HRDID : A Hybrid Intrusion Detection System

H. Salama and M. Zaki

Computer Engineering Department, Al-Azhar Universty, Egypt
hmhsalama@yahoo.com , *mzaki.azhar@gmail.com*

Abstract

HRDID is a proposed intrusion detection classifier that works on the basis of a combination of evolutionary computing algorithms. That combination comprises of decision trees to provide fast recognition of predefined attacks and rough sets to generate, automatically, a set of rules capable to accomplish attack detection. In case of rule(s) non determinism, Bays probabilistic rules is relied upon to reach a proper decision. In addition HRDID uses a Hamming distance to allow for approximate reasoning.

HRDID architecture is given and its performance is analyzed. The proposed classifier is tested using a DARPA data set and its performance is evaluated in terms of accuracy, precision and ROC diagram. Actually, all the test results and comparisons have confirmed the system dependability and responsiveness.

Keywords: *Rough sets, Decision tree, Intrusion detection, Pattern classification, Bayes rule.*

1. Introduction

Intruders represent a real danger for computer systems. Therefore, enterprise are keen to obtain qualified Intrusion Detection Systems, IDSs, that are able to classify successfully normal and intrusive actions. Historically, traditional methods [1] have been used to accomplish intrusion detection, however, recently evolutionary computing and machine learning approaches [2] are exploited to find out both misuse and anomaly situations.

This paper utilizes a hybrid approach and presents HRDID as a hybridization of rough sets and decision trees for intrusion detection systems. Such IDS has unique features that are pointed out in the following:

i - Making use of both DTs to achieve quick response for previously known (predefined) attacks and RSs to precise realize response for sophisticated new attacks.

ii - The reduct of the underlying RS is used as a minimal subset of attributes that has the same capability of traffic classification as the entire set of attributes. On the basis of the obtained reducts, the detection rules are computed automatically.

iii - If the generated rules, from the RS classifier, are unable to afford a decision, a Baysian probabilistic rule is employed to mitigate rules non-determinism.

iv - Also, HRDID makes use of Hamming Distance, HD, for providing are approximate decision in case of no generated rules could match the input characteristics.

Several experiments are conducted to evaluate the performance of HRDID at various operational conditions. In addition its performance is computed with other ID evolutionary classifiers [3], [2]. Such experiments and comparisons rely on benchmark data from MIT Lincoln Labs that has been developed by DARPA as a defect standard for evolutionary intrusion detection systems. Actually, the obtained results and the conducted comparisons have confirmed the superiority of HRDID.

The rest of this paper is organized as follows. The related work is given in section 2 while section 3 presents the unique architecture of HRDID system. Its implementation is given in section 4 where the experimental results are reported and investigated. Eventually section 5 provides the paper conclusions.

2 . Related Work

There are many techniques that can be used in developing ID systems. Such systems make use of classifying the data packets to obtain different types of computer networks attacks. New evolutionary methods are used in building these classifiers. Such classifiers are based on fuzzy logic, support vector machines, artificial neural networks, decision trees or rough sets.

2.1 Fuzzy-Rules Classifiers

The prediction process of intrusion detection usually generates false alarms in many anomaly based intrusion detection systems. However, with fuzzy logic, the false alarm rate in determining intrusive activities can be reduced, where a set of fuzzy rules is used to define the normal and abnormal behavior in a computer network, and a fuzzy inference engine can be applied over such rules to determine intrusions [4]. In addition several efficient ID systems are proposed in [2].

2.2 Support Vector Machines (SVMs)

SVMs are learning machines that plot the training vectors in high-dimensional feature space, labeling each vector by its class. SVMs view the classification problem as a quadratic optimization problem. They combine generalization control with a technique to avoid the "curse of dimensionality" by placing an upper bound on the margin between the different classes, making it a practical tool for large and dynamic data sets. SVMs classify data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space [5]. They are based on the idea of structural risk minimization, which minimizes the generalization error, i.e. true error on unseen examples.

The number of free parameters used in the SVMs depends on the margin that separates the data points but not on the number of input features, thus SVMs do not require a reduction in the number of features in order to avoid over fitting [6]. Upon using SVMs [7] the ID system attack detection rate has reached 81.8% at FP equals 1 %.

