Int. J. Nonlinear Anal. Appl.
Volume 12, Special Issue, Winter and Spring 2021 , 541-562
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/IJNAA.2021.5353

An incremental intrusion detection model using alarms correlation

Mohammad Ahmadzadeh[a], Javad Vahidi[b,*], Behrouz Minaei Bidgoli[c], Alireza Pourebrahimi[d]

[a]*Department of Management and Economics, Science and Research Branch, Islamic Azad University, Tehran, Iran.*
[b]*School of Mathematics, Iran University of Science and Technology, Tehran, Iran.*
[c]*School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.*
[d]*Department of Management and Accounting, Karaj Branch, Islamic Azad University, Karaj, Iran.*

(Communicated by Mohammad Bagher Ghaemi)

Abstract

Today, intrusion detection systems are extremely important in securing computers and computer networks. Correlated systems are next to intrusion detection systems by analyzing and combining the alarms received from them, appropriate reports for review and producing security measures. One of the problems face intrusion detection systems is generating a large volume of false alarms, so one of the most important issues in correlated systems is to check the alerts received by the intrusion detection system to distinguish true-positive alarms from false-positive alarms. The main focus of this research is on the applied optimization of classification methods to reduce the cost of organizations and security expert time in alert checking. The proposed intrusion detetection model using correlation(IIDMC) is tested on a valid test dataset and the results show the efficiency of the proposed model and consequently its high accuracy.

Keywords: Intrusion Detection, Fuzzy Correlator, Incremental Online Learning, Active Learning.

1. Introduction

Intrusion detection systems are security systems that are responsible for detecting attacks on computer systems and networks and they have great importance in today's world. But the question is, how fast can security experts update this enormous amount of data to discover new patterns of intrusion and update their security systems according to new rules?

*Corresponding author
Email addresses: m.ahmadzadeh@srbiau.ac.ir (Mohammad Ahmadzadeh), jvahidi@iust.ac.ir (Javad Vahidi), b_minaei@iust.ac.ir (Behrouz Minaei Bidgoli), a.pourebrahimi@kiau.ac.ir (Alireza Pourebrahimi)

Received: January 2020 *Accepted:* January 2021

How many security experts should be hired to monitor this environment? And how long and how much longer can organizations withstand the overwhelming cost of upgrading specialized personnel and updating security systems? This indicates that as the volume of information on signature-based approaches based on predefined patterns decreases, their performance in practice is declining and artificial intelligence-based approaches can be a viable alternative. Intrusion Detection Tools (IDS) usually operate on three main principles:

1. Based on predefined rules (signature)
2. through data mining techniques and artificial intelligence
3. A combination of both approaches.

But in practice, the signature-based approach is used. Therefore, in this research, the researcher seeks to present an intelligent model of intrusion detection in computer networks that is slightly dependent on historical data.
On the other hand, the main problem of intrusion detection systems is the generating of a large volume of false alarms, which are called false positive alarms [16]. For this reason, correlation systems have been developed that solve some of the problems of intrusion detection systems.
Correlation systems receive alarms and occurrences from different sensors on the network covered and process them in a multi-step process to create a high-level view of the current network security situation. Correlation systems identify false alarms received from intrusion detection tools, discard and prioritize incident-related alerts. A comprehensive model for correlating alarms is presented in [35], one of the most important of which is the alert checking stage. In this study, a comprehensive and hybrid intrusion detection model for alert checking using three online and incremental learning-based alarm generator systems, correlation system and active learning system is presented. The model presented is tested on a valid test dataset and the results are presented.
The next section reviews some related work in the area of intrusion detection and alerting.

2. Related Work

2.1. Intrusion Detection Systems

Research by Anderson in 1980 can be considered the beginning of intrusion detection systems [5]. Since then, researchers have repeatedly addressed this issue and developed various intrusion detection systems. The artificial immune system, as a system inspired by biology, has shown promising results in this regard.
In another study of intrusion detection, the association rule algorithm was used. The algorithm identified a variety of attacks and used the coarse-grained set theory algorithm for intrusion detection systems as well as the overall performance of the intrusion detection system [37].
In the reference [7] they have presented a hybrid negative selection algorithm for detecting anomalies. The purpose of this algorithm was to improve the negative selection algorithm when faced with high dimensional problems and data. In this study, binary and real identifiers have been used concurrently [7]. Each input sample is first evaluated with binary identifiers. If this sample is identified by binary identifiers, this sample is considered anomalous. If the input sample is not recognized by binary identifiers, it will be evaluated using real identifiers, if this sample is not recognized by binary identifiers, this sample will be considered as a normal sample and if identified by these real identifiers, Is considered an anomaly in the system.
In a study, the use of a combination of classifiers instead of a classifier in the intrusion detection

system was investigated. This improved the performance of the model. The results indicated that the performance of the proposed model was better than the models compared in this study [24].In another study the use of a genetic algorithm technique to extract weights in a neural network for a Intrusion detection system was performed. The results showed that neural networks improved performance [32]. A study using the data mining algorithm to develop a model for intrusion detection system presented by [8] showed that when applied to the correct number of clusters in the K-Means algorithm it can reach a high effectiveness rate [8].
The paper [1] reviews a hybrid classification based intrusion detection systems, in this study by comparing different approaches, the greedy randomized adaptive search procedure approach with the random annealing classification by [14] was introduced as one of the most effective approaches with a higher percentage of accuracy.In another study a two-layer clustering approach based on the nearest neighbor proposed by [23],The results also show very high accuracy.
In an article presented by Khan and et al., some data mining approaches such as SVM, KSVM, ELM, KELM are reviewed for intrusion detection and finally, it is concluded that combining the results of more than one data mining algorithm together can eliminate the other disadvantages [2].
In [11] has been attempted to generate rules for the classification of network activities using different data mining algorithms in network intrusion detection including K-means algorithm and linear regression and finally a comparative study of the performance of these algorithms is performed on the NSL-KDD dataset. The paper [6], under the fuzzy semi-supervised learning approach for intrusion detection, has shown that due to the lack of labeling for intrusion data and the need for expert efforts to label the data, Unlabeled real-world problems provide a fuzzy semi-surveillance model that, by large volumes of unlabeled samples, helps the supervised learning algorithm to improve classification performance for intrusion detection systems and finally compares its model with Bayesian classification algorithms. Simple, SVM and random forests are addressed by the NSL-KDD dataset Set. Guo et al. proposed a non-incremental IDS using representative instances[9].Yu et al. proposed an incremental learning method to cascade SC and ITI methods for supervised intrusion detection, called 'SC+ITI' [28]. Yi et al. proposed an improved incremental SVM algorithm, called RS-ISVM, to deal with network intrusion detection [36].

2.2. **Correlation Systems**

An adaptive learning system to classify alarms into two categories of "true positive" and "false positive" was proposed by Mr. Pietraszek [25]. The system uses human analyst feedback to build and update alerts, which has two working modes: recommender mode and operating mode. In recommender mode, the system does not process alarms, only predicts alarms and sends alerts to the human analyst. Predicted alerts are used to evaluate the accuracy of classification and prioritization of alerts by human analysts. In operating mode, the system can identify false and true positive alarms for the cluster confidence.
Another method of reducing false positive alerts has been proposed by Julisch, which is based on the use of data mining methods on the occurrence of recorded alert views [13]. The main idea behind this method is to identify the root cause of the event using alert clustering. If the root cause is a false positive event, all alarms resulting from such a root are also considered false positive.
In [18], an adaptive classification procedure based on the weighting of rules using human feedback is presented. In this method, the classification rules are first generated by a learning algorithm and weighted by the rules. Then the weights of the rules are updated using data labeled by human expert and adaptive learning.
In [33] a classification algorithm is presented to identify the important alarms of useless alerts. This method first uses the duplicate item exploration algorithm to extract duplicate patterns in

vain alerts. Then, using appropriate clustering algorithms, the appropriate patterns are extracted from the discovered patterns and finally, the knowledge gained is used to classify future alerts. The adaptive filtering method for alarms has been proposed by [19] to reduce false-positive and repetitive alarms. The proposed system has two subsystems, with one subsystem responsible for reducing alarms. The other subsystem has the task of adaptive learning for environmental changes and the use of expert human feedback.
In [27], a real-time framework for correlating alarms is presented that uses a new technique for alert accumulation and extracts new patterns of alarms. The post-processing alert filter is based on the processing and high sequence of alerts presented by [31], which is based on two important assumptions. First, the distribution of the number of neighbor alerts changes greatly from false positive to false positive. The second hypothesis is that if the frequency of a repeating alert exceeds the average of the alerts with similar signatures, the probability of the true positive alarm is high.
Yansong Liu1 and Li Zhu1 are proposed neural network-based intrusion detection and alarm method based on the research on the working principle and work flow of the existing intrusion detection system. Through the experiment of the prototype system, the results show that intrusion detection and neural network-based alarm systems have a higher detection rate and a lower false alarm rate for intrusion behaviors such as the denial of service attack and has higher detection ability for unknown attack behaviors[20].
Riyanat Shittua et al. is proposed a new framework titled A Comprehensive System for Analyzing Intrusion Alerts (ACSAnIA). The post-correlation methods include a new prioritization metric based on anomaly detection and a novel approach to clustering events using correlation knowledge. One of the key benefits of the framework is that it significantly reduces false-positive alerts and it adds contextual information to true-positive alerts [30].

