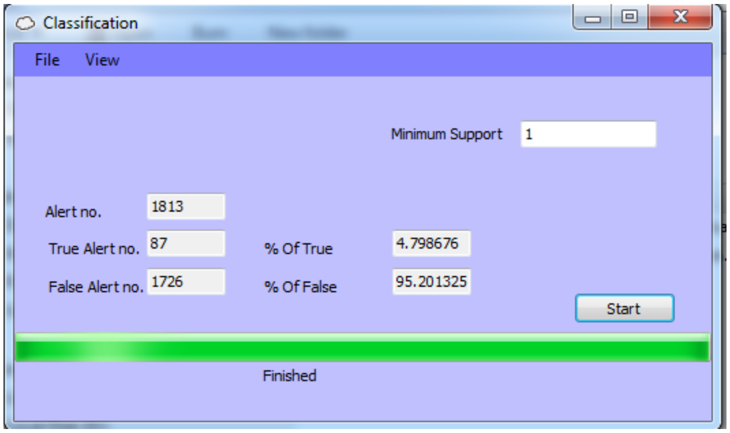
Alert no. 1314

True Alert no. 95 % Of True 7.229833

False Alert no. 1219 % Of False 92.770165

Finished

Figure 22. Fourth week – Thursday



Minimum Support 1

Alert no. 1813

True Alert no. 87 % Of True 4.798676

False Alert no. 1726 % Of False 95.201325

Finished

Figure 23. Fourth week – Wednesday

Table 15. Results of the second experiment

Day	Total alerts before preprocessing	No. alerts	True alerts no.	False alerts no.	% of true	% of false
Friday	2598	2597	117	2481	4.50 %	95.4 %
Monday	1295	1294	103	1191	7.99 %	92.04 %
Thursday	1728	1727	77	1651	4.45 %	95.54 %
Tuesday	1314	1314	95	1219	7.22 %	92.77 %
Wednesday	1813	1812	87	1726	4.79%	95.20%

Table 15 explained the results of the second experiment according to the implementation of the classification system on the fourth week.

From Table 15, it became evident how the total number of alerts was classified into true alerts (attacks) and false alerts (not attacks). Moreover, the percentage of correct alerts and wrong alerts were also obtained. The results indicate that the average of the percentage of false alarms (PFA) was equal to 94.19% ((95.4 + 92.04 + 95.54 + 92.77 + 95.20)/5). This explains that 94.19% of the total alerts in five days were false alerts. In addition, the percentage of true alerts was equal to 5.81%. Fig 24 illustrates the contradiction between the percentage of false alerts (PFA) and percentage of true alerts (PTA) for the fourth week.

