OMC-IDS: At the Cross-Roads of OLAP Mining and Intrusion Detection

Hanen Brahmi[1], Imen Brahmi[1], and Sadok Ben Yahia[1,2]

[1] LIPAH, Computer Science Department, Faculty of Sciences of Tunis, Tunis, Tunisia
[2] Institut TELECOM, TELECOM SudParis, UMR 5157 CNRS SAMOVAR, France
{hanenbrahmi,imen.brahmi}@gmail.com
sadok.benyahia@fst.rnu.tn

Abstract. Due to the growing threat of network attacks, the efficient detection as well as the network abuse assessment are of paramount importance. In this respect, the Intrusion Detection Systems (IDS) are intended to protect information systems against intrusions. However, IDS are plugged with several problems that slow down their development, such as low detection accuracy and high false alarm rate. In this paper, we introduce a new IDS, called OMC-IDS, which integrates data mining techniques and On Line Analytical Processing (OLAP) tools. The association of the two fields can be a powerful solution to deal with the defects of IDS. Our experiment results show the effectiveness of our approach in comparison with those fitting in the same trend.

Keywords: Intrusion detection system, Data warehouse, OLAP, Audit data cube, Association rules, Classification.

1 Introduction

As far as interconnections among computer systems grow rapidly, network security is becoming a major challenge. An *Intrusion Detection System* (IDS) has been of use to monitor the network traffic thereby detect whether a system is being targeted by network attacks [14]. Even that IDSs have become a standard component in security infrastructures, they still have a number of significant drawbacks [14]. Indeed, the volume of the audit data which an IDS has to monitor is huge and grows rapidly. In addition, they flag out lower accuracy and higher false alarm rates. Moreover, current IDS do not provide support for historical data analysis and data summarization [13]. Supporting a historical network database in conjunction with an IDS raises two important technical challenges [8]: (*i*) since network traffic monitors generate data continuously and at high-rate, the database needs to support a high data insertion rate [8]; (*ii*) to facilitate the security analysis, the database must quickly answer historical queries [8,13].

Recently, *Data Warehouses* (DW) and *On Line Analytical Processing* (OLAP) technologies have gained a widespread acceptance as a support for decision making [7]. In a DW architecture, data are manipulated through OLAP tools which

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 13–24, 2012.

offer visualization and navigation mechanisms of multidimensional data views, commonly called *data cubes* [7]. Along with the increasing complexity of networks, protecting a system against new and complex attacks, while keeping an automatic and adaptive framework, is a thriving issue. One answer to the problem could rely on the association of OLAP and data mining to allow elaborated analysis tasks exceeding the simple exploration of the traffic data. DW and OLAP techniques can help the security officer in detecting attacks, monitoring current activities on the network, historical data analysis about critical attacks in the past and generating reports on trend analysis [13]. While, data mining is known for its ability to discover knowledge from audit data [14].

In this paper, we investigate another way of tackling the aforementioned problems. Thus, we introduce a new IDS based on a DW perspective to enhance the accuracy of detection as well as to minimize the false alarm rates. To that end, our proposed system integrates the OLAP and data mining techniques to improve the performance and usability of an IDS. Firstly, we model the network traffic data as a multidimensional structure, called *audit data cube*. Secondly, we introduce a novel algorithm that provides a concise representation of multidimensional association rules mined from the audit data cube. Finally, a classifier is used to decide whether a new connection record is an attack or not using the set of multidimensional detection rules. Through extensive carried out experiments on the standard intrusion detection DARPA dataset, we show the effectiveness of our proposal on the IDS performance aspects related to the false alarms as well as the detection rates.

The remaining of the paper is organized as follows. Section 2 sheds light on some representative related work applying the data mining techniques into the IDS. We introduce our new IDS based on the OLAP and data mining techniques in Section 3. We also relate the encouraging results of the carried out experiments in Section 4. Finally, Section 5 concludes and points out avenues of future work.

2 Scrutiny of the Related Work

Before data mining techniques are introduced into the intrusion detection field, the latter was heavily dependent on a manually maintained knowledge basis to reflect the ever-changing situations. However, this traditional way is difficult and expensive [14]. Otherwise, within data mining techniques, the rules (or signatures) of normal and abnormal activities can be created automatically. It is also possible to detect new types of attacks through an incremental learning process. Additionally, data mining techniques provide the means to easily perform data summarization and visualization, that would be of great help to the security analyst in identifying areas of concern [14]. In the following, we survey the most prominent approaches dedicated to apply data mining techniques within the intrusion detection field.