3. Proposed Approach

The biggest weakness of intrusion detection systems is false positive generation. The security expert should examine the correlations of the generated alarms so that he can distinguish real alarms from false positives.
This drastically increases the cost of the organization. This study provides a comprehensive intrusion detection model that can help security experts by detecting knowledge gained using active learning and incremental online learning techniques.
The proposed model consists of two main parts, including the correlator system and the alarm generation system, which overall scheme is shown in Figure 1.

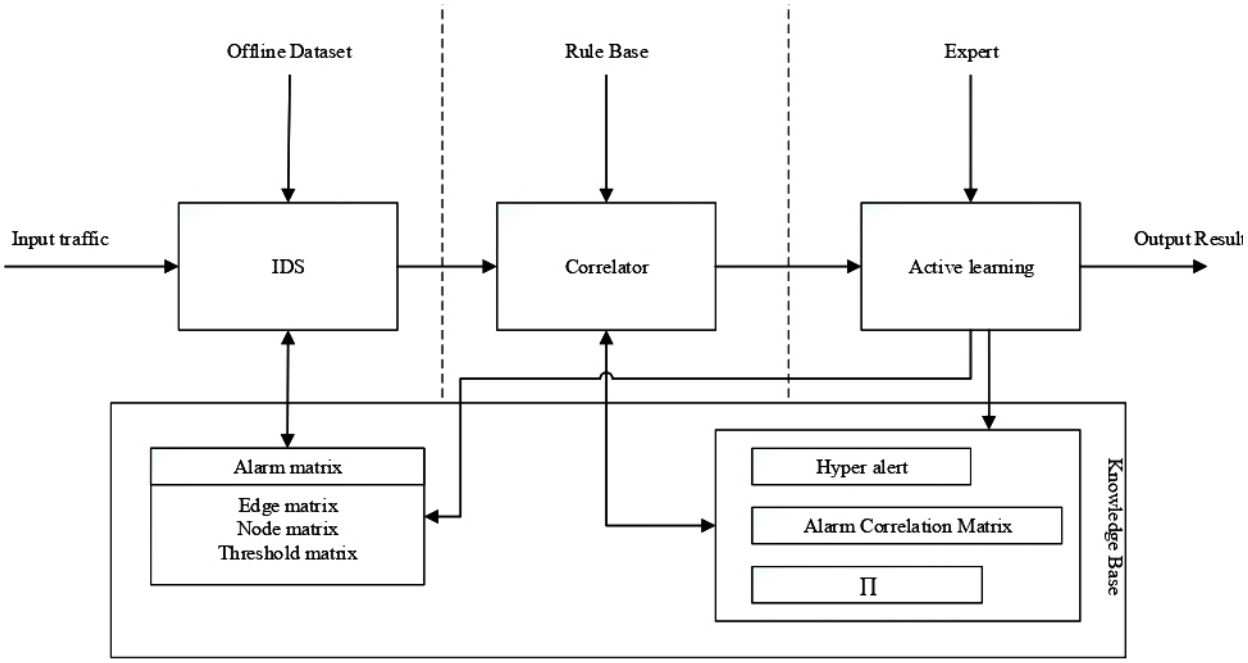


Figure 1: Incremental Intrusion Detection Model using Correlation.

As shown in Figure 1, the first part contains two modules. The first module uses a self-organizing incremental neural network-based algorithm to learn and the second module generates alerts using prior knowledge.
The generated alerts are transmitted to the correlator system to detect false positives and intrusion detection system errors. The correlator system examines some of the alert features generated with the help of knowledge stored in the knowledge base and measures its correlation, then alarm(s) and its correlations are transmitted to the third layer, an alert is detected as an attack if the correlation is higher than the threshold. Otherwise, the alert exact label is detected by asking the expert and then is proven editing and storing new knowledge with the help of active learning. The functionality of the proposed model modules is described below.

3.1. Proposed Alarm Generator System

As mentioned in the previous section, the alert generation system consists of two basic online learning modules and online incremental learning.

Offline Initial Learning Subsystem:
This module, as shown in Figure 2, clustered conventional attack classes using the two K-means and SOINN algorithms [12, 29]. Each of the defined classes is given separately to the SOINN algorithm. The SOINN algorithm has two important capabilities: first,The detection of number of clusters in a class and second, The initial location of cluster centers. SOINN has the unique feature of being able to extract such topology subclasses of the main class, but due to the incremental nature of SOINN, the input data order results in inaccuracies in the location of prototypes. But after using SOINN Now that the number of clusters and initial locations of prototypes have been determined, the K-means algorithm can efficiently determine the exact location of prototypes.

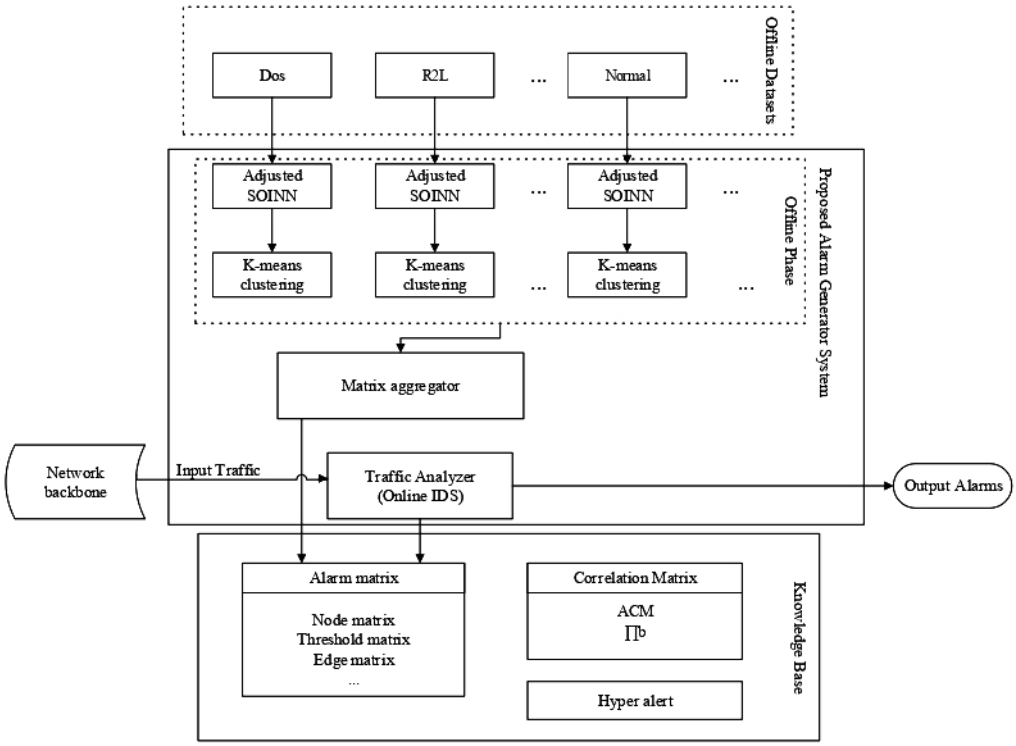


Figure 2: Proposed Alarm Generator System .

The output of this step is presented in an aggregation module. This module is required to aggregate the information generated from the previous step into the characteristic matrices, similarity thresholds, density edges and density matrices of the prototypes generated by each class separately, and store this knowledge into the knowledge base.

Incremental Online Learning Subsystem:
This subsystem analysis inbound traffic using the knowledge generated from the previous subsystem stored in the knowledge base. As shown in Figure 3, after receiving each input signal, find the nearest and second nearest prototype stored in the knowledge base, hereinafter referred to as the winner and the second winner.

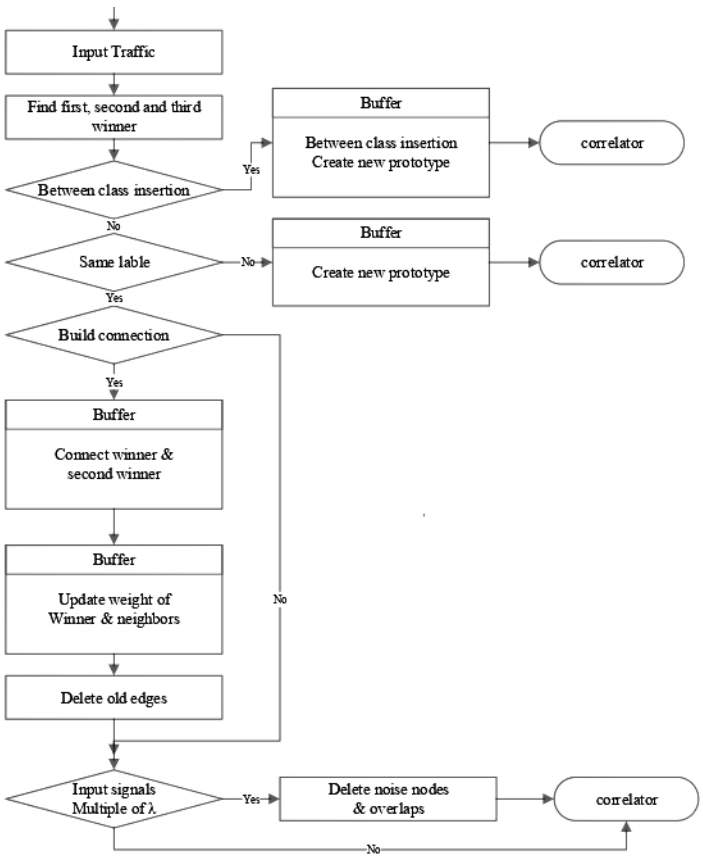


Figure 3: Incremental Online Learning Subsystem .