2.3 The Artificial Neural Network (ANN)

Multi-layer preceptor (MLP) is a famous ANN module that has been used successfully in different classification problems. The network-based intrusion detection system (NBIDS) that uses ANN as MLP classifier. In such architecture, the neurons are organized in discrete layers and within each layer one neuron is not connected to another.

If every neuron in one layer is connected to every other neuron in the next layer, the layers are said to be fully connected. If some of the connections were missing, the network would be referred to as partially connected.

The main function of the ANN ID classifier is to classify the connections as normal or abnormal ones, and to identify the attacks types in case of abnormal connections. Basically, this system consists of two modules:

i - Features Preprocessing Module.

ii- Artificial Neural Network Classification Module. Which consists of three phases (ANN creation Phase- ANN training Phase- Classification phase).

The results of [8] have indicated that ANN based systems could reach 98.4%.

2.4 Use of Decision Trees

There are two types of variables involved in the decision tree techniques : target variable, T, and predictor variable(s),P. The value of the target variable for a data record often indicates the class of this data record (called target class) . The value of predicator variables in a data record are used to predict the value of the target variable for this data record or classify this data record.

A decision tree classifier is depending on the calculation of the entropy to each path to choose the rout with the highest entropy, in the next data example we choose some attributes of sample network data with a sample of attributes to clarify the process of entropy calculation.

The entropy function H is given by:

$$H = - \sum p_i \log p_i \qquad (1)$$

Where, p_i is the probability that the system is being in the i-th state. By making use of entropy calculations the algorithm chooses the best path that locates more information notes closer to the root of the tree. Several DT based algorithms [9], [10] are used to detect network intrusions. The error rates of these systems range from 0.1% to 20% depending on the selected features.

All the above proposals confirm the ability of self computing and evolutionary algorithm for building a dependable ID system. In addition, by making use of hybridization such ID systems can be considerably improved regarding accuracy, precision and response. In section 3 a hybrid system, that utilizes the speed of DT's and the capability of rough sets the work with vagueness, is proposed.

3. The Proposed System

The proposed ID System, as such, combines more than one classification method to realize both efficiency and speed. The system is given as a hybrid IDS that exploits decision trees to provide quick reorganization of known attacks and rough sets to classify more sophisticated attacks.

The proposed hybrid system utilizes a Bayesian approach to recognize the new attacks. In addition a Hamming distance measure is used to make use of imprecision detection.

3.1 Confusion Matrix and System Parameters

Any classifier must be examined for comparing with others and to measure its efficiency, Using some unified parameters we can examine and evaluate the classifier, so using the confusion matrix to figure out the classifier efficiency.

The system parameters are calculated on the basis of confusion matrix. Such as, Figure (1) explains an example in which the classifier classifies DoS and Land Attack. The inputs contain 5 known DoS Attacks, 9 known Land Attacks and 9 unknown attacks, Figure (1).

Predicted / Classified	DoS	Land	Non
DoS	3	1	1
Land	2	5	2
Non	1	1	7

Figure(1). An example of a confusion matrix

In our case we have three classes, and consequently a 3x3 confusion matrix.

$$\text{Accuracy} = (TP + TN) / (TP + FP + FN + TN) \quad (2)$$

$$\text{Precision} = (TP) / (TP + FP) \quad (3)$$

$$\text{Recall} = (TP) / (TP + FN) \quad (4)$$

$$\text{True Negative Rate} = (TN) / (TN + FP) \quad (5)$$

Where, TP, TN, FP and FN are true detected attacks, true detected non attacks, non attacks detected as attacks and attacks detected as non attacks, respectively.

Table 1: Results obtained from the confusion matrix

	Accuracy	Precision	Recall	TNR
DoS	0.782	0.5	0.6	0.4
Land	0.739	0.714	0.555	0.444
Non	0.782	0.7	0.777	0.286

3.2 Decision Tree Classifier

Decision tree can be pruned where a sample of a pruned decision tree is shown in Figure (2).

```
Count <= 1
|   srv_count <= 1
|   |   serror_rate <= 1
|   |   |   srv_serror_rate <= 1
|   |   |   |   protocol_type = tcp : DoS
|   |   |   |   protocol_type = udp: Land
|   |   |   |
```

Figure (2) Sample of a pruned decision tree

The decision tree classifier algorithm can be described as in Figure (2) and the procedure is clarified as follows:

Procedure DT (*count, srv_count, serror_rate, srv_serror_rate, protocol_type ,decision*)

1 - Choose an attack attribute that best differentiates the unerlying attribute values, according to equation (1)

2 - Create a separate tree branch for each value of the chosen attribute.