- The **MADAM-ID system** [10] is considered as the first research work that shows how data mining techniques can be used to construct IDS in a more systematic and automated manner. Firstly, all network traffic is abstracted to connection records. The latter are classified into “normal” and “intrusion”.

- The **ADAM system** [2] is one of the best-known approaches that use association rules mining and classification algorithms to detect intrusions. The main moan that can be addressed to ADAM stands in its high dependency on training data for normal activities. However, the attack-free training data is difficult to afford, since there is no guarantee that we can prevent all attacks in real world networks.
- The **MINDS system** [6] allows the development of scalable data mining algorithms and tools for detecting attacks and threats against computer systems. In fact, the system clusters audit data using a density-based local outliers algorithm to detect intrusions. In addition, it applies an association pattern analysis to summarize the network connections that are highly ranked as anomalous by the algorithm.

On the one hand, although the data mining techniques could provide beneficial characteristics to IDS, there is a compelling need to develop methods and tools that can help in historical data analysis. On the other hand, within a typical network environment, many different audit streams, collected from multiple cyber sensors, are shown to be useful for detecting intrusions. Such data includes: (*i*) raw network traffic data; (*ii*) netflow data; (*iii*) system calls; and so on. Consequently, it is important to have an architecture that can integrate these heterogenous data sources into a unified framework. The research works of [13] focus on the OLAP techniques to represent network traffic data and relate it to the corresponding IDS alerts. In contrast, we propose to couple OLAP and data mining techniques for intrusion detection. The main idea behind our approach is to take advantage from OLAP as well as data mining techniques and to integrate them to the same analysis framework in order to improve the performance of an IDS. In this paper, we introduce a new IDS, called OMC-IDS (Olap *Mining and Classification-based IDS*), which affords a support for historical data analysis and data summarization as well as the capacity to handle any kind of data for intrusion detection.

3 OMC-IDS: Intrusion Detection Based on Olap Mining and Classification

The OMC-IDS enriches the OLAP techniques with data mining facilities to benefit from their cross capabilities they offer. Indeed, the audit data collected from different heterogenous resources goes through four stages. Firstly, the data is filtered to remove irrelevant information and a relational database is created containing the meaningful remaining data. This database facilities information extraction and data summarization based on individual attributes such as day, source, destination, etc. Secondly, an audit data cube is constructed using the available dimensions. Thirdly, the OMC-IDS system integrates OLAP technology and association rule mining in order to extract interesting information under different perspectives and levels of granularity. Finally, OMC-IDS uses a classifier to classify each connection record either as one of the attack types or normal.

In the following, we focus on the study of the three last steps of the OMC-IDS system.

3.1 Audit Data Cube: Construction and Manipulation

The data feeding a data warehouse and OLAP systems is usually organized into multidimensional data views commonly called *data cubes*. The latter contain fact tables related to several dimension tables. A fact table represents the focus of analysis and typically includes attributes called *measures*. These are usually numerical values that facilitate a quantitative evaluation of various aspects of interest. Dimensions include attributes that form hierarchies. As long as a hierarchy is traversed from finer to coarser levels, measures are aggregated. Hierarchies can be included in a flat table forming the so-called *STAR schema* [7].