If the distance between the input vector with the winner and the second winner exceeds the similarity threshold of both nodes, the input signal is recognized as a new node in the network and is referred to the correlation system to determine the probability of correlation.
In the next step, it is checked that if the winner and second winner labels contradict together, the input signal is referred to the correlator with the winner alarm type, otherwise, using the similarity threshold matrix is determined that the input signal belongs to the winner and the second winner cluster. If the winner node and the second winner node are not connected in the matrix, they will be connected and then all the edges connected to the winner node will be added to their lifetime.
In step six, the winner weight vector and its direct neighbors are updated, i is used to mark the winner node and M_i to show the times for node i to be a winner and M is called nodes density vector. The change to the weight of winner ΔW_i and change to the weight of the neighbor node $j(\in N_i)$ of $i\Delta W_j$ are defined as:

$$\Delta W_i = \frac{1}{M_i}(W_s - W_i) \quad (1)$$

$$\Delta W_j = \frac{1}{100M_i}(W_s - W_i) \quad (2)$$

In the seventh step, the edges with a life that exceeds a predetermined threshold are removed. In step 8, it is checked that if the number of generated input signals is less than λ times the input signal of the winner class type is referred to the correlator system, otherwise it eliminates nodes that have no neighbor or only one neighbor. This eliminates prototypes created by noise or overlapping two clusters.
In the end, if the predefined thresholds(, including the number of input signals and the lifetime of the

edges) be higher then,the learned knowledge will be stored for a longer. But the lower the two, the more memory space and clustering speed are added and the system is updated with newer methods of intrusion. These parameters are adjustable for system detection efficiency.

3.2. Correlation System

The system receives the relevant alert as shown in Figure 4, the correlator system is required to calculate the probability of correlating with previous alarms by receiving an alert and labeling the type of attack from the alarm generation system. Previous alarms are stored in a module called the Hyper Alarm in the knowledge base and reviewed. Over time and with the knowledge gained, the information in this module will also evolve.

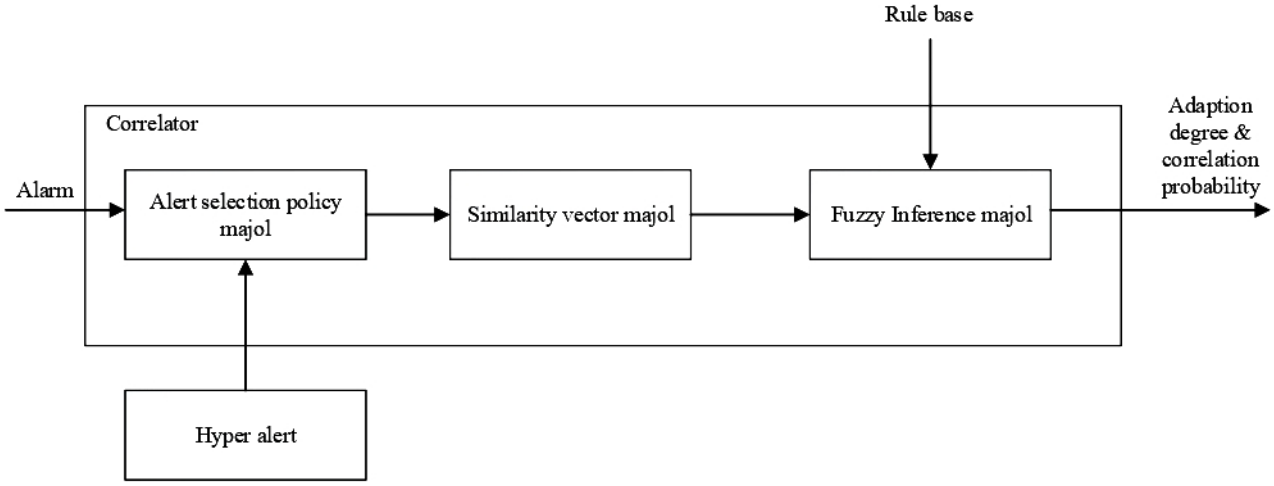


Figure 4: Fuzzy Correlation System .

Since it is not possible to calculate the probability of correlating all past alarms with the present alarms, it is possible to select the most probable alarms in the Hyper Alert by using a unit called Alert Selection Policy Unit. Another unit called the Similarity Vector Generation Unit combines the features of these two types of alerts to produce a similarity vector.
The fuzzy matching unit has the task of adapting this vector to the fuzzy rule base. A similar example of this Rule-base has been used in numerous sources [38].
The alarm is now being transferred to the active learning system for accurate labeling. The most important modules of this system are described below.

Correlation Data Retention Matrices:
The intensity of the correlation between the two types of alarms plays a fundamental role in analyzing the pattern of attacks and identifies the causal relationship between the two alarms. Instead of defining correlations between all types of alarms, Walds et al. [34] defined matrices to define the severity of correlations between eight classes of alarms.
In this study, the alerts correlation matrix was used. The elements of this matrix represent the correlation weight of the two types of aj and ai alarms obtained from the equation(3) [38].

$$w_{c(ai,aj)} = \sum_{k=1}^{n} p_{i,j(k)} \tag{3}$$

The term $w_{c(ai,aj)}$is the element corresponding to the rows i and j of the alert correlation matrix (ACM). In this equation, $p_{i,j(k)}$ is the probability of correlating the alarms of ai and aj in their k

observation, which is generated by the fuzzy correlator.
The correlation intensity of the two types of alarms ai and aj are equal by dividing the correlation weight of the two alarms ai and aj by the sum of the correlation weight of all alarms that occurred before aj and correlated with that obtained from the equation (4) [38].

$$\prod_{c(ai,aj)}^{b} = \frac{w_{c(ai,aj)}}{\sum_{k=1}^{n} w_{c(ai,aj)}} \tag{4}$$

Contrary to correlation weight, correlation intensity is an absolute value between zero and one and their magnitude indicates the high probability of occurrence following different types of alerts in the system. The elements present on the diameters of these matrices are also significant and indicate the probability of two successive alarms of the same type.

Similarity Vector Generation Unit:
Before any alert can be processed in the correlator system, It is necessary to generate the similarity vector using two alert information. Suppose a2 is the last alert generated by the intrusion detection system (the current alert) and a1 is the alert selected by the Alert Selection Policy Unit to check for correlation with a2. As shown in Figure 5, both a1 and a2 alerts contain multiple information such as alert time,destination IP address and port address, alert generator IDS, source and destination physical address, protocol number, alert priority, alert type (alert type is the chosen name by IDS) and so on. The similarity vector generator unit uses six features used in [38] to construct the similarity vector.

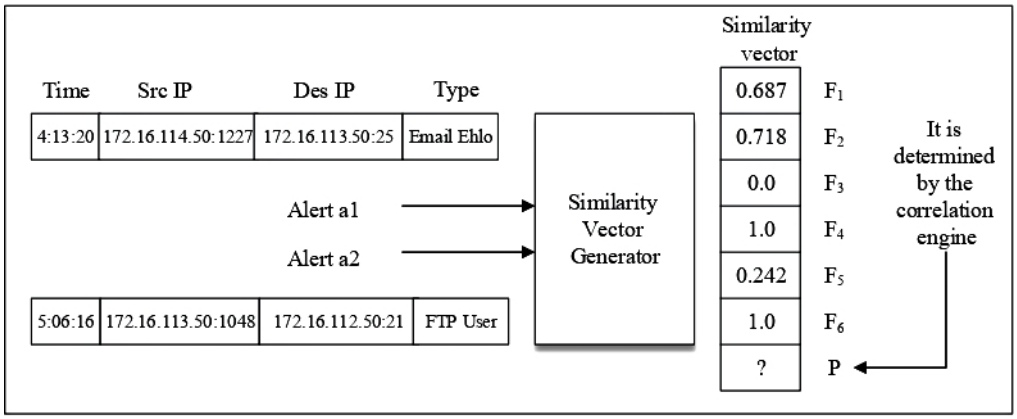


Figure 5: Similarity Vector Generation Unit .

Of the six features used, four are extracted directly from the two $a1$ and $a2$ alarms, but the other two are calculated by statistical analysis and using the knowledge stored in the $\prod$ and ACM matrices. These two features make the system somehow explore its operating history and use it to correlate new alerts. By using these two features in a kind of data mining concept and the use of statistical methods we also enter into the correlation probability calculation. The features used in generation each similarity vector are:

F1: The similarity of source IP addresses is $a1$ and $a2$, with values between zero and one.

F2: The similarity of the IP address of destination is $a1$ and $a2$, with values between zero and one.

F3: The destination port of $a1$ and $a2$ is equal to zero or one.