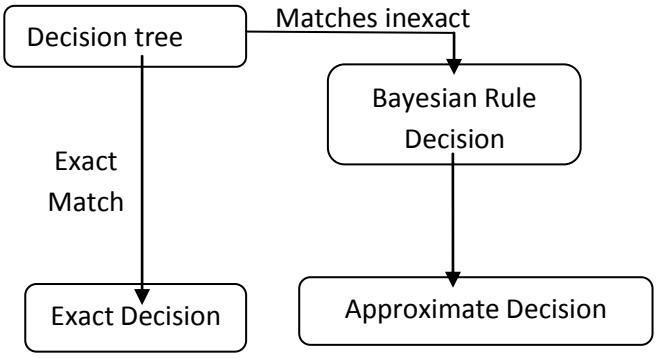
3 - Divide the instances into subgroups so as to reflect the attribute values of the chosen node.

4 - For each subgroup, terminate the attribute selection process if:

(a) All members of a subgroup have the same value for the output attribute, then terminate the attribute selection process for the current path and label the branch on the current path with the specified value.

(b) The subgroup contains a single node or no further distinguishing attributes can be determined. As in (a), label the branch with the output value seen by the majority of remaining instances.

5 - For each subgroup created in (3) that has not been labeled as terminal, repeat the above process.



Figure(3). Decision tree approach

3.4 Rough Set Classifier

The rough set [11] model Figure(4) uses rules which are created from the sample data [12]. These rules may be deterministic i.e. any rule refers only to one decision, and a non deterministic rule which means that the rule refers to many possible decisions. No matched rule(s) means there is no alike rule in the rough set classifier model. Accordingly, rough set model must be enhanced using probability equation to figure out the approximate decision depending on this probability algorithm. Thus using the Bayesian rule to calculate the probability ratio enhances the RS classification model. Adding the Hamming Distance Algorithm also enhances the model for the case of no rules matched.

Procedure Rough set (count, srv_count, serror_rate, srv_serror_rate, protocol_type ,decision)

1 – Compare the first atom attribute value and select all the rules that have the same value of the first atom attribute.

2 – Then compare the second atom attribute, if any, and select the values that have the same value also, and so on, till the algorithm match a rule or more than one rule.

3 – If the algorithm continue till we have any result of the next four probabilities:

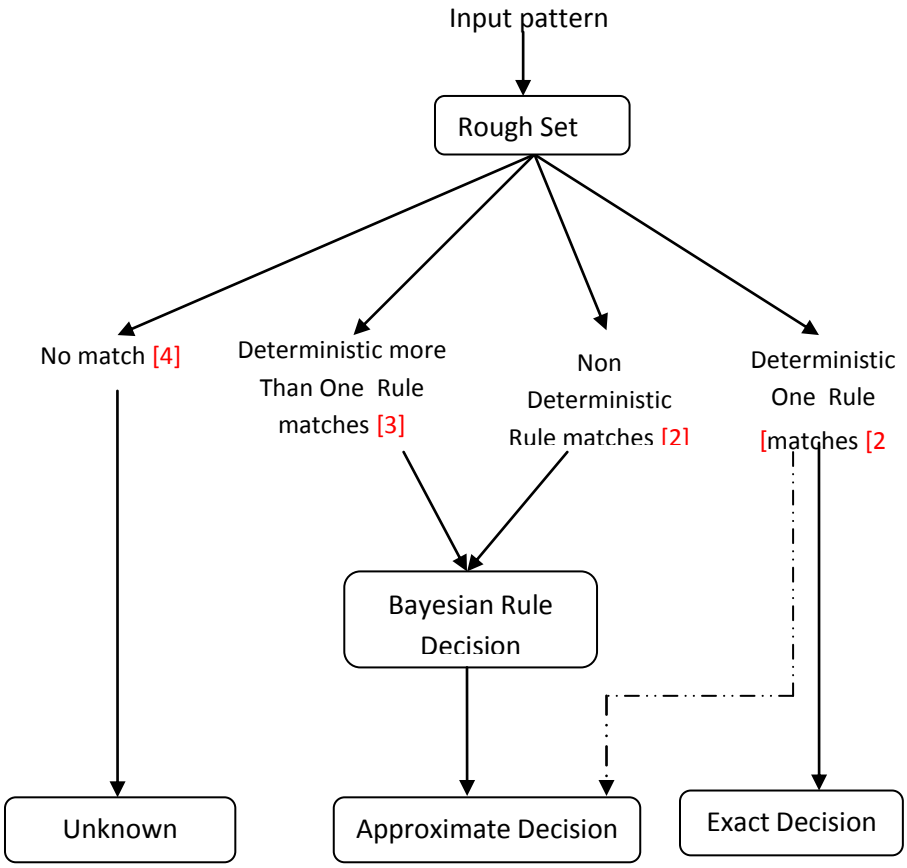
(a) If the message matches only one deterministic rule, then the decision is exact extracted.