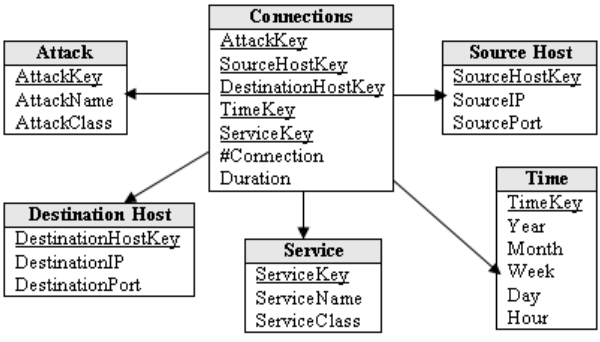

Fig. 1. A STAR schema for the IDS data warehouse

We propose to model the audit data as a multidimensional structure based on the STAR schema shown in Figure 1. The fact table "Connections" contains the attribute "#Connection" that measures the number of connections. The dimension "Time" includes information of date and time when the network packet was captured. The dimension "Service" contains the name and the class of service (or protocol) that was attacked. "Source Host" describes the source of IP addresses and port number. Likewise, the dimension "Destination Host" describes the destination of IP address and port. Similarly, the dimension "Attack" contains both the name of the attack and its type. Furthermore, hierarchies would give an extra edge for analysis purpose, since they allow decision-making users to see quantified data at different levels of abstraction. Therefore, security analysts must deal with hierarchies to exploit OLAP systems to their fullest capabilities. To do so, we define a concept hierarchy for each dimension in the audit data cube. For example, "Hour $\rightarrow$ Day $\rightarrow$ Week $\rightarrow$ Month $\rightarrow$ Year" is the hierarchy on the "Time" dimension. The dimension "Attack" can be organized into the hierarchy "Name $\rightarrow$ Class", *e.g.*, "*Smurf* $\rightarrow$ *DoS*". In addition, the hierarchies can be pre-defined or generated by partitioning the dimension into ranges. For instance, the dimension "Duration" could be partitioned into categories as "Low", "Medium" and "High".

Using the STAR schema described in Figure 1, a corresponding audit data cube would be a six dimensional structure in which a cell contains aggregates of the operations measures. For instance, a cell could correspond to short duration attacks over the FTP service in the period 1 pm to 2 pm on Oct. 20th 2011. The audit data cube can be constructed by using the SQL aggregation functions (*e.g.*, COUNT, SUM, MIN, MAX). For example, the COUNT value refers to the number of connections. The audit data can be manipulated with great flexibility and viewed from different perspectives by the use of data cubes. Indeed, OLAP operations (*e.g.*, ROLL-UP, DRILL-DOWN, SLICE and DICE) offer analytical modeling capabilities that can be applied on the audit data. The ROLL-UP operation allows the going from specific to general by climbing up the aggregation hierarchy. Otherwise, going from generalized data to more specific by stepping down the aggregation hierarchy is called DRILL-DOWN. The SLICE and DICE operations reduce the dimensionality of data by projecting the data on a subset of dimensions for selected values of other dimensions.

3.2 Multidimensional Association Rule Mining

The association rule extraction is a technique of data mining to discover interesting correlation relationships among data. In fact, the formalization of the association rule mining problem was initially introduced by Agrawal et *al.* [1]. Given a set of records, the objective of mining association rules is to extract all rules of the form $X \Rightarrow Y$ that satisfy a user-specified minimum support and minimum confidence thresholds, *i.e.*, $minSup$[1] and $minConf$[2]. X is the antecedent of the rule and Y is its consequent.

In the recent years, the problem of mining association rules from data cubes is knowing an increasing interest. The association rule mining can make OLAP more useful and easier to apply in the overall scheme of decision support systems. Further, OLAP is closely interlinked with association rules and shares with them the goal of finding patterns in the data. Indeed, data cube structures make good use of aggregated data, at the desired granularity levels, in the computation of the support and the confidence [3].

The multidimensional association rules is shown to be useful in increasing the detection accuracy and decreasing the false positives rate [12]. Consequently, the IDS performances can be greatly improved whenever the association rules are mined from the audit data cube. However, the number of the mined rules can be quite large, which affects the speed of IDS and hampers its whole performance [6,12]. Some of these rules are redundant since they contain patterns that correspond to the subsets of other patterns.

Example 1. Let R and R_1 tow multidimensional association rules. R: {Src_Port = 21 $\wedge$ Dst_IP = 192.63.11.11 $\wedge$ service = telnet $\wedge$ Duration = Long} $\Rightarrow$ {Attack = *Smurf*} and R_1: {Src_Port = 21 $\wedge$ service = telnet} $\Rightarrow$ {Attack = *Smurf*}. R and R_1 share similar features, *i.e.*, the patterns "Src_Port = 21" and "service =

[1] $minSup$ refers to the minimum support threshold pre-defined by the user.
[2] $minConf$ refers to the minimum confidence threshold pre-defined by the user.

telnet". If the respective supports of these two patterns are equal, then the rule R_1 is redundant *w.r.t* R.