F4: The IP address of the second alarm source ($a2$) is equal to the IP address of the first alarm ($a1$) whose value is zero or one.

F5: correlation intensity between previous alarms of type $a1$ and $a2$, which is extracted from the history correlation matrix and is between zero and one.

F6: The number of times the previous alarms of type $a1$ and $a2$ are correlated. This value is initially zero and can gradually increase to zero.

To calculate the similarity of two IP addresses, we use the same number of bits in the two addresses by counting the most valuable bits. Assuming this number is n, the similarity of two IP addresses in IPv4 is obtained from the following:

$$Similarity(IP_1, IP_2) = n/32 \tag{5}$$

For example, binary displays for the two addresses are 192 .168 .10 .60 and 192 .168 .42.25 as follows:

192.168.10.60 : 11000000.10101000.00001010.00111100
192.168.42.25 : 11000000.10101000.00101010.00011001

They have 18 similar bits in their high-value bits, so their similarity is 18.32 and 0.56. If the two IP addresses are identical, the answer will be one, and if completely different, the answer will be zero.
Before exploiting any vulnerabilities in service, the attacker must check the port to see if it is open or closed. Because attacks are usually based on trial and error, they generate a large number of alerts that can indicate a network scan for that port if the two alarms are considered a destination port number. Therefore, the two alarms with equal port number can also indicate the relationship of those alarms. If the destination port is equal for both alerts then the value of this feature is equal to one, otherwise zero.
The F5 feature, which varies between zero and one, is a logical correlation intensity that essentially indicates the probability of correlating two alerts based on system experience of correlating previous alerts. At the beginning of the system, the value of the correlation probability is zero and does not affect the correlating probability but, over time the observed correlations may increase and even increase to 1 because of the correlation probability nature, It is possible two alarms without having a high amount of intensity and because of their other features are correlated together. $\prod$ can be helpful when there is uncertainty. For example, the similarity of the source IP address is low due to the use of spoofed addresses, but the correlation history in the form of correlation logger intensity indicates the high probability of correlation between the two types of alerts.

For example, suppose a system of two alarms, such as $a1$ and $a2$, of type $t1$ and $t2$, receive the following specifications and the system tries to evaluate their correlation.

$$(a_1 \cdot SrcIP = a_2 \cdot SrcIP) \; and \; (a_1 \cdot DestIP = a_2.DestIP) \; and \; (a_1 \cdot DestPort = a_2 \cdot DestPort)$$

The decision to correlate the above two alarms, despite the low correlation intensity, is clear. Subsequently, with the retrieval of $t1$ and $t2$ alerts and in uncertainty (for example 0.5 IP address similarity value), decision-making can be made with a high degree of correlation correlator and because of a previous correlation between high $t1$ and $t2$ alerts. This feature compensates for the low similarity of source IP addresses.
The sixth feature F6 indicates the frequency or relative frequency of these two types of alerts depending on system experience. This feature is a statistical feature and will still be useful under

uncertainty. This feature also plays a regulating role for f5, meaning that it does not allow the effect of F6 to be highly correlated after one or two correlations since the correlations of the two alarms cannot be statistically significant yet. They were assumed to be dependent and trusted at F5, but by repeating this correlation, it can be assumed that the two alarms are truly dependent. At this time the F6 feature has also increased to close to zero, and the F5 feature is fully mature and can be trusted. Naturally, F6 is initially zero but with increasing correlations the alarms will increase. The speed of this increase is one of the controllable parameters in the system.

Educational Data:

The training data used in the correlator is a finite set of basic knowledge that is manually defined and labeled. This knowledge is a set of simple correlations that can be described with basic knowledge of security concepts and attack scenarios.

Table 1: Rules used in correlation system .

	F_1	F_2	F_3	F_4	F_5	F_6	Class
1	high	high	1	0	high	high	20
2	low	low	1	1	low	high	15
3	high	high	0	0	low	low	16
4	low	low	1	0	low	high	9
5	high	high	0	0	medium	medium	18
6	medium	high	0	0	medium	medium	17
7	medium	medium	1	1	medium	medium	19
8	medium	medium	0	0	low	mod_low	3
9	low	high	0	0	low	mod_low	5
10	high	medium	1	0	medium	mod_low	14
11	low	low	0	0	low	low	1
12	medium	high	0	0	high	high	18
13	medium	medium	1	0	high	high	17
14	high	high	1	0	low	low	19
15	medium	medium	0	0	medium	low	4
16	low	low	1	0	high	high	14
17	low	low	1	1	high	high	19
18	low	low	0	1	medium	low	17
19	low	low	1	1	medium	medium	18
20	low	low	0	1	low	low	17
21	medium	medium	0	0	medium	high	7

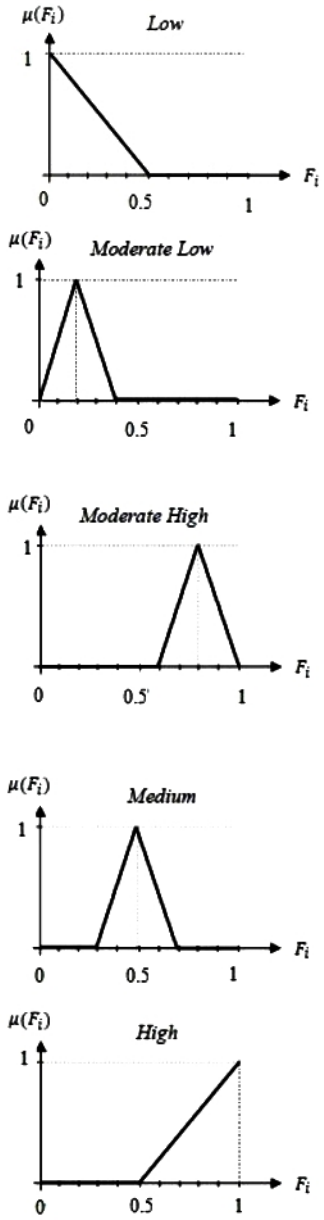
The number of training data defined in the correlator is very limited so that using the 21 training data the most general correlations between the two alarms are defined. Of the 21 training data, the 18 data are from the data used in [38] and three new data have been added to increase accuracy. To use this data in the correlation layer based on fuzzy rules, the same data are used with slight modifications such as fuzzy words like ”low”, ”high”, ”medium” and ”moderate” low to represent the value of each attribute is used.

Fuzzy Rules And Adapter:

Table 1 shows the twenty-one rules defined in the first correlation layer. As is clear, each rule consists of an antecedent, including status check for six attributes F1 to F6, and a label containing the class number assigned to that status. The general form of the rules used is as follows.

$$if\ (F_1 = V_1),\ (F_2 = V_2), \ldots\ and\ (F_6 = V_6)\ Then\ (Class = C)$$

The membership functions of all fuzzy expressions that define the rules as shown in Figure 6.



$$\mu(F_i) = \begin{cases} 1 - 2 * F_i & if\ 0.0 \le F_i < 0.5 \\ 0 & if\ 0.5 \le F_i \le 1.0 \end{cases}$$

$$\mu(F_i) = \begin{cases} 5 * F_i & if\ 0.0 \le F_i < 0.2 \\ 2 - 5 * F_i & if\ 0.2 \le F_i < 0.4 \\ 0 & if\ 0.4 \le F_i \le 1.0 \end{cases}$$

$$\mu(F_i) = \begin{cases} 0 & if\ 0.0 \le F_i < 0.3 \\ 5 * F_i - 1.5 & if\ 0.3 \le F_i < 0.5 \\ 3.5 - 5 * F_i & if\ 0.5 \le F_i < 0.7 \\ 0 & if\ 0.7 \le F_i \le 1.0 \end{cases}$$

$$\mu(F_i) = \begin{cases} 0 & if\ 0.0 \le F_i < 0.6 \\ 5 * F_i - 3 & if\ 0.6 \le F_i < 0.8 \\ 5 - 5 * F_i & if\ 0.8 \le F_i \le 1.0 \end{cases}$$

$$\mu(F_i) = \begin{cases} 0 & if\ 0.0 \le F_i < 0.5 \\ 2 * F_i - 1 & if\ 0.5 \le F_i \le 1.0 \end{cases}$$

Figure 6: Membership functions of all fuzzy expressions .

To determine the compatibility of a similarity vector such as x with values $(v_1, v_2, v_3, v_4, v_5, v_6)$ for its properties, with a law like R_j which with fuzzy values for its properties, with a law like $(v_1, v_2, v_3, v_4, v_5, v_6)$ Define the mean of the attribute values in that similarity vector to the sets contained in R_j. The equation of computing the degree of compatibility of x with R_j is expressed as a equation(6) [26].

$$Compatibility(x, R_j) = \frac{1}{n} \sum_{i=1}^{n} \mu(v_i, V_i) \tag{6}$$

In the following example, the compatibility of the similarity vector x with R1 (Rule 1 of Table1) is calculated to be 0.644 using the equation6 and the membership functions of Figure 6.

