A Proposal to Detect Computer Worms (Malicious Codes) Using Data Mining Classification Algorithms

Dr.Soukaena Hassan Hashim
Computer Science Department, University of Technology/ Baghdad
Email:soukaena_hassan@yahoo.com
Inas Ali Abdulmunem
Computer Science Department, College of Science, University of Baghdad/ Baghdad

Received on: 15/10/2012 & Accepted on: 10/1/2013

ABSTRACT

Malicious software (malware) performs a malicious function that compromising a computer system's security. Many methods have been developed to improve the security of the computer system resources, among them the use of firewall, encryption, and Intrusion Detection System (IDS). IDS can detect newly unrecognized attack attempt and raising an early alarm to inform the system about this suspicious intrusion attempt. This paper proposed a hybrid IDS for detection intrusion, especially malware, with considering network packet and host features. The hybrid IDS designed using Data Mining (DM) classification methods that for its ability to detect new, previously unseen intrusions accurately and automatically. It uses both anomaly and misuse detection techniques using two DM classifiers (Interactive Dichotomizer 3 (ID3) classifier and Naïve Bayesian (NB) Classifier) to verify the validity of the proposed system in term of accuracy rate. A proposed HybD dataset used in training and testing the hybrid IDS. Feature selection is used to consider the intrinsic features in classification decision, this accomplished by using three different measures: Association rules (AR) method, ReliefF measure, and Gain Ratio (GR) measure. NB classifier with AR method given the most accurate classification results (99%) with false positive (FP) rate (0%) and false negative (FN) rate (1%).

Keywords: Malware, Intrusion detection system, Hybrid, Data mining.

مقترح لكشف ديدان الحاسوب (البرمجيات الخبيثة) باستخدام خوارزميات التصنيف لتنقيب البيانات

الخلاصة

البرمجيات الخبيثة (malware) تؤدي وظيفة خبيثة و التي تساوم أمن نظام الحاسوب .وقد تم تطوير طرق عديدة لتحسين أمن موارد نظام الحاسوب، من بينها استخدام جدار الحماية، التشفير، ونظام كشف التطفل (IDS). IDS يمكن أن يكشف محاولة هجوم غير مميزة حديثا و يرفع إنذار مبكر لإعلام النظام

حول محاولة التطفل المشكوك بها. هذا البحث اقترح IDS هجين لكشف التطفل، والبرمجيات الخبيثة خطةً ، مع الاخذ بنظر الاعتبار ميزات حزمة الشبكة والمضيف. IDS الهجين صمم باستخدام طرق التصنيف لتنقيب البيانات (DM) و ذلك لقدرتها لاكتشاف تطفلات جديدة لم تشاهد مسبقا بدقة وبشكل تلقائي . هو يستخدم كل من تقنيتي الشذوذ وكشف سوء الاستخدام باستخدام اثنين من مصنفات DM (مصنف ID3 (3 Interactive Dichotomizer) و مصنف النظرية الافتراضية البسيطة (NB)) للتحقق من صحة النظام المقترح بدلالة نسبة الدقة. مجموعة بيانات HybD مقترحة استخدمت في تدريب واختبار IDS الهجين. استخدم اختيار الميزة للاخذ بنظر الاعتبار الميزات الجوهرية في قرار التصنيف، هذا انجز باستخدام ثلاثة مقاييس مختلفة : طريقة قواعد الارتباط (AR)، مقياس ReliefF، ومقياس نسبة المكسب (GR). مصنف NB مع طريقة AR اعطى نتائج التصنيف الأكثر دقة (99٪) مع نسبة ايجابية كاذبة (FP) (0%) و نسبة سلبية كاذبة (FN) (1%).

INTRODUCTION

With the rapid expansion of computer systems during the recent years, and the large developments in new technologies in this domain, the important data are under constant threats of intrusion. All those make the security a critical issue for modern computer systems [1]. Malicious software (malware) is software that is intentionally included or inserted in a system for a harmful (malicious) purpose. Malware is the most sophisticated type of threats to computer systems that exploit vulnerabilities in computing systems [2]. Current antivirus systems attempt to detect these new malicious programs with heuristics generated by hand. This approach is costly and oftentimes ineffective [3].

An intrusion is any set of deliberate, unauthorized inappropriate, and/or illegal activity by perpetrators either inside or outside a system, which can be deemed a system penetration, that attempt to compromise the integrity, confidentiality or availability of a resource [4][5]. Intrusion detection (ID) is a technique of monitoring systems for evidence of intrusions or inappropriate usage. The detection of intrusions either manually or via software expert systems that operate on logs or other information available from the system or the network. ID is an important component of infrastructure protection mechanisms. It is an important component of infrastructure protection mechanisms [6] and it analyzes the occurring events in the aim to identify intrusive behavior and establish a response plan [7]. An IDS is a security mechanism that monitors and analyzes system events to provide real-time warnings to unauthorized access to system resources or to archive log and traffic information for later analysis [5][8].