(b) If the message does not match any rule, then the system will use the Hamming distance to get the closest rule to figure out the approximate decision is extracted.

(c) If the message matches a non deterministic rule or:

(d) Matches more than one deterministic rule, then using the Bayesian algorithm to figure out the approximate decision.

4 – Then the system will create reports of the cases of (b), (c) and (d) for the administrator of the network who decides the behavior of these messages.



Figure(4). Rough set model

3.4 Hybrid Model

As discussed in this chapter the decision tree classifier is a reliable classifier for some case and the rough set classifier is a reliable for others, Mixing them in one hybrid solution give the chance to achieve most benefits of their united properties.

The proposed hybrid IDS Figure (5) works as follows.

If an input message reached the system interface, then the system examines it using the decision tree classifier. If that message is a previous known attack the system will take quickly the exact decision and figures out the attack correctly, path 1, Figure (5). Thus when the message contains attributes values can be examined through the pruned tree that until reaching an accurate leave. Thus the system, accurately detected the attack, otherwise the message will be checked using the rough set classifier.

Such rough set classifier which checks the messages into 4 different possibilities as follows:

If the message matched one deterministic rule, Then the system determines an exact decision and determines the attack correctly, path2, Figure (5).

If the message matched non deterministic rule, or matched more than one deterministic rule, then the system will check using the Bayesian rule to afford a decision to figure out the most probable attack In an approximate decision mode, paths 3 and 4.

If the message does not match any rule, then the system uses the Hamming distance measure to classify this message into one of three possibilities, path 5.

If the message matches only one distance rule so using the decision tree to classify the message into two cases, path 6, If it matches a decision, Then the system figures out an approximate decision, path 7.

But if matches no decision the system gives unknown attack, path8. If the message matches more than one distance rule so the Bayesian rule is used to provide a decision to figure out the most probable attack, path 9.

If the message matches no paths the system gives unknown attack, path 10. The hybrid model procedure is given in what follows:

Procedure Hybrid ***(count, srv_count, serror_rate, srv_serror_rate, protocol_type ,decision)***

1 – The hybrid system is a mix of the previous two systems, such that the steps of the first algorithm "Decision tree algorithm", so this algorithm could detect an attack of the input message using its levels and leaves.

2 – If the message is not a known attack, then the decision tree will not detect a decision for the input message, then the second algorithm runs the rough set algorithm steps in order to figure out the input message, and then to classify it into the four mentioned cases according to the rules which are created.

3 – If the hybrid system detected the attack from the first algorithm "decision tree algorithm" or the second one "rough set algorithm", the system will deal with this attack immediately.

4 – But if the first algorithm does not detect the message as an attack and the second algorithm classifies the message into one of the following cases:
(b) "the message does not match any rule".
(c) "the message matches non deterministic rule".
(d) "the message matches more than deterministic rule".