$x : (v_1 = 1,\ v_2 = 0.81,\ v_3 = 0,\ v_4 = 0,\ v_5 = 0.62,\ v_6 = 1)$
$R_1 : (V_1 = high,\ V_2 = high,\ V_3 = 1,\ V_4 = 0,\ V_5 = high,\ V_6 = high,\ Class = 19)$
$\mu(v_1, V_1) = \mu(1, high) = 1$ $\quad (v_4 = V_4) \Rightarrow \mu = 1$
$\mu(v_2, V_2) = \mu(0.87, high) = 0.625$ $\quad \mu(v_5, V_5) = \mu(0.62, high) = 0.24$
$(v_3 \neq V_3) \Rightarrow \mu = 0$ $\quad \mu(v_6, V_6) = \mu(1, high) = 1$
$$Compatibility(x, R_j) = \frac{1 + 0.625 + 0 + 1 + 0.24 + 1}{6} = 0.644.$$

The degree of compatibility of the x-similarity vector with each of the rules in Table 1is similarly examined to find the most consistent law with x. The degree of compatibility of x with the twenty-one rules used is as follows:

$R_1 : 0.644,\ R_2 : 0.167,\ R_3 : 0.603,\ R_4 : 0.333,\ R_5 : 0.67,\ R_6 : 0.503,\ R_7 : 0.067,$
$R_8 : 0.333,\ R_9 : 0.437,\ R_{10} : 0.4,\ R_{11} : 0.333,\ R_{12} : 0.644,\ R_{13} : 0.373,\ R_{14} : 0.437,$
$R_{15} : 0.4,\ R_{16} : 0.373\ R_{17} : 0.207,\ R_{18} : 0.233,\ R_{19} : 0.067,\ R_{20} : 0.167,\ R_{21} : 0.567.$

After examining the x-similarity vector with all existing rules, three rules with the highest degree of compatibility with x are selected. The class number assigned to the x similarity vector is then obtained from the class number in the above rules. After defining the final class number (C) at each step, it is necessary to convert the class number to the probability as follow:

$$P = \frac{C - 1}{\lambda} + \frac{1}{2\lambda} \tag{7}$$

Where λ is the total number of classes ($\lambda = 20$) to how the intervals corresponding to each class number are defined in equation7. In this equation,$\frac{1}{2\lambda}$ is used to transfer the probability value to the defined interval. Finally, the alarm, along with the probability of correlation and degree of adaptation, is referred to the third layer of active learning for decision making.

3.3. **Active Learning**

The output of the correlator system is the probability of correlating the two alarms. If the correlation probability is lower than the average correlation threshold and the rule selection threshold (that is considered 0.5 in this study),so for refining the label of that prototype(alarm),true label is asked from a reliable labeling reference, such as a human expert that in the literature of machine learning and data mining it is called active learning.
For example as shown Figure 7, if the correlation probability for an alert with the winner label from the alert generation system is less than 0.5, the first, even second, and third winner alarms must be forward to the active learning system with their's correlation probability and the rule compatibility degree.

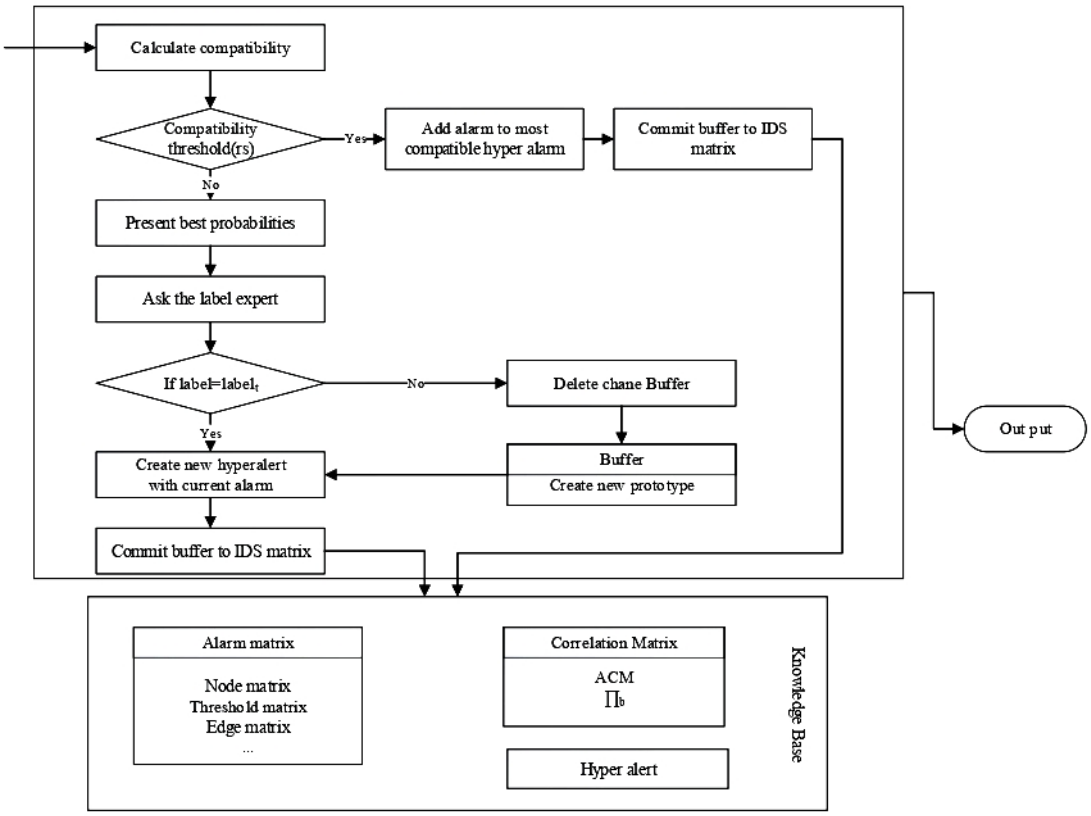


Figure 7: Active Learning .

This information can help the labeling authority, such as a human expert, to reach the exact label, although there is no need to use this information because the advantage of this system is that it can detect patterns of attacks that the system has not yet learned and incremental learning has not even recognized them over time, adding it to their knowledge base and thus saving time for security experts in the future. The functionality of the active learning system is that if there was a label other than the winner label of alarm generator system, active learning would add a prototype(node) with the true attack label in the set of matrices associated with the prototype in the knowledge base including node matrix, edge matrix, threshold vector and density vector(based on winning number). Second, if the winner node attack label is true, it adds a new hyper alarm containing an alert attack-type that is actively learned in the system.
This model can enhance the knowledge base through the systems and mechanisms mentioned above and improve its efficiency and effectiveness over time. Therefore, it is more efficient and more applicable than the theoretical models and algorithms mentioned in related work.
In the next section, we examine the functions of each system separately and, evaluate efficiency model.

4. Experimental Results

This section first introduces the data used to evaluate the proposed model, then evaluates the appropriate evaluation criteria, and in the last section evaluates the fuzzy correlation system based on active learning and then, the whole proposed intrusion detection model is examined.

4.1. Data Set

One of the most important challenges in the research and development of new intrusion detection systems based on anomalies are the lack of appropriate and publicly accessible datasets. This challenge has made it difficult to educate, evaluate, and compare innovations and methods in detecting malware-based intrusions. The following is a collection of the best datasets used by authoritative articles.

(1) DARPA2000 Dataset

The Department of Technology and Cyber Systems at the Lincoln Laboratory at MIT University with the support of the US Defense Advanced Projects Agency and the US Air Force Research Laboratory began collecting and publishing the first standard data for evaluating network IDSs in years 1998 and 1999. This assessment measured the detection probability and false-alarms probability for the IDSs under investigation and played a significant role in advancing IDS-related research and targeting it. The result were two DARPA 1999 and DARPA1998 datasets that are still used as the most well-known and most used datasets in IDS evaluation.
The DARPA2000 data is provided with more emphasis on multi-step attacks and scenario identification of these attacks [17]. There are two multi-stage attacks called LLDOS1.0 and LLDOS2.0. In this paper, using these two scenarios, we evaluate the correlation module of the proposed model.

(2) KDD CUP99 Dataset

This dataset is one of the most widely used intrusion detection databases available [15]. This dataset is a copy of the 1998 DARPA dataset prepared by MIT Lincoln Laboratory.
This dataset is used for the third international competition of data mining and knowledge discovery tools. The competition was to develop an intrusion detection system to detect good and bad communications. In other words, we can say that with the help of this dataset we can detect aggressors and attackers. This database contains a standard dataset for review that has a variety of simulated intrusions into military applications.
In most research papers, 10% of the training and experimental sets are usually used. Each sample contains 41 attributes and is labeled normal or one of 24 different attack types, which are grouped into four main attack types as shown in the following table.

Table 2: The number of instances for each group of attack in the KDD99 dataset .

Class	Training set	Test set
Normal	97,277	60,593
Probe	4,107	4,166
DoS	391,458	229,853
R2L	1,126	16,349
U2R	52	68
Total	494,020	311,029

Although this dataset has drawbacks and limitations, many articles have been used for evaluation [10], so it has been used to evaluate the proposed intrusion detection model in this paper.