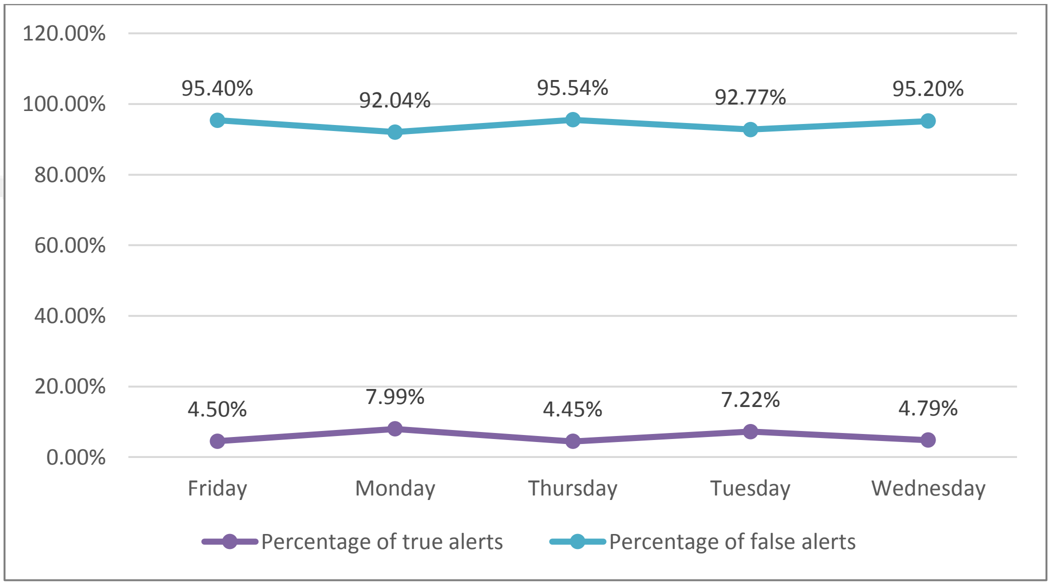
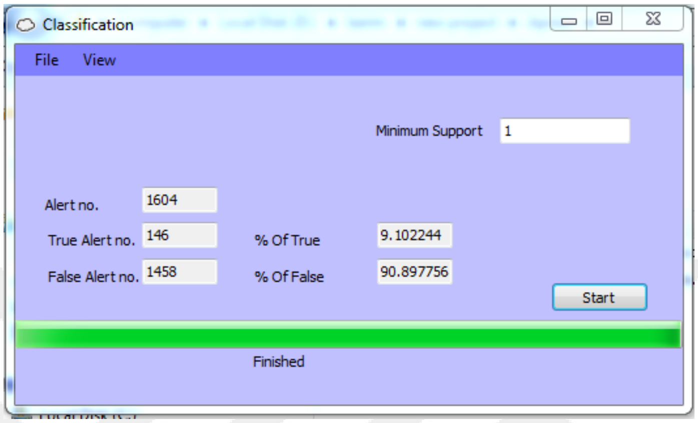


Figure 24. The contradiction between the PFA and PTA

According to Fig 24 on Monday, the percentage of true alerts was the highest value at 7.99% when compared with the other days in this week. This means that it had the maximum number of attacks. Moreover, the percentage of false alerts was the lowest value at 92.04% when compared with the same days. However, the percentage of true alerts on Thursday was the lowest at 4.45% when compared with the other days in the week. This indicates that it had the minimum number of attacks. In addition, the percentage of false alerts was the highest value at 95.54% when compared with the other days.

3.4.3 The Third Experiment

The third experiment was performed in five days from the third week of the Darpa 1999 data set (Friday, Monday, Thursday, Tuesday and Wednesday)