IDS can be classified according to IDS's environment as: a network-based IDS (NIDS) that is a dedicated computer, or special hardware platform, with detection software installed that captures packets in a promiscuous mode [9], or as a host-based IDS (HIDS) that monitors the resource usage of the operating system (OS) and the network. HIDS can only monitor the resource usage of the applications and not the applications themselves [10].

Data Mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner [11]. Classification is a method of categorizing or assigning class labels to a pattern set under the supervision of a teacher (i.e. learning). Decision boundaries are generated to discriminate between patterns belonging to different classes. The datasets are initially partitioned into training and test

sets, and the classifier, which is a construct (algorithm) that discriminate between classes of patterns, is trained on the training set to create a model. The test set is used to evaluate the generalization capability of the classifier. The derived model may be constructed using one of various methods, such as 1) decision tree which is one of the most widely used supervised learning methods used for data exploration. It is easy to interpret and can be re-represented as If-then-else rules [12], 2) Statistical methods which work under the assumption that the underlying pattern generating mechanism is faithfully represented by a statistical model [12]. The Bayes decision theory provides a framework for statistical methods for classifying patterns into classes based on probabilities of patterns and their features. The goal of classification, based on Bayesian decision theory, is to classify objects based on statistical information about objects in such a way as to minimize the probability of misclassification [13], and etc.

The remaining part of this paper organized as follows: next section presents ID with DM, and then paper presents previous researches that related to IDS and their limits. The methodology in building a hybrid IDS will be describes. Finally, a discussion about the resulted proposed IDS and the obtained results presented followed by conclusions and suggestions for future work.

INTRUSION DETECTION WITH DATA MINING

DM based ID techniques generally fall into two main categories: *misuse detection* and *anomaly detection*. In misuse detection systems, use patterns of well-known attacks to match and identify known intrusion. These techniques are able to automatically retrain ID models on different input data that include new types of attacks, as long as they have been labeled appropriately. Unlike signature-based IDSs, models of misuse are created automatically, and can be more sophisticated and precise than manually created signatures. A key advantage of misuse detection techniques is their high degree of accuracy in detecting known attacks and their variations. Misuse detection techniques in general are not effective against novel attacks that have no matched rules or patterns yet. Anomaly detection, on the other hand, builds models of normal behavior, and flags observed activities that deviate significantly from the established normal usage profiles as anomalies, that is, possible intrusions. Anomaly detection techniques thus identify new types of intrusions as deviations from normal usage. Anomaly detection techniques can be effective against unknown or novel attacks since no *a priori* knowledge about specific intrusions is required. However, anomaly detection systems tend to generate more false alarms than misuse detection systems because an anomaly can just be a new normal behavior. Some IDSs use both anomaly and misuse detection techniques [14].

DM framework detects new, previously unseen intrusions accurately and automatically. The DM framework automatically found patterns in the used dataset and uses these patterns to detect a set of new intrusions. By comparing detection methods used DM with a traditional signature based methods; see that, DM based detection methods are more than doubles the current detection rates for new malwares [3].

RELATED WORKS

In [15] Al-Janabi S. et al proposed an anomaly based IDS that can promptly detect and classify various attacks. Anomaly-based IDSs need to be able to learn the dynamically changing behavior of users or systems. The proposed IDS experimenting with packet behavior as parameters in anomaly ID. There are several methods to assist

IDSs to learn system's behavior, the proposed IDS uses a back propagation artificial neural network (ANN) to learn system's behavior and uses the KDD CUP'99 data set in its experiments. In [16] Bensefia H. et al propose a new approach for IDS adaptability by oriented toward Evolving COnnectionist Systems (ECOS) and Learning Classifier Systems (LCS). These two learning machine approaches are actually suggested very suitable to build adaptive learning intelligent systems in a dynamic changing environment. This integration puts in relief an adaptive hybrid ID core that plants the adaptability as an intrinsic and native functionality in the IDS. In [17] Haldar N. et al presented an IDS which employs usage of classification methods to model the usage patterns of authenticated users and uses it to detect intrusions in wireless networks. The key idea behind the proposed IDS is the identification of discriminative features from user's activity data and use them to identify intrusions in wireless networks.