4.2. Evaluation Criteria

Incremental Fuzzy Correlator Commonly used three criteria of soundness, completeness, and error rate in correlational systems review [21].
Soundness of the transmitter indicates the accuracy of the alerts selected in the extraction scenario. Completeness indicates the completeness of the generated scenario by the correlator, than the desired

scenario.
False Correlation Rate Indicates the error rate in selecting alarms for correlation.

$$Soundness = \frac{TC}{TC + FC} \tag{8}$$

$$Completeness = \frac{TC}{TC + FN} \tag{9}$$

$$FalseCorrelationRate = \frac{FC}{TC + FN} \tag{10}$$

TCs are truely correlated alarms, TN represents alarms that are not properly correlated, and FC shows all false-correlated alarms and FN alarms that are not correlated.
It is clear that the value of the first two criteria varies between zero and one and the closer one is, the greater the accuracy of the extraction scenario. The third criterion value is greater than or equal to zero and the lower the value, the greater the accuracy of the extraction scenario. Together, the criteria presented can represent the accuracy of a scenario, and each expresses one aspect of its accuracy.

Intrusion Detection Model Using Correlator, The accuracy is not a reliable metric to evaluate IDSs because of the presence of the imbalanced class distribution. It is easy to achieve high accuracy for minority class that depends on the sample size of the training and test set [10]. Alternatively, the true positive rate (TPR) and false negative rate (FPR) are two useful metrics to evaluate the IDSs, which are independent of the sample size of the training and test set:

$$TPR = \frac{TP}{TP + FN} \tag{11}$$

$$FPR = \frac{FP}{TN + FP} \tag{12}$$

In equations 11 to 12, TP, FP, TN and FN denote the number of true positives, false positives, true negatives and false negatives.

4.3. Evaluation And Comparison

Expriment 1: The accuracy of the extracted scenario by the correlator is investigated using three criteria of soundness, completeness and false correlation rate and is also compared with some known works by the proposed incremental fuzzy correlator results as shown in Figure 8. Proposed fuzzy set of static methods presented in [3, 4, 21, 22] which already know how to attack LLDOS1.0 with the help of dozens of laws.
Because these methods were based on the rules of prerequisites and consequence. Therefore, before correlating, it is necessary to define the relationship between the steps of each attack and in the form of dozens of rules store it in the system database, it seems to have acceptable results. The result generated by the proposed correlator is very close to the results reported in [21] and [22], which is one of the most well-known and most cited works on these data.

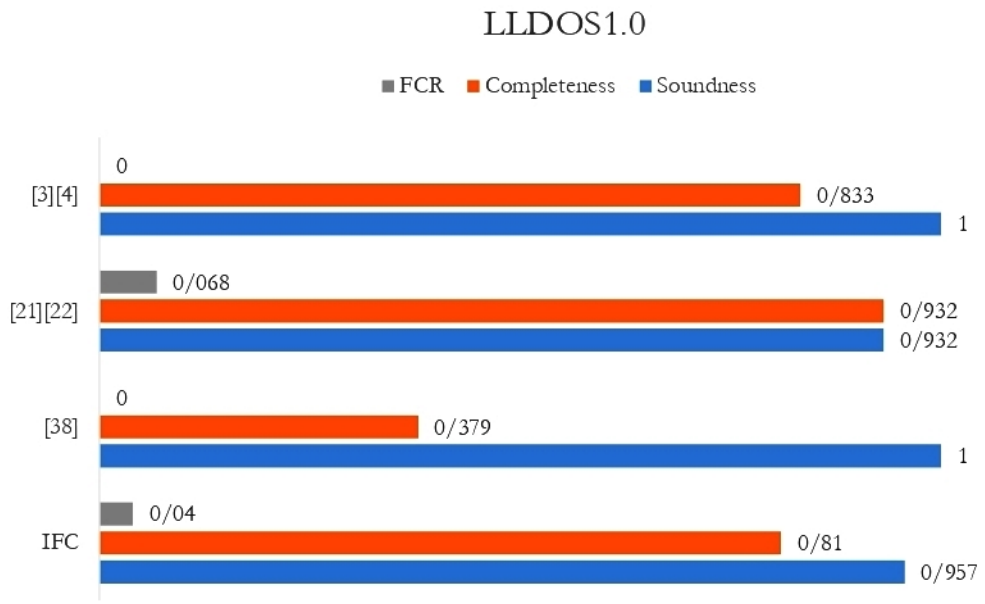


Figure 8: Comparison of precision Incremental Fuzzy Correlator(IFC) with several similar research on the scenario LLDoS1.0 .

Also, the results obtained by the proposed fuzzy correlator are very close to the results reported in [3] and [4]. Although the results reported in these two tasks appear to be strong for the LLDOS1.0 attack but in extracting the LLDOS2.0attack works very poorly as shown in Figure 9.

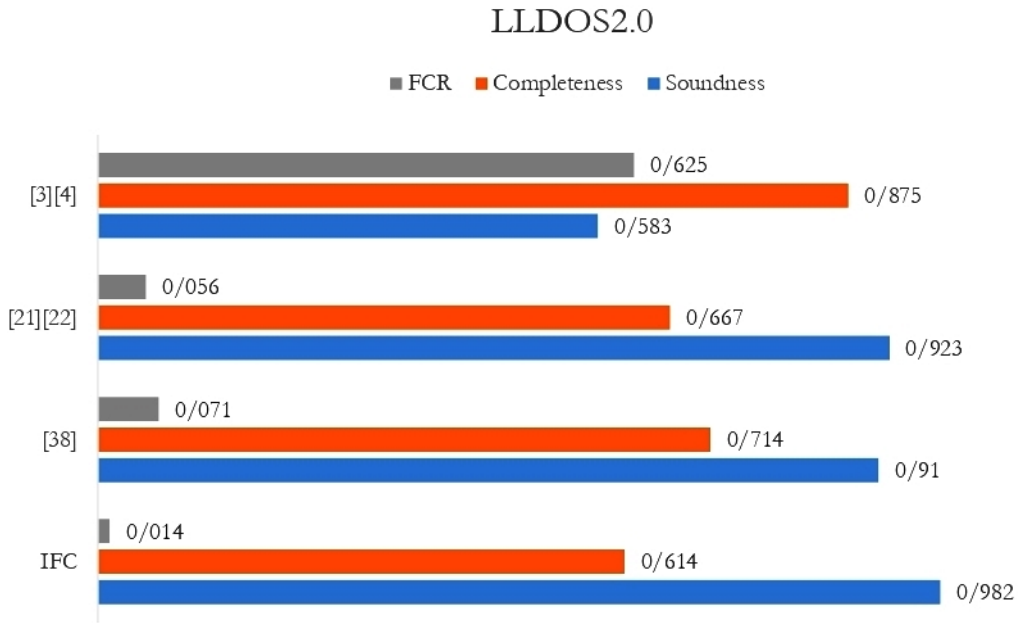


Figure 9: Comparison of precision Incremental Fuzzy Correlator(IFC) with several similar research on the scenario LLDoS2.0.

Also, both such works as [3] and [4] were based on static rules and prior knowledge acquisition and were not capable of detecting new attacks. The proposed fuzzy correlator is also very accurate compared to the dynamic methods. For example, in [38], which, like the proposed active learning-based fuzzy correlator, operates dynamically, the inclusion criteria are significantly less problematic than the active learning-based fuzzy correlator.

As shown in Figure 10 and 11 examine the three main criteria used in accuracy assessment, such as soundness, completeness and false correlation rate in correlator. The alarms in this series were investigated using two different alert selection policies, namely all and random selection for two single-layer and two-layer modes. The purpose of this experiment design is to show the positive effect of two-layer design on improving the accuracy of correlation detection.

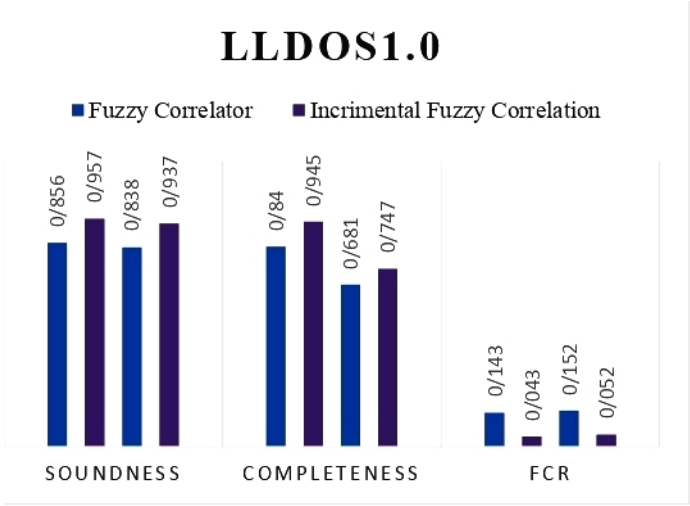


Figure 10: Accuracy assessment for the LLDoS1.0 scenario.

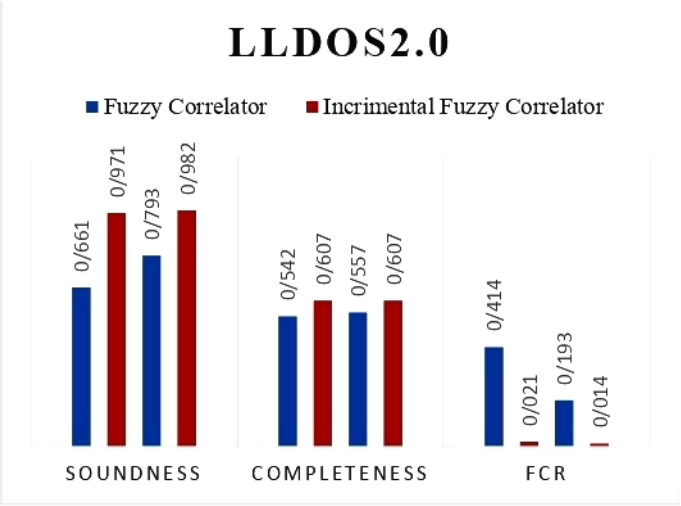


Figure 11: Accuracy assessment for the LLDoS2.0 scenario.