Minimum Support 1

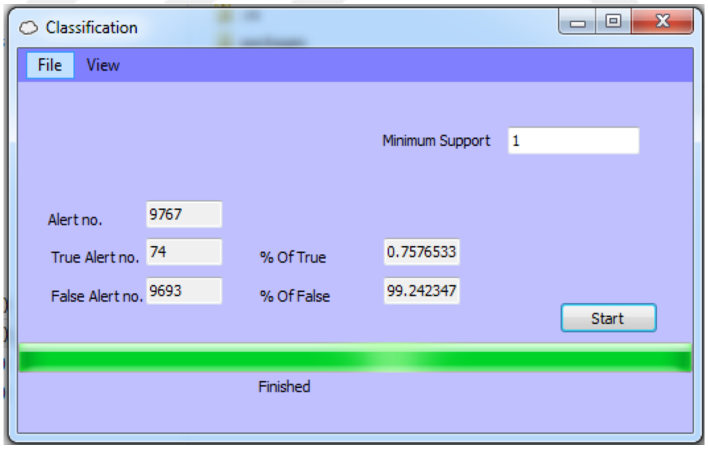
Alert no. 1604

True Alert no. 146 % Of True 9.102244

False Alert no. 1458 % Of False 90.897756

Finished

Figure 25. Fifth week –Friday



Minimum Support 1

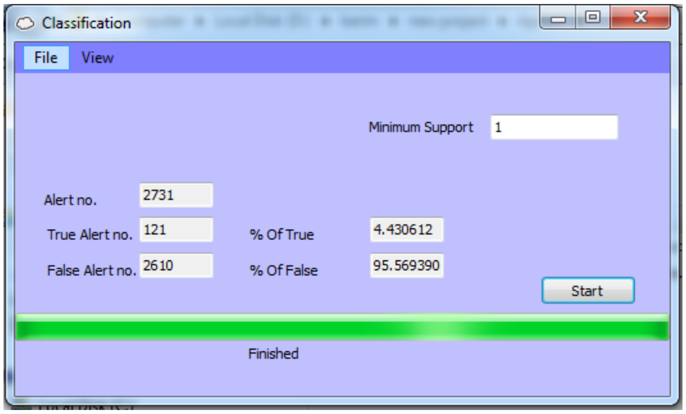
Alert no. 9767

True Alert no. 74 % Of True 0.7576533

False Alert no. 9693 % Of False 99.242347

Finished

Figure 26. Fifth week –Monday



Minimum Support 1

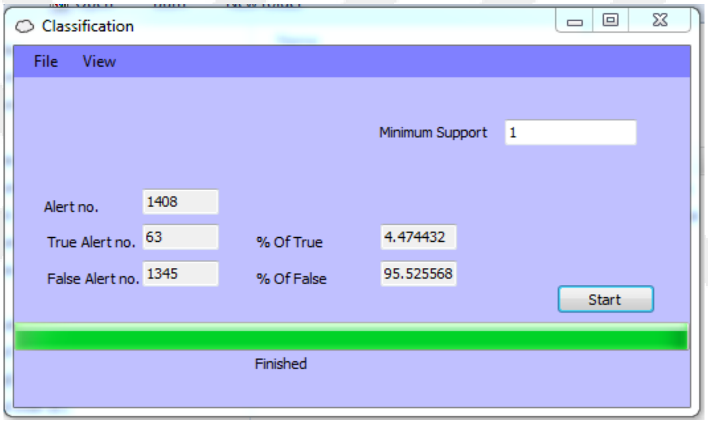
Alert no. 2731

True Alert no. 121 % Of True 4.430612

False Alert no. 2610 % Of False 95.569390

Finished

Figure 27. Fifth week –Thursday



Minimum Support 1

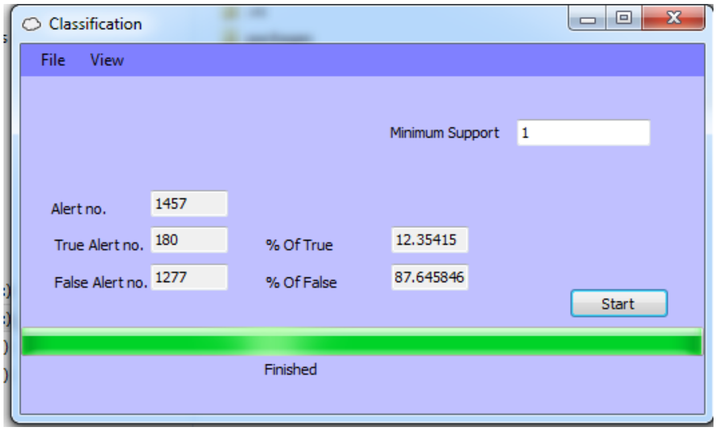
Alert no. 1408

True Alert no. 63 % Of True 4.474432

False Alert no. 1345 % Of False 95.525568

Finished

Figure 28. Fifth week –Tuesday



Minimum Support 1

Alert no. 1457

True Alert no. 180 % Of True 12.35415

False Alert no. 1277 % Of False 87.645846

Finished

Figure 29. Fifth week –Wednesday

Table 16. Results of the third experiment

Day	**Total alerts before preprocessing**	**No. alerts**	**True alerts no.**	**False alerts no.**	**% of true**	**% of false**
Friday	1604	1604	146	1458	9.10%	90.89%
Monday	9769	9766	74	9693	0.75 %	99.24 %
Thursday	2731	2730	121	2610	4.43 %	95.55%
Tuesday	1408	1408	63	1345	4.47 %	95.52%
Wednesday	1457	1457	180	1277	12.35 %	87.64 %

Table 16 illustrates the results of the third experiment according to the implementation of the system on the fifth week.