To effectively mine the non-redundant multidimensional association rules from the audit data cube, we use the concept of closure [11] defined as follows:

Definition 1. *A pattern X is a closed pattern if there exists no pattern X' such that: (i) X' is a proper superset of X; and (ii) every connection record in a network traffic containing X also contains X'. The closure γ of a pattern X is the maximal superset of X having the same support value as that of X.*

In this respect, we introduce the AMAR (*Audit Multidimensional Association Rules mining*) algorithm intended to mine a concise representation of multidimensional association rules from an audit data cube $\mathcal{AC}$. The pseudo-code is shown by Algorithm 1.

Algorithm 1. The AMAR algorithm.

Input: $\mathcal{AC}$, $\mathcal{D}$, $\mathcal{H}_{\mathcal{D}}$, *minSup*, *minConf*.
Output: Set of multidimensional non-redundant association rules, *i.e.*, $\mathcal{X} \Rightarrow \mathcal{Y}$, with corresponding Supp and Conf.

```
1  Begin
2    C_1 := {1-candidate};
3    k := 1; /* |1-candidate| is the cardinality of attributes corresponding to D and
     H_D.*/
4    While C_k ≠ ∅ and k ≤ |1-candidate| do
5      CC_k := ∅;
6      FC_k := ∅;
7      Foreach candidate pattern A ∈ C_k do
8        CC_k := CC_k ∪ γ(A);
9      Foreach candidate closed pattern A ∈ CC_k do
10       Supp := ComputeSupport(A);
11       If Supp ≥ minSup then
12         FC_k := FC_k ∪ A;
13     Foreach A ∈ FC_k do
14       Foreach B ≠ ∅ and B ⊂ A do
15         Conf := ComputeConfidence(A − B, B);
16         If Conf ≥ minConf then
17           X := A − B;
18           Y := B;
19           return (X ⇒ Y, Supp, Conf);
20     C_{k+1} := ∅;
21     Foreach A ∈ FC_k do
22       Foreach B ∈ FC_k that shares (k-1) items with A do
23         If All Z ⊂ {A ∪ B} of k items are inter-dimensional and closed
           frequent then
24           C_{k+1} := C_{k+1} ∪ {A ∪ B};
25     k := k + 1;
26 End
```

Usually the user is interested in specified subsets of attributes in order to extract interesting relationships among them. So, (s)he needs to exclude the set of irrelevant attributes from the examination. To that end, AMAR allows the user to guide the analysis process by: (*i*) defining the set of dimensions $\mathcal{D}$ to be analyzed; (*ii*) choosing the hierarchies levels $\mathcal{H}_{\mathcal{D}}$ associated to the analysis dimensions; and (*iii*) setting the *minSup* and the *minConf* thresholds. As

sketched by Algorithm 1, we proceed by a bottom-up level wise search for frequent closed k-patterns, where the level k is the number of items in the set. We denote by C_k the sets of k-patterns that are potentially closed, CC_k the sets of closed k-patterns that are potentially frequent and FC_k the sets of frequent closed k-patterns. During the **initialization step** (line 2), our algorithm captures the 1-candidates from the user defined analysis dimensions $\mathcal{D}$ over the audit data cube $\mathcal{AC}$. These 1-candidates correspond to the attributes of $\mathcal{D}$, where each one complies with the chosen hierarchies $\mathcal{H}_{\mathcal{D}}$.

Within the **first step**, AMAR applies the closure concept (*cf.* Definition 1). The **second step** (lines 9-12) of our algorithm derives the frequent closed patterns FC_k from the closed candidate patterns CC_k that have a support greater or equal to $minSup$. The **third step** (lines 13-19) allows the extraction of association rules with a confidence greater or equal to $minConf$. The computation of support and confidence are performed respectively by the ComputeSupport and ComputeConfidence functions. Both functions directly pick up required precomputed aggregates from the data cube via MDX (MultiDimensional eXpression) queries [3]. The **fourth step** (lines 20-24) uses the set of frequent closed k-patterns FC_k to derive a new set of $(k+1)$-candidates, denoted by C_{k+1}. One $(k+1)$-candidate is the union of two k-patterns $\mathcal{A}$ and $\mathcal{B}$ from FC_k that respects three conditions: (i) $\mathcal{A}$ and $\mathcal{B}$ must have k-1 common patterns; (ii) all non empty sub-patterns from $\mathcal{A} \cup \mathcal{B}$ must be instances of inter-dimensional[3] patterns in $\mathcal{D}$; and (iii) all non empty sub-patterns from $\mathcal{A} \cup \mathcal{B}$ must be frequent closed patterns.