THE PROPOSED INTRUSION DETECTION SYSTEM

The proposed IDS is a "hybrid IDS" (NIDS and HIDS) that because it consider all features of data network packets and consider critical features of host that are directly affected by malwares. The proposal is a DM-based IDS in which both the misuse and anomaly detection techniques depended in the detection of intrusion, where each instance in a dataset is labeled as "normal" or "intrusion" and a learning algorithm is trained over the labeled data. Misuse technique is able to automatically retrain ID models on different input data that include new types of attacks, as long as they have been labeled appropriately. While anomaly technique should first learn the characteristics of normal activities and abnormal activities of the system, and then the IDS detect traffic that deviate from normal activities.

For training and testing of the proposed IDS a proposed dataset, named "HybD", will be used. HybD dataset composed of: 1) "KDD'99 dataset" which represents the most widely used dataset for the evaluation of ID methods since 1999. This dataset is prepared by Stolfo et al. and is built based on the data captured in DARPA'98 IDS evaluation program. 2) Host-based features combined with the KDD'99 dataset. This HybD dataset could be used in researches for designing NIDSs, HIDSs, and hybrid IDSs. This new features are related to host and are used in conjunction with the 41 features in order to be able to detect intrusion in host level as well as in network level.

The design and the implementation of the proposed hybrid IDS, as depicted with flowchart in Figure (1), will be according to the following consequence steps:

- Preprocessing with the HybD dataset
- Feature selection
- Classifiers building
- and, the classification (testing)

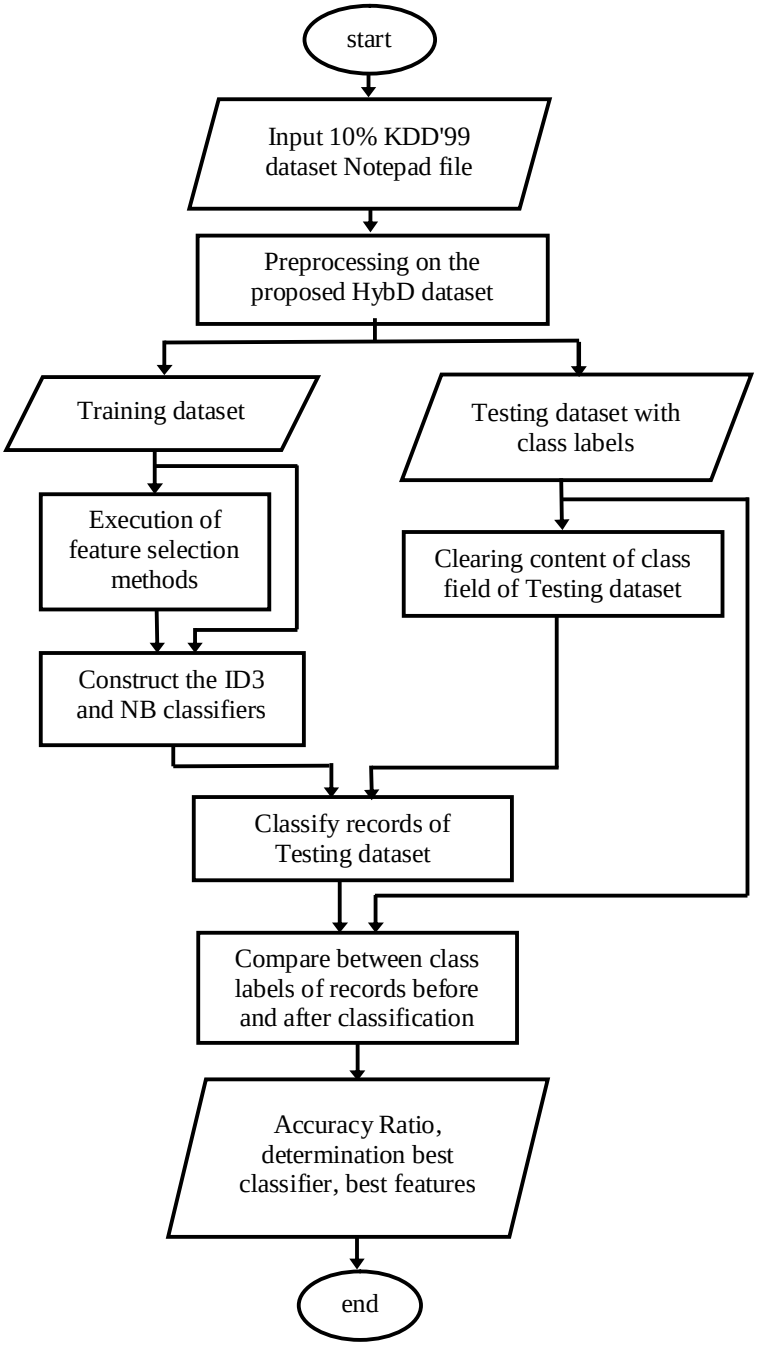


Figure (1) Flowchart of the proposed hybrid IDS Dataset Description

The 10percent KDD'99 Dataset.