According to Table 16, it became clear how the total number of alerts was classified into true (attack) alerts and false alerts. Moreover, the percentage of correct alerts and the wrong alerts were obtained. The results show that the average of the percentage of the false alarms (PFA) was equal to 93.768% ((90.89+99.24+95.55+95.52+87.64)/5). This means that 93.768% of the total alerts in five days were unreal alerts, while the average of the percentage of true alerts was equal to 6.232%. Fig 30 illustrates the contradiction between the percentage of false alerts (PFA) and the percentage of true alerts (PTA) for the fifth week.

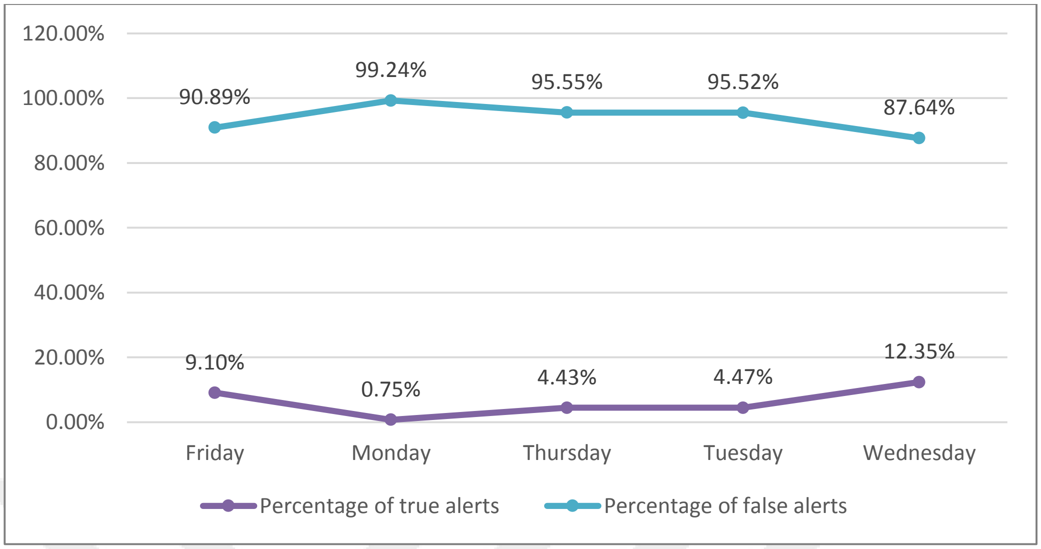


Figure 30. The contradiction between the PFA and PTA

According to Fig 30 on Wednesday, the percentage of the true alerts was the highest value at 12.35% when compared with the other days. This means that it had the maximum number of attacks. Moreover, the percentage of false alerts was the lowest value at 87.64% when compared with the same days. Nevertheless, the percentage of the true alerts on Monday was the lowest at 0.75% when compared with the other days. This indicates that it had the minimum number of attacks while the percentage of false alerts was the highest at 99.24% on the same day when compared with other days.

When the average of the percentages of true and false alerts were compared between weeks, it was observed that the highest average of the percentage of the false alerts was obtained in the second week. This means that the second week had the lowest number of attacks. After that, the fourth week came in second with a small difference with the first week. Finally, the fifth week had the highest average of the percentage of true alerts, which means that it had the highest number of attacks. In Fig 31, the results of the percentage of true alerts (PTA) and the percentage of false alerts (PFA) were drawn for the three weeks.