Table 1. A snapshot of an audit data cube with four dimensions

Service	Src_Port	Dst_Port	Attack	#Con
Imap	63587	143	*Neptune*	44
Imap	6161	143	*Satan*	26
Pop3	6161	110	*Neptune*	15
Pop3	63587	143	*Satan*	20
Tcpmux	63587	1	*Neptune*	64

Table 2. Multidimensional association rule list

ID	Rules	Sup	Conf
R_1	143 ⇒ *Satan*	0.3	0.5
R_2	143 ∧ 63587 ⇒ Imap ∧ *Neptune*	0.3	0.7
R_3	*Satan* ⇒ Imap ∧ 143 ∧ 6161	0.2	0.6
R_4	63587 ∧ *Neptune* ⇒ Tcpmux ∧ 1	0.4	0.6
R_5	Pop3 ∧ 143 ⇒ 63587 ∧ *Satan*	0.1	1.0
R_6	Pop3 ∧ 63587 ⇒ 143 ∧ *Satan*	0.1	1.0

Example 2. Table 1 sketchs an example of an audit data cube with four dimensions. The last row measures the number of connections using the aggregation function Count. The set of closed patterns, with their corresponding supports, is as follows: {("Pop3": 0.2), ("143":0.5), ("63587":0.7), ("6161": 0.2), ("*Neptune*": 0.7), ("Pop3, 143, 63587, *Satan*": 0.1), ("Pop3, 110, 6161, *Neptune*": 0.08), ("63587, 143": 0.3), ("143, *Satan*": 0.2), ("Imap, 143": 0.4), ("63587, *Neptune*": 0.6), ("Imap, 143, 6161, *Satan*": 0.1), ("Imap, 143, 63587, *Neptune*": 0.2), ("Tcpmux, 1, 63587, *Neptune*": 0.3)}. We extract the set of multidimensional association rules using the AMAR algorithm. Throughout our example, we set the $minSup$ to 10% and the $minConf$ to 50%. The algorithm generated 40 rules. Some of the extracted rules are illustrated in table 2.

[3] An inter-dimensional pattern is composed of items coming from different dimensions.

4 Classification

Intrusion detection can be considered as a classification problem where each connection is identified either as one of the attack types or normal based on some existing data [13]. Some of the association rules extracted by the AMAR algorithm are not useful since they do not imply an intrusion type in their consequent part. Therefore, we select the set of rules whose consequents include an intrusion label. For instance, according to the set of rules illustrated by Table 2, the rules R_3 and R_4 are excluded to retain only the rules R_1, R_2, R_5 and R_6. Then, we apply a decomposition axiom introduced in [4] (*cf.* Definition 2) to obtain new rules of the form "feature$_1$ $\wedge$ feature$_2$ $\wedge \ldots \wedge$ feature$_n$ $\Rightarrow$ intrusion". Even though, the obtained rules are redundant, their generation is mandatory to guarantee a maximal cover of the necessary rules.

Definition 2. *Given an association rule R, a decomposition axiom is defined as follows: If* $R : X \Rightarrow Y$ *then* $R_1 : X \Rightarrow Z$ *is a derivable valid rule,* $\forall Z \subset Y$.

Example 3. Let us consider the rule R_2: {Dst_Port = 143 $\wedge$ Src_Port = 63587} $\Rightarrow$ {Service = Imap $\wedge$ Attack = *Neptune*}. Using the decomposition axiom, R_2 is transformed in R'_2: {Dst_Port = 143 $\wedge$ Src_Port = 63587} $\Rightarrow$ {Service = Imap} and R''_2: {Dst_Port = 143 $\wedge$ Src_Port = 63587} $\Rightarrow$ {Attack = *Neptune*}. We retain the rule R''_2, since it includes an intrusion label in its consequent part.

Whenever the rules imply the same intrusion, we retain the rule which poses less constraints and can match more audit records.