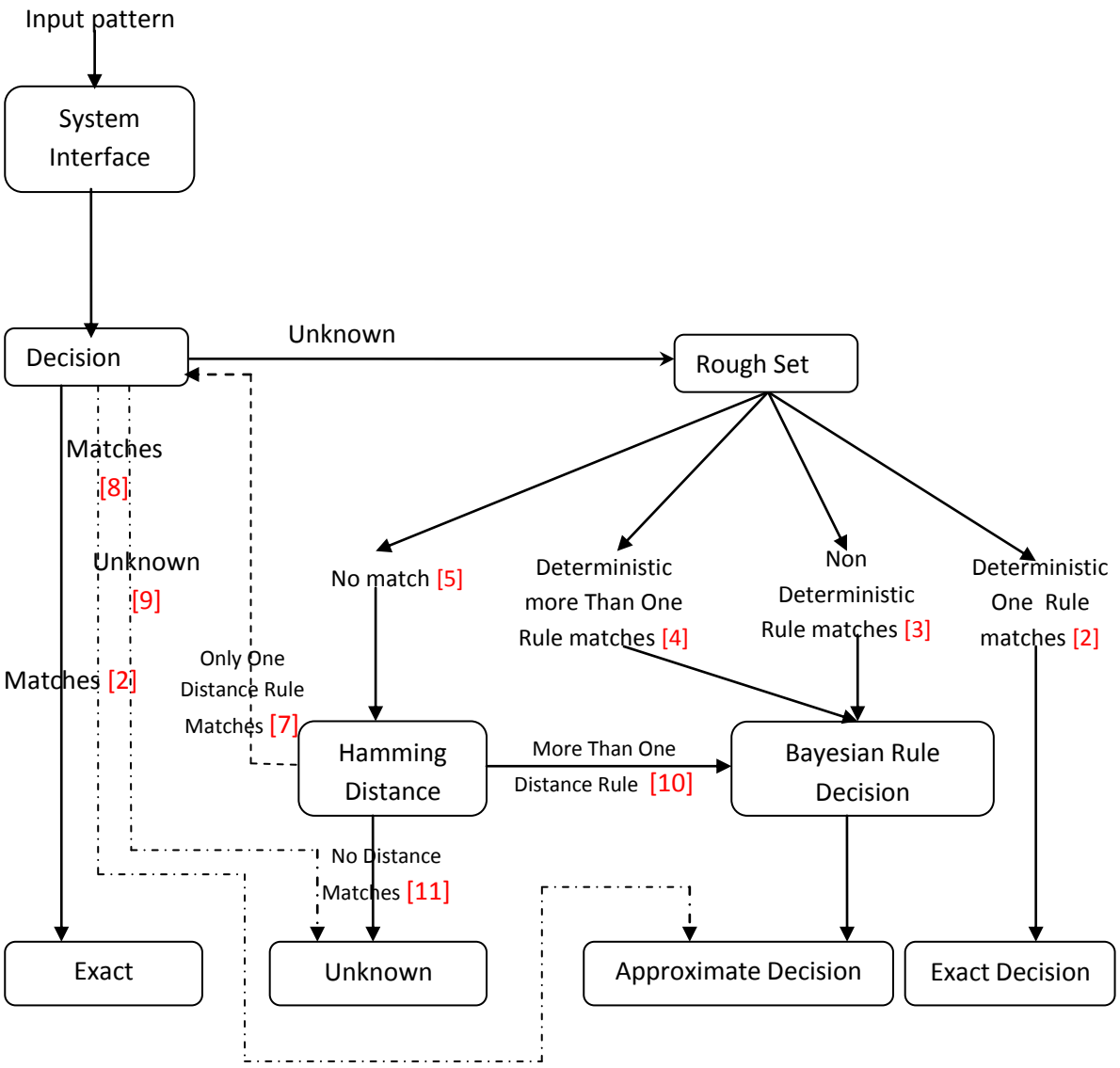


Figure (5). The proposed hybrid model

4. Implementation

Realistic data samples, from DARPA data set are generated and the properties of each sample are described in Table 2.

Table (2) Sample Data

data sample 1	20% of the sample messages are randomly selected from known 80% are randomly created (30% of them have full descriptors)
data sample 2	40% of the sample messages are randomly selected from known 60% are randomly created (30% of them have full descriptors)
data sample 3	60% of the sample messages are randomly selected from known 40% are randomly created (30% of them have full descriptors)
data sample 4	80% of the sample messages are randomly selected from known 20% are randomly created (30% of them have full descriptors)
data sample 5	20% of the sample messages are randomly selected from known 80% are randomly created (70% of them have full descriptors)
data sample 6	40% of the sample messages are randomly selected from known 60% are randomly created (70% of them have full descriptors)
data sample 7	60% of the sample messages are randomly selected from known 40% are randomly created (70% of them have full descriptors)
data sample 8	80% of the sample messages are randomly selected from known 20% are randomly created (70% of them have full descriptors)

It is noticed that in Table (2) the data samples are randomly selected. In all cases they contain either 70% full attack descriptors or 30% full attack descriptors.