Expriment 2: Evaluation of the comprehensive model for incremental intrusion detection: **Comparison with the SC+ITI method,** in [28] proposed model with the SC-ITI method of incremental intrusion detection methods has been referenced at least 10 times. In their experiments, they have applied Sarasamma's training set. The training set contains 169,000 instances from the '10% KDD' dataset, and the test set contains 311,029 instances from the 'corrected KDD' dataset. They have applied two metrics, TPR and FPR, to evaluate their proposed IDS.

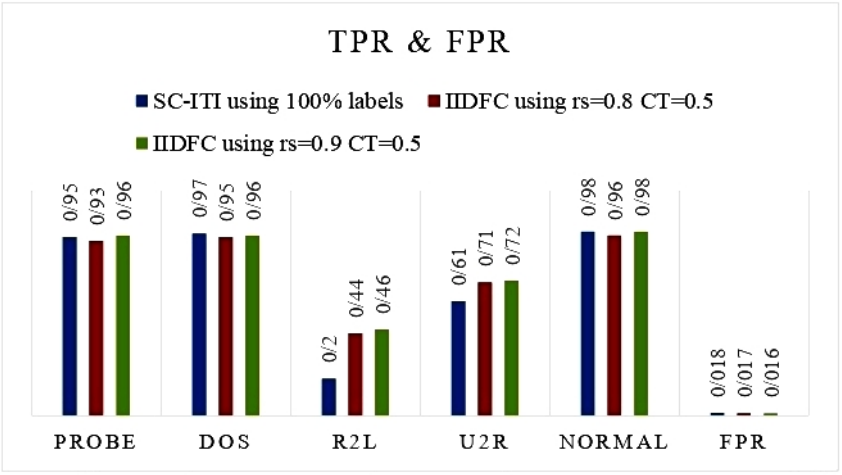


Figure 12: The true positive rates (TPRs) and false-negative rate of service classifier incremental tree inducer (SC-ITI) and incremental intrusion detection model using correlation (IIDMC) using different rule selection threshold (rs) .

As depicted in Figure 12, the TPRs of R2L and U2R attacks of IIDMC using 80% and 90% rule selection(rs) outperform SC-ITI using 100% labeled data significantly (for the other attacks, the

TPRs are nearly the same.) Notably, the average FPRs of SC-ITI method and ISF-NIDS using 15% and 100% labeled data are 0.018, 0.017, and 0.016, respectively.

Comparison with the incremental SVM method, The incremental support vector machine approach of incremental intrusion detection methods The KDD99 dataset has been tested, this article has been published in a non-security journal but has been cited at least 35 times in reputable security journals.
In [36], the authors select about 10% of the KDD99 dataset as the source dataset.

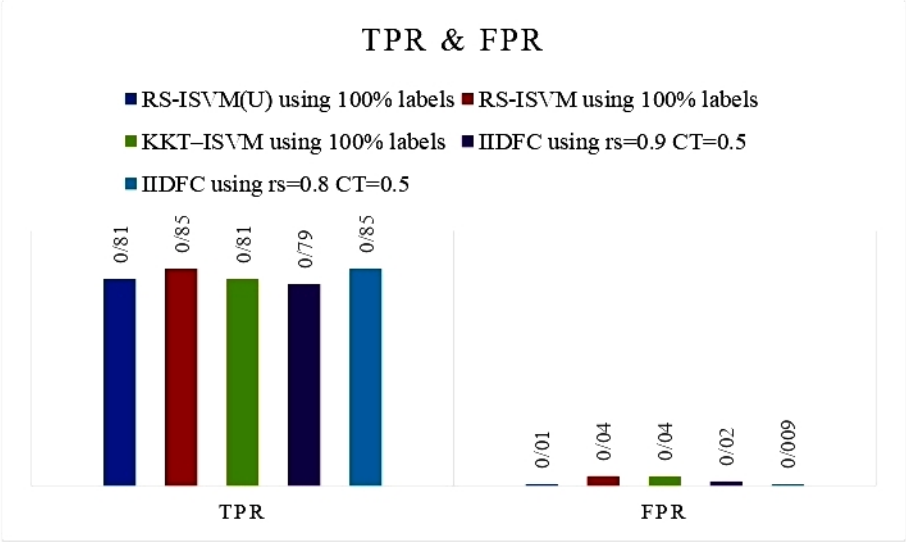


Figure 13: The true positive rates (TPRs) and false-negative rate of incremental support vector machine based on reserved set; KKT,Karush-Kuhn-Tucker and incremental intrusion detection model using correlation (IIDMC) using different rule selection threshold (rs) .

As shown in Figure 13, the best TPRs belong to IIDMC using $rs = 0.8$. The TPRs of IDFC using $rs = 0.9$ are approximately the same as that of RS-ISVM that has the highest TPR among the tested methods in [36]. Also, Figure 13 depicts FPRs of the tested methods. The FPRs of IDFC using $rs = 0.8$ are the same as the minimum FPR of the other tested methods in [36]. The minimum FPR belongs to IDFC using $rs = 0.8$.

5. Disscusion

The values of two important parameters of correlation threshold and correlation sensitivity were intuitively selected as 0.5 and 0.1, respectively, and subsequent studies confirmed these values. When two alarms are examined in the form of a similarity vector and their correlation probability is determined to be truly correlated, they must provide a minimum of correlation probability. If the probability of correlation between two alarms is less than 0.5, it does not seem reasonable to spend more time on it, so a correlation threshold value of 0.5 is chosen. In the same work, the same number is used for this parameter value. But when two correlated alarms were detected, their probability was greater than 0.5. All alarms that are in the same alert and have a probability of correlating less than 0.1 with the alert are also assumed to correlate with the values in the correlation matrices. This will allow the system to gain more experience from its current success and remember it for future correlations. If the parameter has a larger value, it expands the range of experience gained to more alarms.

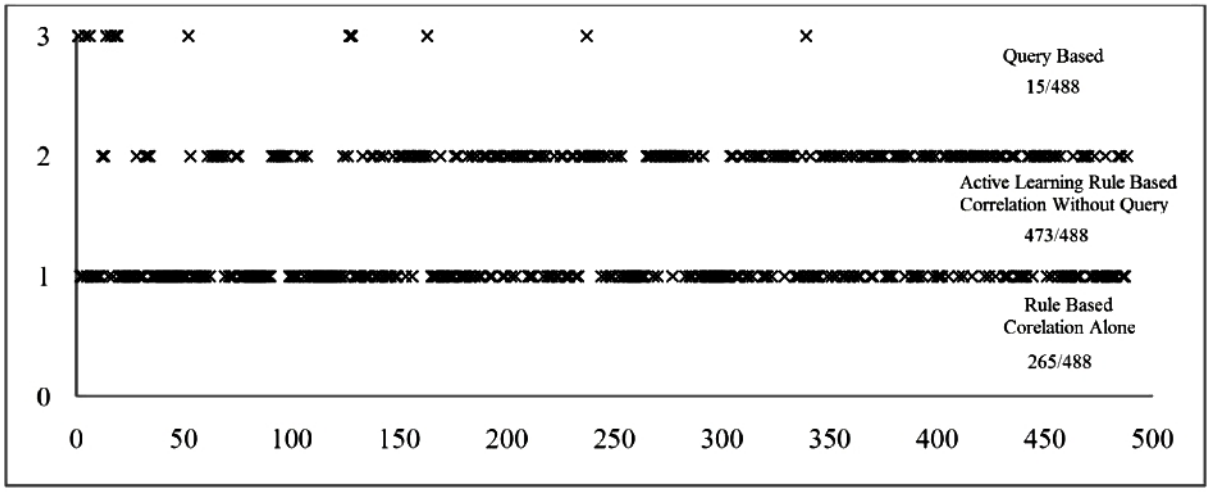


Figure 14: Number of correlated alarms by fuzzy engine, rule-based using active learning without query number and number of expert queries with rs=0.8 in LLDoS2.0 .