Example 4. Let us consider two rules R: {Service = frag $\wedge$ Src_IP = 209.30.71.165 $\wedge$ Src_port = 110 $\wedge$ Dst_port = 32} $\Rightarrow$ {Attack = *Pod*} and R_1: {Service = frag, Dst_port = 32} $\Rightarrow$ {Attack = Pod}. Both rules R and R_1 imply the same intrusion label (*i.e.*, "Attack = *Pod*"). R_1 is considered to be more interesting than R, since it is needless to satisfy the features "Src_IP = 209.30.71.165" and "Src_port = 110" to highlight the attack "*Pod*". Hence, R_1 implies less constraints and can match more connection records than R.

Once the detection rules are generated, the OMC-IDS system applies a classifier [5] to classify the new connection records. Indeed, while having a new connection record C_{New}, the detection of an intrusion consists in traversing the detection rules from up to down in the classifier. The first reached rule, whose antecedent's part corresponds (*i.e.*, included or equal) to the features of C_{New}, will be of use. Thus, C_{New} will obtain the conclusion of the rule which indicates an attack.

Example 5. Let us consider a new connection record C_{New}: "service = frag, Src_IP = 209.30.71.165, Dst_port = 32". If we have in the classifier just the rule R (*c.f.* Example 4), we cannot classify C_{New} since the attribute "Src_port = 110" does not permit the matching. However, the rule R_1 (*c.f.* Example 4), which has a smaller antecedent than R, can classify C_{New}.

The latter example shows that the AMAR algorithm provides the relevant set of detection rules of need for the classification step of OMC-IDS. In fact, the use of such set of rules is of benefit for classifying new connection records.

5 Experimental Results

To evaluate the effectiveness and efficiency of our proposed system OMC-IDS, we carried out extensive experiments on a PC equipped with a 3 GHz Pentium IV and 2 Go of main memory running under Linux Fedora Core 6. Indeed, we compare our approach with the pioneering approaches falling within the intrusion detection-based classification trend, namely, ADAM [2] and C4.5[4] [9]. During the carried out experiments, we use the DARPA1998[5] dataset. The latter consists of training data and test data. The training data are generated in the first seven weeks and testing data are derived in the rest two weeks. The attacks consisting of a total of 33 different attack types are divided into four different attack categories, namely *DoS*, *R2L*, *U2R* and *Probing*. To build the audit data cube, we use the seven weeks' training data. To that end, we adopt the STAR schema showed in Figure 1. The audit data cube construction is done using the Analysis Services of SQL Server 2008.

Through these experiments, we put the focus on the assessment of the IDS performances in terms of detection and false alarms rates.

1. The *Detection Rate* (DR) is the number of correctly detected intrusions;
2. The *False alarms Rate* (FR) is the number of normal instances that were incorrectly considered as attacks.

Table 3. The DR (%) of OMC-IDS with respect to the dimension's variation

Dimensions	*DoS*	*Probe*	*U2R*	*R2L*
2-D	96.8	86.4	66.6	74.9
3-D	97.9	83.2	67.8	76.7
4-D	98.2	91.1	69.8	79.5
5-D	98.5	**95.3**	71.5	81.3
6-D	**99.5**	95.2	**74.9**	**86.6**

Table 3 shows the DR of OMC-IDS with respect to the dimension's variation for the four attack categories. The dimensions variation was established using the AMAR algorithm. From the results, we can remark that the dataset with six dimensions gives the best performances to detect the *DoS* class with 99.5% DR whereas the dataset with five dimensions gives the worst DR with 98.5%. Moreover, the dataset with five dimensions generates the best performance to detect the *Probe* class with 95.3% DR. The 6-D dataset gives the best performance to detect the *U2R* class with 74.9% DR and 5-D generates the worst performance with only 71.5%. Finally, the DR of *R2L* class on the 6-D dataset is the highest one, *i.e.*, 86.6% while on the 5-D we have the worst performance with only 81.3% DR. As consequence, OMC-IDS allows the detection of the attacks with best DR as far as the number of dimensions is the highest one, *i.e.*, six dimensions. Even

[4] Available in Weka: http://www.cs.waikato.ac.nz/ml/weka/

[5] Available at http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html

though, the DR decreases according to the decrease of the dimension number, it is still high.