The hybrid classifier uses both the decision tree and the rough set classifiers. Its basic procedure is summarized by the following steps.

S1: Examining the input message by the pruned decision tree at first. If the message matches an attack, then the system will announce that attack, else the control is switched to the rough set algorithm.

S2: Examining the message using the rough set algorithm using the five mentioned steps (in chapter 4), If the rough set testing recognizes an attack (exact or approximate decision), then the system will announce that attack.

S3: If the system gives unknown as a decision in S2, then the system should use the Hamming distance algorithm which calculates the distances between the input message attributes and the whole rough set rules. If the best distance matches only one rule then we test this rule by making use of S1. In this case the decision "is approximate.

S4: If the best Hamming distance matches more than one attack in S3, then we use the Bayesian probability to provide an approximate decision. In case of no matching the system announces unknown decision.

By making use of equation (2) one can obtain the accuracy of DT, RS and hybrid classifiers, Figure(6). In addition the precision and recall of such classifiers are given in Figures (7) and (8), respectively. The TNRs are compared for the three classifiers in Figure(9).

Moreover, the Receiver Operating Characteristics, ROC, of the proposed ID system is calculated, Figure(10), to provide the classifier false positives (cost) versus its true positives (benefit). A comparison of such cost benefit relationship of the proposed classifier with that of both the fuzzy (only) classifier [2] and SVM classifier [7] emphasizes the superiority of assembling together both rough sets and decision trees.