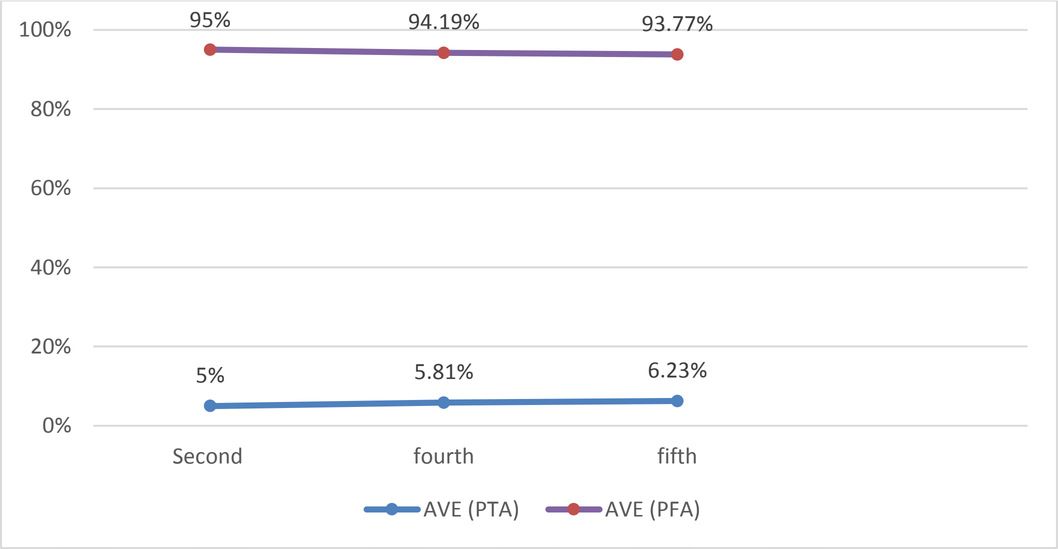


Figure 31. Contradictory the results of the average of the PTA and PFA

3.5 Comparison with other studies

The results were compared with previous studies of other researchers.

Katsikas and Spathoula tested their system on the DARPA 1999 dataset by using the second week, where they obtained a result of 75% as the average of the five days to find the number of false alerts. On the other hand, the proposed classification system in this thesis obtained 95% as the average of the second week.

Jiawei and Waita tested their method to find false positive alarms. They obtained a result of 78% as an average for three days (Thursday and Friday from the fifth week and Thursday from the fourth week). However, in this study, the classification system obtained 93.99 % ((95.52 + 90.89 + 95.45)/3) as average for the three days.

Finally, Giacinto, Roli and Perdisci applied their proposed alarm clustering system and obtained a result of 64.6% as an average of three days (Thursday and Friday from the fifth week and Thursday in the fourth week). However, the classification system produced an average of 93.99% for the three days.