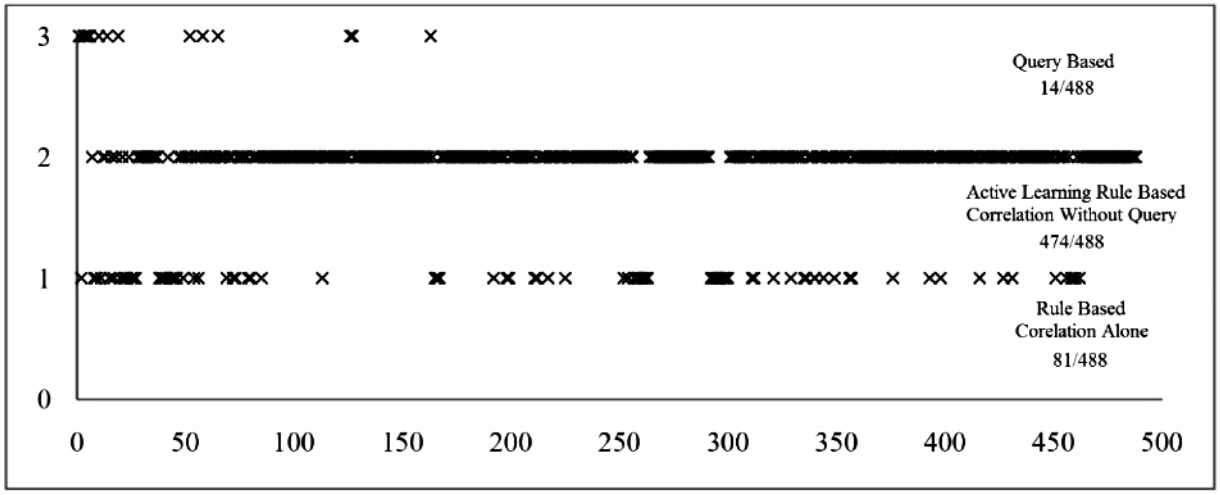


Figure 15: Number of correlated alarms by fuzzy engine, rule-based using active learning without query number and number of expert queries with rs=0.9 in LLDoS2.0 .

Figure 14 and 15 show the percentages of alarms that are correlated in each of one layer and two layers with different values of these two parameters. At $rs = 0.8$, the number of correlated alerts in the single-layer rule-based correlation is higher than when $rs = 0.9$, but in the dual-layer mode with (active learning), as shown in Figures 14 and 15, the accuracy increases appropriately, so this issue indicates that knowledge acquired appropriately leads to accuracy in future correlations.

6. Conclusions

The biggest weakness of intrusion detection systems is false positive generation. The security expert should examine the correlations of the generated alarms so that he can distinguish real alarms from false positives.This drastically increases the cost of the organization. This study provides a comprehensive intrusion detection model that can help security experts by detecting knowledge gained using active learning and online incremental learning techniques. The proposed model consists of two main parts, including the correlator and alarm generator system.

References

[1] A. A. Aburomman, M. B. IbneReaz, *A survey of intrusion detection systems based on ensemble and hybrid classifiers,* Computers & security, 65 (2017) 135-152.
[2] J. Akhtar-Khan and N. Jain, *A survey on intrusion detection systems and classification techniques*, International journal of scientific research in science, Engineering and technology, 2 (2016) 202-208.
[3] S.O. Al-Mamory and H.L. Zhang, *Building scenario graph using clustering*, Proceedings of the 2007 international conference on convergence information technology, IEEE computer society, (2007) 799-804.
[4] S.O. Al-Mamory and H.L. Zhang, *Scenario discovery using abstracted correlation graph*, International conference on computational intelligence and security, IEEE computer society, (2007) 702-706.
[5] J. P. Anderson, *Computer security threat monitoring and surveillance,* Technical report, James P. Anderson company, Fort Washington, 1980.
[6] R. A. R. Ashfaqand et al, *Fuzziness based semi-supervised learning approach for intrusion detection system*, Information sciences 0 0 0 (2016) 1-14.
[7] A. Chmielewskiand S. T. Wierzchon, *Hybrid negative selection approach for anomaly detection*, In computer information systems and industrial management, Springer Berlin Heidelberg, (2012) 242-253.
[8] S. Duque and et al, *Using data mining algorithms for developing a model for intrusion detection system (IDS)*, Procedia computer science, 61 (2015) 46-51.
[9] C. Guo, Y. Zhou, Y. Ping, S. Luo, Y. P. Lai and Z. Zhang, *Efficient intrusion detection using representative instances*, Computers and security, 39 (B) (2013) 255-267.
[10] K. K. Gupta, B. Nath and R. Kotagiri, *Layered approach using conditional random fields for intrusion detection*, IEEE trans., Dependable secur. comput. , 7 (1)(2010) 35-49.
[11] K. Gupta, S. Singhal, S. Malik and A. Singh, *Network intrusion detection system using various data mining techniques,* International conference on research advances in integrated navigation systems (RAINS), (May 2016) 6-7.
[12] T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: Data mining, inference and prediction*, Springer, 2001.
[13] K. Julisch, *Clustering intrusion detection alarms to support rootcause analysis*, ACM transactions on information and system security (TISSEC), 6 (4) (2003) 443e71.
[14] N. K. Kanakarajan and K. Muniasamy, *Improving the accuracy of intrusion detection using GAR-Forest with feature selection*, Proceedings of the 4^{th} international conference on frontiers in intelligent computing: theory and applications (FICTA) 2015 Springer, (2016) 539-547.
[15] KDD cup 1999 data,(1999), Available:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[16] B. Khosravifar, M. Gomrokchi and J .Bentahar, *A multi-agent based approach to improve intrusion detection systems false alarm ratio by using honeypot,* International conference on advanced information networking and applications workshops, (2009) 97-102.
[17] Laboratory ML, Darpa2000 intrusion detection scenario specific data sets, https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets.
[18] H. H. Lin, C. H. Mao and H. M. Lee, *False alarm reduction by weighted score-based rule adaptation through expert feedback,* At the second international workshop on multimedia, Information privacy and intelligent computing systems (MPIS), Jeju island, Korea, 2009.
[19] H. S. Lin, H. K. Pao, C. H. Mao, H. M. Lee, T. Chen and Y. J.Lee, *Adaptive alarm filtering by causal correlation consideration in intrusion detection,* First KES international symposium on intelligent decision technologies (IDT), 2009.
[20] Y. Liu and L. Zhu, *A new intrusion detection and alarm correlation technology based on neural network,* EURASIP Journal on Wireless communications and networking 2019, (2019) 109.
[21] P. Ning, Y. Cui and S. Reeves, *Constructing attack scenarios through correlation of intrusion alerts,* In proceedings of the 9^{th} ACM conference on computer and communications security, ACM, (2002) 245-254.
[22] P. Ning, Y. Cui, D. S. Reeves, *Techniques and tools for analyzing intrusion alerts,* ACM transactions on information and system security, 7 (2) (2004) 274-318.
[23] H. H. Pajouh, G. Dastghaibyfard and S. Hashemi, *Two-tier network anomaly detection model: a machine learning approach*, J Intell Inf Syst, (2015) 1-14.
[24] M. Panda, A. Abraham and M. R. Patra, *A hybrid intelligent approach for network intrusion detection*, Procedia engineering, 30 (2012) 1-9.
[25] T. Pietraszek, *Using adaptive alert classification to reduce false positives in intrusion detection,* Proceedings of the 7^{th} symposium on recent advances in intrusion detection (RAID), Springer-Verlag, 3224 (2004) 102-124.
[26] K. Polat, S. Gunes, *Principles component analysis, fuzzy weighting pre-processing and artificial immune recogni-*

tion system based diagnostic system for diagnosis of lung cancer, Expert systems with applications, 34 (1) (2008) 214-221.

[27] R. Sadoddin, A. A. Ghorbani, *An incremental frequent structure mining framework for real-time alert correlation,* Computers and security, 28 (3-4) (2009) 153-173.

[28] ST. Sarasamma, QA. Zhu, *Min-max hyper ellipsoidal clustering for anomaly detection in network security*, IEEE transactions on systems, Man and cybernetics, 36(4) (2006) 887-901.

[29] F. Shen, O. Hasegawa, *A fast nearest neighbor classifier based on self-organizing incremental neural network*, Neural networks, 21 (2008) 1537-1547.

[30] R. Shittu, A. Healing, R. Ghanea-Hercock, R. E. Bloomfield and M.Rajarajan, *Intrusion alert prioritisation and attack detection using post-correlation analysis,* Computers & security, 50 (2015) 1-15.

[31] G. Spathoulas and S. Katsikas, *Reducing false positives in intrusion detection systems,* Computers & security, 29 (1) (2010) 35-44.

[32] P. Srinivasu and P. S. Avadhani, *Genetic algorithm based weight extraction algorithm for rtificial neural network classifier in intrusion detection*, Procedia engineering, 38 (2012) 144-153.

[33] R. Vaarandi, K. Podins, *Network IDS alert classification with frequent itemset mining and data clustering*, CNSM 2010, (2010) 451-456.

[34] A. Valdes and K. Skinner, *Probabilistic alert correlation,* In proceedings of the 4^{th} international symposium on recent advances in intrusion detection, (2001) 54-68.

[35] F. Valeur, G. Vigna, C. Kruegel and R. Kemmerer, *Acomprehensive approach to intrusion detection alert correlation,* IEEE transactions on dependable and secure computing, 1(3) (2004) 146-169.

[36] Y. Yi, J. Wu, W. Xu, *Incremental SVM based on reserved set for network intrusion detection,* Expert systems with applications, 38(6) (2011) 7698-7707.

[37] J. Zhang, X. Chen, *Research on intrusion detection of database based on Rough set*, Physics procedia, 25 (2012) 1637-1641.

[38] B. Zhu, A. Ghorbani, *Alert correlation for extracting attack strategies*, International journal of network security, 3 (3) (2006) 244-258.