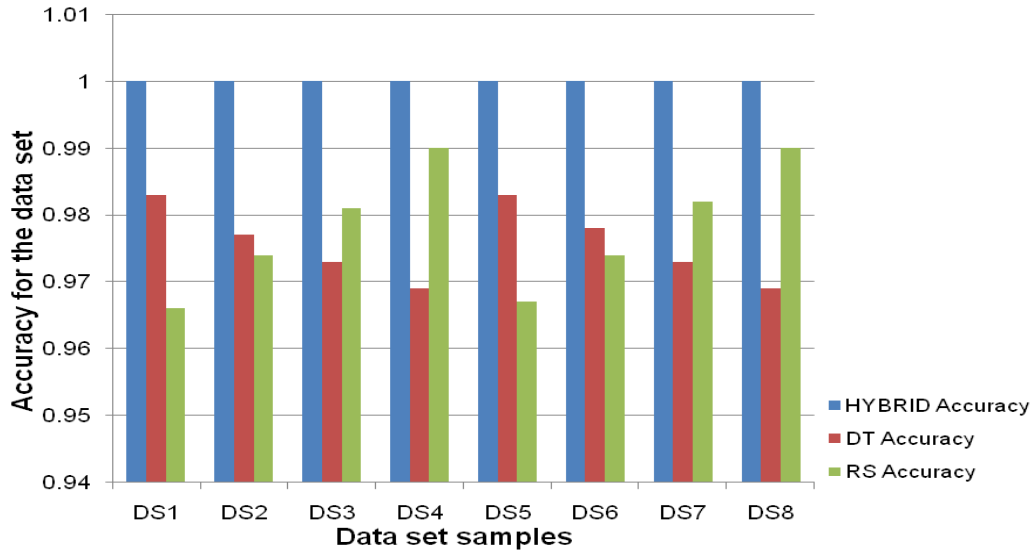


Figure (6) Accuracy of the DT, RS and hybrid classifiers for the all data sets

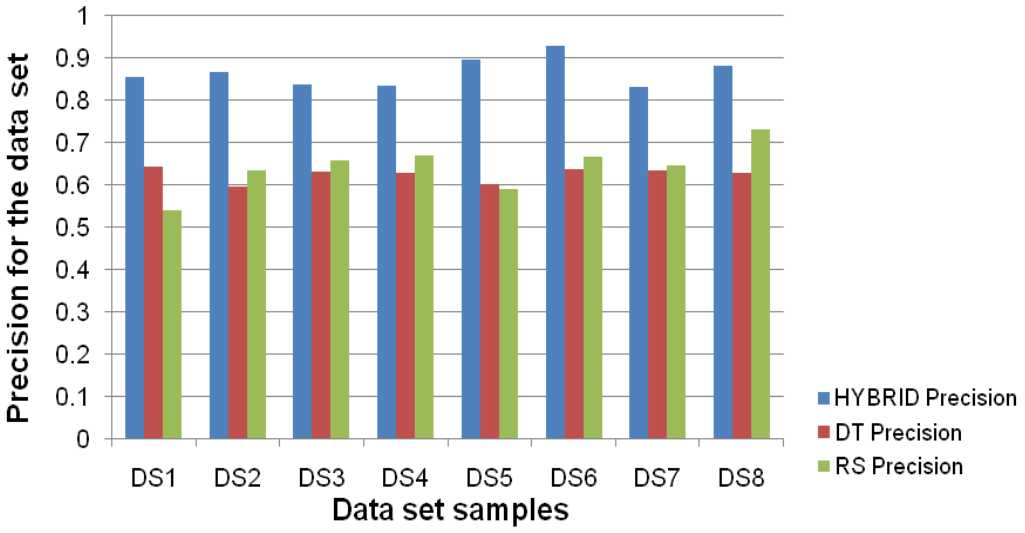


Figure (7) Precision of the DT, RS and hybrid classifiers for all the data sets

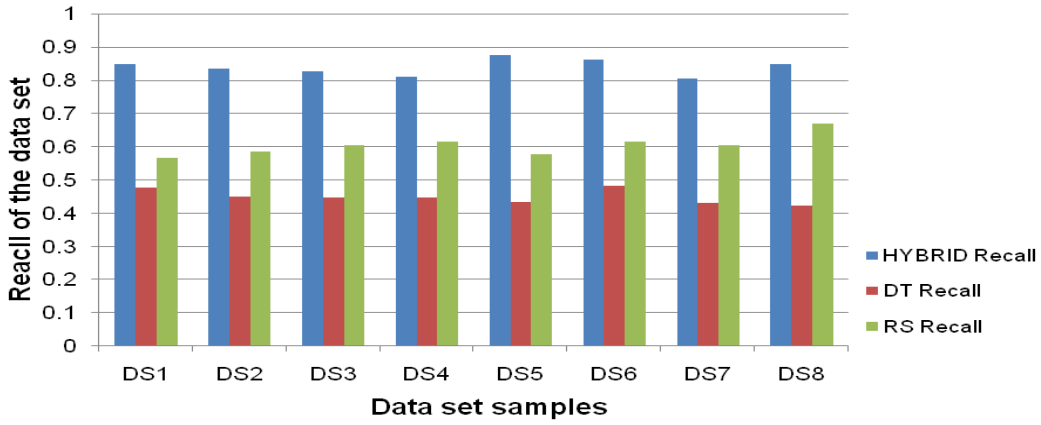


Figure (8) Recall of the DT, RS and hybrid classifiers for all the data sets

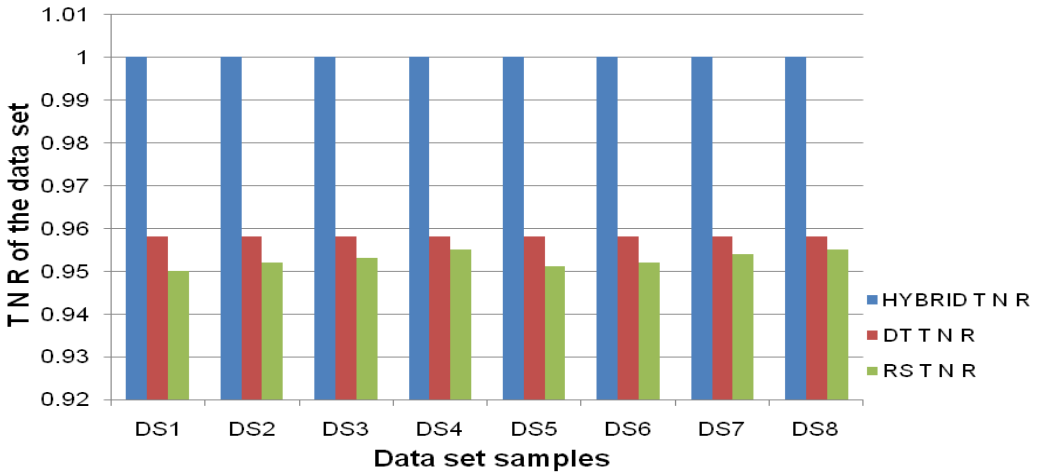


Figure (9) T N R of the DT, RS and hybrid classifiers for all the data sets

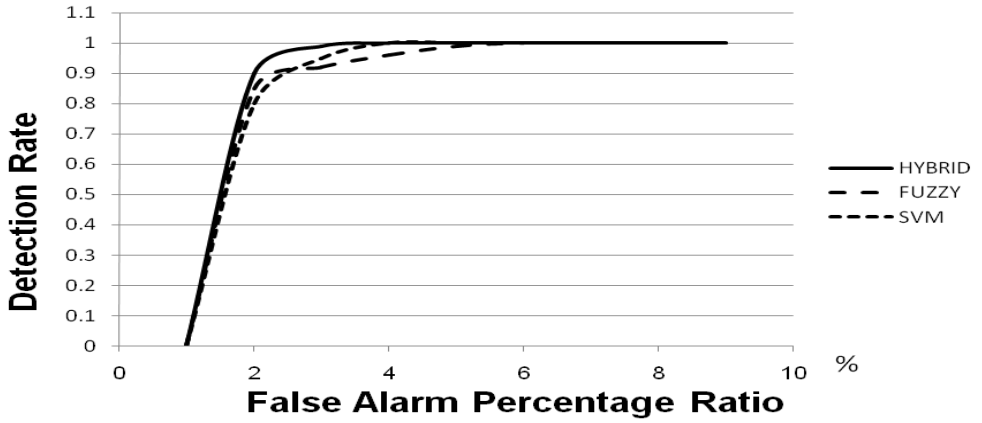


Figure (10) Detection rate against the false alarm percentage "ROC Diagram"

5. Conclusion

This paper presents HRDID as a hybrid IDS that has been built on the basis of a combination of DT, to discover quickly predefined attacks, and RS to deal with sophisticated unknown attacks. HRDID consists of:

i - DT module: accordingly the most informative node(s) is located closest to the root.

ii - RS module: in which the reduct of the data attributes has the same classification capabilities as whole set of traffic attributes.

iii - Baysian module: that exploits Bayes probabilistic rule to overcome the non determinism of the rules generated by RS, if any.

iv - HD module: to provide approximate reasoning within a tolerance measured by an arbitrary HD.

HRDID is tested using DARPA data set and its performance is analyzed. Its performance as a classifier is evaluated in terms of accuracy, precision and recall. Moreover its ROC is obtained and compared with ROC's of previously known ID classifiers. All the results have verified the proposed system dependability and have confirmed its superiority.