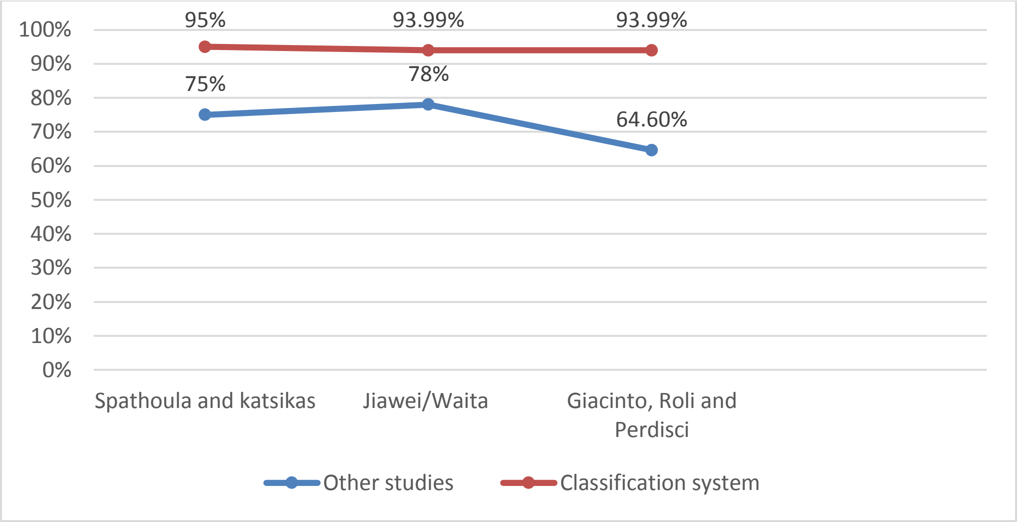


Figure 32. The contradiction between the proposed system and the other studies

Fig 32 explains the comparison of results between the proposed classification system and the other studies. The proposed system in the second week produced on average a percentage value of false alerts of 95%. This was the best ratio when compared with the results of Spathous and Katsikas, who reported 75%. Moreover, the classification system obtained a percentage value of false alerts in the three days of 93.99%. This is also a better ratio than the results of Waita and Jiawei, who produced a result of 78%, and the results of Perdisci, Giacinto and Poli, who obtained 64.60%.

It becomes clear that the proposed system would provide better results in each week and achieve the goal of its design.

4. CONCLUSION AND FUTURE WORK

This chapter focuses on the conclusions that can get from the content of the thesis and the future work to enhance the work.

4.1 Conclusion

The basic goal of the IDs is to reveal whether the system or network is under attack. The IDs collects and analyzes information from different parts of a network or computer. This identifies security violations, which include attacks against an organization from outside (intrusions) and from inside (misuse). The IDs detects these intrusions, then informs the administrator as alerts. However, the IDs suffers from the major problem of too many unreal alarms. Because of the abundant number of alerts, a security analyst is not able to distinguish between correct and incorrect alarms. Moreover, it has been found that analyzing the alerts is futile and ineffectual whenever a long time is required to analyse a large number of alarms. Therefore, a method has been proposed to solve this issue by enhancing an apriori algorithm which is used to design the classification system.

The proposed classification system helps to classify the input alerts and produce a percentage of correct and false alarms. The architecture and methodology of the system are explained in Chapter 2. This system consists of two modules, the first of which is the preprocessing module which includes two components, the first of which is used to remove unnecessary features and duplicate alerts. The second module is a classification module which consists of five components. These components are used to generate the feature sets, calculate the confidence from each rule, calculate threat scores for each IDs alarm, classify the alarms of the IDs as true or false, and calculate the percentage of true and false alerts.

The system is applied on the DARPA dataset, which contains five weeks each of which has five days. However, the three experiments were performed separately on the second, fourth and the fifth weeks. The results of the PFA and PTA in one day were compared with the other days in the same week. Later, the average of the PTA and PFA of each week were compared with each other to check which week had the maximum number of false alerts. From the results, the classification system was tested to verify whether it obtained the best results when compared to previous studies. Based on the comparison, the proposed system achieved the best percentages.

According to the results in Chapter 3, it is clear that the proposed system has achieved the goal of its design, namely to classify the alerts depending on their degree of seriousness into false and true alerts. This will give a security analyst the facility to focus on true alerts and perform his work in the shortest time after spending a long time to analyze alerts.

4.2 Future work

In the future, the work on finding a way to detect false alerts will continue. Therefore, it is suggested to add a method of evaluation which contributes to yielding high results. Moreover, it is recommended to improve the proposed classification system and apply it to another dataset, such as the KDD dataset.

REFERENCES

[1] D. Dharmendra G. Bhatti, P. Viparia , "Data processing for reducing false positiverate in intrusion detection", International journal of computer applications(0975-8887) volume57-no.5, november2012.