The main challenge of IDS is to increase the value of the DR, while decreasing the value of FR. Figure 2 presents the DR and the FR, obtained respectively by, OMC-IDS, ADAM and C4.5-based systems. It can be seen that our approach drastically outperforms the other ones. In fact, Figure 2 (A) shows that OMC-IDS achieves a total DR above 99%, 97%, 86% and 74%, respectively corresponding to the detection of four attack categories (*i.e.*, *DoS*, *Probe*, *R2L* and *U2R*).

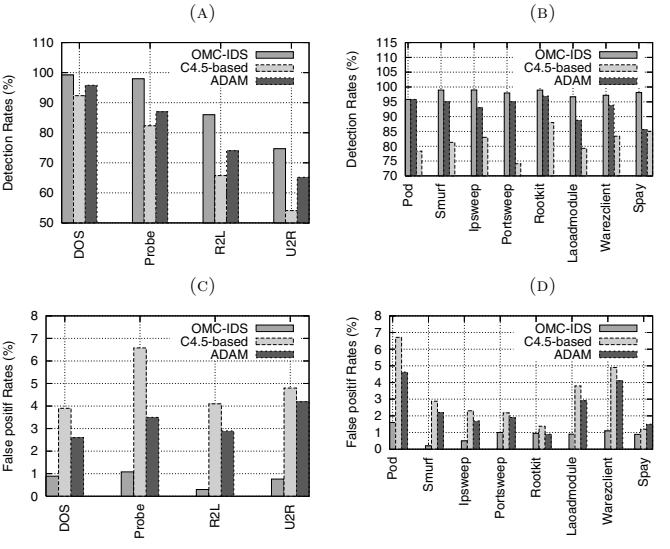

Fig. 2. Performance of OMC-IDS *vs.* ADAM and C4.5-based

Compared to ADAM, we remark that OMC-IDS provides a higher successful DR. Indeed, we achieved an average DR of 89% compared to 71%, over the four attack categories. On one hand, the high value of DR is explained by the use of pruning techniques that reduce the search space. In fact, the closed patterns have been shown to present the best compactness rates [11]. Thus, the mechanism adopted by the AMAR algorithm is more effective than that adopted by ADAM which is hampered by the ineffectiveness of the redundant association rules. On the other hand, the use of multidimensional association rules helps in improving the performance of detecting attacks. For example, let us consider the multidimensional association rule "{Src_Port = 21 $\wedge$ Dst_Port = 63 $\wedge$ Src_IP = 209.30.71.165 $\wedge$ Dst_IP = 180.66.11.11 $\Rightarrow$ Attack = *Satan*}". Obviously, the latter rule has higher accuracy than a single dimensional association rule "{Dst_IP = 180.66.11.11 $\Rightarrow$ Attack = *Satan*}". Consequently, we conclude that OMC-IDS is more efficient than ADAM due to the use of OLAP tools. In fact, the mining of multidimensional association rules from audit data cubes enhances the IDS process. Among the three tested systems, the C4.5-based IDS has the lowest

DR for the four attack classes. For instance, whenever OMC-IDS and ADAM have 74% and 65% DR for the *U2R* attacks, respectively, C4.5-based system has 54% DR. This is due to the stealthy nature of those attacks. Moreover, it is shown that C4.5 can classify more accurately on smaller datasets [9]. The results illustrated by Figure 2 (A) are confirmed by Figure 2 (B). The latter presents the DR of eight different attacks, including *Pod*, *Smurf*, *Ipsweep*, *Portsweep*, *Warezclient*, *Spay*, *Rootkit* and *Loadmodule*.

In addition, Figure 2 (C) shows that the FR ranges from 0.2% to 1%. The lowest FR is achieved for *DoS* attacks. The highest FR of *R2L* attacks generated by OMC-IDS is equal to 0.2%, which is a very low value compared to ADAM and C4.5-based systems. Precisely, according to Figure 2 (D), it is clear that the improvement of OMC-IDS with respect to ADAM is of 2%, 1.2%, 2% and 3%, respectively corresponding to the FR of the attacks *Smurf*, *Ipsweep*, *Loadmodule* and *Warezclient*.