References

[1] H Debar, M Dacier and A Wespi, Towards a Taxonomy of Intrusion-Detection Systems, Computer Networks, Volume 31, Issue 8, Pages 805-822, 1999.

[2] J Gomez and D Dasgupta, Evolving Fuzzy Classifiers for Intrusion Detection, Proceedings of the IEEE, Pages 168-176, 2002.

[3] L Zhang, G Zhang, YU Lang, J Zhang and Y Bai, Intrusion detection using rough set classification, Journal of Zhejiang University Science, Volume 5, Issue 9, Pages 1076-1086, 2004.

[4] R Shanmugavadivu and N Nagarajan , Network Intrusion Detection System Using Fuzzy logic, Indian Journal of Computer Science and Engineering (IJCSE), Volume 2, Issue 1, Pages 101-111, 2011.

[5] V Vladimir, The Nature of Statistical Learning Theory, Springer, Berlin, 2000

[6] T Joachims, Making Large-Scale SVM Learning Practical. LS8-Report, University of Dortmund, LS VIII-Report, 1998.

[7] W Hu, Liao and V. Rao Vemuri, Robust Support Vector Machines for Anomaly Detection in Computer Security, http://wwwcsif.cs.ucdavis.edu/~vemuri/papers/rvsm.pdf

[8] H Allouni, An Intrusion Detection Approach to Computer Networks, M.Sc thesis, Military Technical College, 2003

[9] Bouckaert R, Frank E,--- ,WEKA Manual for Ver --,Date

[10] G Stein, B Chen, A S.Wu and K A.Hua, Decision Tree Classifier For Network Intrusion Detection With GA-based Feature Selection, ACM-SE, Volume 43, Issue 2, 2005.

[11] Z Pawlak, Rough sets. International Journal of Computer and Information Sciences, Volume 11, Pages 341-356, 1982.

[12] B. Walczak and D. Massart, Rough sets Theory, Chemometrics and Intelligent Laboratory Systems, Pages 1-16, 1999.