[2] N. Alwan Hussein, "Design of a network- based anomaly detection system using vft algorithm" , Master of science in computer engineering , Eastern Mediterranean university , may 2014.

[3] V.suramwar, B. Bansode., "A survey on different types of intrusion detection systems", International journal of computer application (0975-8887) Vo. 122-no 16, july 2015.

[4] D. hopkins , P. tokere ,"computer security: intrusion , detection and Prevention ", Nova science publishers 2009.

[5] S. northcutt, R. Jacob, B. Jaybeale, A. James ,C. foster ,T. Michael rash ,"Snort 2.1 Intrusion detection second edition ", Syngress puplishing ,Inc. ,2004.

[6] A. mohmed , N. Bashah , B. Shanmugum, "A brief introduction detection system " , S.G.ponnambalam etal.:iram 2012 , ccis 330, pp.263-2771,2012 .

[7] D. Wang, Y. Yeung, and T. Tsang, E.C., “Weighted Mahalanobis Distance Kernels for Support Vector Machines”, IEEE Transactions on Neural Networks, Vol. 18, No. 5, Pp.1453-1462, 2007.

[8] R. Base, P. Mell, “Special publication on intrusion detection systems”, NIST Infidel, Inc., National Institute of Standards and Technology, Scotts Valley, CA, 2001.

[9] S. Sen,"A survey of intrusion detection system using evolutionary computation", Department of computer engineering, Hacettepe University, Ankara, turkey.

[10] H. Alanazi, R. Noor, B. Zaidan, A. Zaidan,” Intrusion Detection System: Overview”, Journal of Computing, Vol. 2, Issue 2, pp.32-48, February 2010.

[11] K. Kaliyamurthie, D. Parameswari, R. Suresh, “Intrusion Detection System using Memtic Algorithm Supporting with Genetic and Decision Tree Algorithms,” IJCSI International Journal of Computer Science Issues, Vol. 9,Issue 2, No 3, March 2012.

[12] J. kurose ,K. keith W. Ross : "Computer network a top-down approach sixth edition", Pearson Education, Inc, 2013.

[13] H. Byun and S. Lee, “Applications of Support Vector Machines for pattern Recognition: A Survey”, pringer-Verlag Berlin Heidelberg, 2002.

[14] L. AIndukashyap, "Study and analysis of network based intrusion detection System", International journal of advanced research in computer and communication Engineering vol. 2, Issue 5, May 2013.

[15] G. Bin Huang , D. Wang and Y. Lan, “Extreme learning machines: a survey”, Published: 25 May 2011_ Springer-Verlag, 2011.

[16] V. Jaiganesh, S. Mangayarkarasi , Dr. P. Sumathi ," Intrusion Detection Systems: A survey and analysis of classification techniques " , International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 4, April 2013.

[17] A. Sawant , J. Yadav, A. kaur, J. Deo, N. Dhange, "Intrusion detection system using data mining" , International journal of advanced research in computer and communication engineering , vol.4 , issue 2 , February 2015.

[18] J. Joseph , B. Lee, Das, B. Seet, "Cross-Layer Detection of Sinking Behavior in Wireless Ad Hoc Networks Using SVM and FDA", IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 2, Pp. 233-245, 2011.

[19] K. Julisch , M. Dacier , " Mining Intrusion Detection Alarm for actionable Knowledge" ,proceeding of KDD'02 , ACM press , new York , pp .366-375, 2002.

[20] K. Julisch , "Clustering intrusion detection alarm to support root cause analysis " , In ACM transaction on information and system security , vol. 6(4) , 2003 , pp . 443-471.

[21] C. Clifton and G. Gengo. "Developing Custom Intrusion Detection Filters Using Data Mining," , In Proc. of 2000 MILCOM Symposium, pp. 440-443.