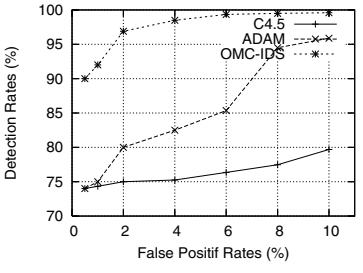

Fig. 3. The ROC curves of OMC-IDS *vs.* ADAM and C4.5-IDS

Within intrusion detection, the ROC (*Receiver Operating Characteristic*) curve is often used to assess the performance of IDSs. Figure 3 compares the ROC curve of OMC-IDS *vs.* those of ADAM and C4.5-based systems. It can be seen that the DR grows quickly to its peak value within a small increase of FR. In addition, the result ensures that our system can achieve the highest DR with the lowest FR. Thus, we conclude that OMC-IDS is more effective than ADAM and C4.5-based systems due to the use of the OLAP techniques that helped in improving the performance of detecting attacks.

6 Conclusion and Perspectives

On Line Analytical Processing (OLAP) provides tools to explore data cubes in order to extract interesting information. In this paper, we have shown the potential of coupling OLAP and data mining techniques in order to improve IDSs. To that end, we designed a new architecture, called OMC-IDS, to model network traffic using a multidimensional data structure based on the STAR schema. Carried out experiments showed that OMC-IDS outperforms the pioneering approaches, *i.e.*, ADAM and C4.5-based systems. Future work will

include exploring the alert correlations to expand the capabilities of our system. We can combine data from multiple sources to obtain a better analysis of the alert correlations.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the ACM-SIGMOD International Conference on Management of Data, Washington, USA, pp. 207–216 (1993)
2. Barbara, D., Couto, J., Jajodia, S., Popyack, L., Wu, N.: ADAM: Detecting Intrusions by Data Mining. In: Proc. of the 2nd Annual IEEE SMC Information Assurance Workshop, West Point, NY, pp. 11–16 (2001)
3. Ben Messaoud, R., Rabaséda, S.L., Missaoui, R., Boussaid, O.: OLEMAR: An Online Environment for Mining Association Rules in Multidimensional Data, vol. 2, pp. 14–47 (2008)
4. Yahia, S.B., Nguifo, E.M.: Revisiting Generic Bases of Association Rules. In: Kambayashi, Y., Mohania, M., Wöß, W. (eds.) DaWaK 2004. LNCS, vol. 3181, pp. 58–67. Springer, Heidelberg (2004)
5. Brahmi, I., Ben Yahia, S., Slimai, Y.: IDS-GARC: Détection d'Intrusions Basée sur les Règles Associatives Génériques de Classification. In: Actes du 9ème Colloque Africain sur la Recherche en Informatique, Rabat, Maroc, pp. 667–674 (2008)
6. Chandola, V., Eilertson, E., Ertoz, L., Simon, G., Kumar, V.: Data Mining for Cyber Security. In: Singhal, A. (ed.) Data Warehousing and Data Mining Techniques for Computer Security, pp. 83–103. Springer (2006)
7. Chaudhuri, S., Dayal, U.: An Overview of Data Warehousing and OLAP Technology. SIGMOD Record 26(1), 65–74 (1997)
8. Geambasu, R., Bragin, T., Jung, J., Balazinska, M.: On-Demand View Materialization and Indexing for Network Forensic Analysis. In: Proceedings of the 3rd USENIX International Workshop on Networking Meets Databases, Cambridge, MA, pp. 4:1–4:7 (2007)
9. Gyanchandani, M., Yadav, R.N., Rana, J.L.: Intrusion Detection Using C4.5: Performance Enhancement by Classifier Combination. In: Proceedings of the International Conference on Advances in Computer Science, pp. 130–133 (2010)
10. Lee, W.: A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems. Phd thesis, Columbia University, New York, NY, USA (1999)
11. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Mining of Association Rules Using Closed Itemset Lattices. Journal of Information Systems 24(1), 25–46 (1999)
12. Ping-Ping, M., Qiu-Ping, Z.: Association Rules Applied to Intrusion Detection. Wuhan University Journal of Natural Sciences 7(4), 426–430 (2002)
13. Singhal, A.: Warehousing and Data Mining Techniques for Cyber Security. Advances in Information Security, vol. 31. Springer (2007)
14. Singhal, A., Jajodia, S.: Data Mining for Intrusion Detection. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 1171–1180. Springer (2010)