[22] J. Long, D. Schwartz, and S. Stoecklin. "Distinguishing False from True Alerts in Snort by Data Mining Patterns of Alerts," In Proc. of 2006 SPIE Defense and Security Symposium, pp. 62410B-1-- 62410B-10.

[23] S. Al-Mamory, H. Zhang, R. Abbas. "IDS Alarms Reduction Using Data Mining", In Proc. of 2008 IEEE World Congress on Computational Intelligence, pp. 3564-3570.

[24] P. Spathoulas, K.Katsikas, "Reducing false positives in intrusion detection systems" , Journal of computers & security 29 ,35-44 , 2010.

[25] W. Waitanjogu , L. Jiawei , "Using alert cluster to reduce IDS alerts " , Master's thesis ,China university of school of computer and communication hunan, .2010.

[26] R. Perdisci , G. Giacinto and F. Roli , "Alarm clustering of intrusion detection system in Computer networks ", International journal engineering application of artificial intelligence 19,429-438, 2006.

[27] P. Tan, M. Steinbach, V. Kumar, "Introduction to Data Mining", Addison-Wesley, 2005.

[28] J. Han ., M. Kamber., "Data mining : concepts and Techniques: Second edition" ,the Morgan Kaufmann, 2006.

[29] Y. Wah, N. Ismail, S. Fong, "Predicting Car Purchase Intent Using Data Mining Approach ", Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) , 2011.

[30] S. Dhotpe , N. Tarapore ,"Design of intrusion detection system using fuzzy class- association rule mining based on genetic algorithm , International journal of computer applications (0975-8887) vol. 53-no.14 , September 2012.

[31] D. Hand, D. Mannila, H.,P. Smyth: "Principles of Data Mining". The MIT Press, Cambridge (2001).

[32] M. Tayyab , " Analysis of data mining mythology and techniques for intrusion detection " , City university research journal , volume 03 number 01 January 2013 article 12 .

[33] D. uadhyaya , S. jain , "Hybired approach for network intrusion detection system using K-medoid clustering and naïve bayes classification ", International journal of computer science issues ,vol.10 ,issue 3 ,no1 , may 2013.

[34] A. rtiRathod, A. Dhabariya, C.thacker, "A Survey on Association Rule Mining for Market Basket Analysis and Apriori Algorithm", International Journal of Research in Advent Technology, Vol.2, No.3, March 2014.

[35] P. Mandave, "Data mining using Association rule based on APRIORI algorithm and improved approach with illustration", International Journal of Latest Trends in Engineering and Technology (IJLTET), ISSN: 2278-621X, Vol. 3 Issue2 November 2013.

[36] S. Agrawal R., imielinski, "Mining association rules between sets of items in large DBs", In proceeding of the ACM SIGMOD conference on management of data, pages 207-216, 1993.

[37] K. Al-Saedi, S. Manickam, S. Ramadass, W. Al-Salihy and A. ALmomani, "Research proposal: an intrusion detection system alert reduction and assessment framework based on data mining", Journal of Computer Science. Vol. 9, Issue 4. PP: 421- 426. New York, USA, 2013

[38] B. Helm, "Fuzzy Association Rules: An Implementation in R," Master's Thesis, Vienna University of Economics and Business Administration Vienna, 2007.

[39] H. Byun and S. Lee, “Applications of Support Vector Machines for Pattern Recognition: A Survey”, Pringer-Verlag Berlin Heidelberg, 2002.

[40] R. Agrawal, H. Mannila, R. Srikant et al. Fast discovery of association rules.in: U.M.Fayyad, G. Piatetsky-ShaPiro, P. Smyth, and R. Uthurusamy, eds. Advances in Knowledge Discovery and Data Mining. CAUSA: MIT Press. pp.307-328,1996.

[41] R. lippmann , J. haines , W. Fried , D. korba , The 1999 DARPA off- line Intrusion detection evaluation .computer networks 34 (4), 579-595 (Special issue on recent advances in intrusion detection systems).