DARPA'98 is about 4 gigabytes of compressed raw (binary) training data of 7 weeks of network traffic. The 2 weeks of test data have around 2 million connection records.

KDD'99 training dataset consists of approximately 5 million connection records (a connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a destination IP address under some well-defined protocol) each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. The simulated attacks fall in one of the following four categories: Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), and Probing Attack. KDD'99 features can be classified into three groups: Basic features, Content features, and Traffic features.

THE NEW HOST-BASED FEATURES

The proposed HybD dataset includes the aforementioned 41 features and the new added host-based features. Each category of attacks has different effects on a host, e.g., (DoS attack makes some computing and memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine), thus a different host-based features have to be added for each category to ensure the precision of detection of the different attack types. In this proposed dataset only some of host-based features that related to DoS attack category will be considered and added, they are cpu usage ratio, memory space ratio, and kernel space ratio. They are the most features affected by the DoS attacks. Only the connection records of "DoS attacks" and "normal" will be used in both training and testing of the classifiers to be designed.

PREPROCESSING ON THE PROPOSED HYBD DATASET

The following processes have been applied to the "proposed HybD dataset" before it being used in design of the proposed system:

1. Converting the *original KDD'99 10percent dataset* from a text file to a Microsoft Access table.
2. Connection records selection from the KDD CUP'99 10percent dataset table resulted from process one. The selected connection records labeled with either "normal" or "one of DoS attacks" (except "land attack").
3. Elimination of all duplicated connection records from the dataset resulted from process two.
4. Adding of new host-based features to construct the proposed HybD dataset and adding their values.
5. Since type of some of HybD dataset's features is continuous, thus a process for normalization these features have been done in order to become of categorical type so it becomes more convenient with the used DM classification algorithms, and also this process will simplify the execution of the A priori algorithm.
6. The resulted dataset from process 5 will be split into two distinct datasets by using, one for classifiers' training which equal $^{2}/_{3}$ of resulted dataset from process five and the other for classifiers testing which equal $^{1}/_{3}$ of resulted dataset from process five.

FEATURE SELECTION

This is an essential process to reduce, if possible, number of features and select the most intrinsic of these features in the classification decision, and hence to minimize the computation time of implementing the classification algorithms and so of the proposed

system. It has been accomplished with three techniques from different fields: ARs from DM which is applied to the training dataset to find the frequent patterns (using A priori algorithm), ReliefF measure from distance measures that try to find the most relevance features, and GR from information theory that selects features with the highest GR value. Thus three sets of features in addition to set of all features in the *training dataset* will be used in design (learning) of the proposed classifiers.

Algorithm (1) Customized_Apriori
Input: TrainD training dataset, min_sup minimum support threshold, and *NI* number of iterations
Output: *FRF* Set of most frequent and related features
Steps:
1. Construct itemsets *IS* from *TrainD*
2. Find all items in *IS*, and put them in set of items *IT*
3. Copy *IT* into itemset *IT2*
4. For i:1 to *NI*
5. Find all frequent items from *IT2*, , and put them in *FI*
6. Empty *IT2*
7. Construct new items from *FI* and *IT*, and put them in *IT2*
8. From *FI* extract *FRF*
9. End

Algorithm (2) Proposed ReliefF
Input: *TrainD* training dataset
Output: *FWZ* set of features with weight greater than zero
Steps:
1. For each feature in *TrainD*, initialize its weight to zero
2. For i:1 to number of records in *TrainD*
3. For record *R*, find its nearest hit *H* of same class and its nearest miss *M* from different class
4. Update feature's *weight* = *weight* – *sqr(diff(R,H))* + *sqr(diff(R,M))*
5. For each feature with *weight* > zero add it to *FWZ*
6. End

Algorithm (3) Gain Ratio
Input: either *TrainD* training dataset (when algorithm used as feature selection measure)
or TrainDN training dataset node (when algorithm used with Decision Tree classifier)
both of them will referred in this algorithm as *D* Dataset
Output: *GRS* set of GR values for each feature in *(D)*
Steps:
1. For each feature in *(D)*
2. Find its InfoGain
3. Find its Split Information

4. If Split Information = 0
   then set it to very small value(<0)
5. Find its *GR* and add it to *GRS*
6. End

CUSTOMIZED ID3 AND NB CLASSIFIERS

After the intrinsic features had been selected, the two popular DM classification algorithms: ID3 from Decision Tree field and Naïve Bayesian from Bayesian theorem field, used in the design of the proposed IDS.

A "Decision Tree classifier" is one of the most widely used supervised learning methods used for data exploration. It is easy to interpret and can be re-represented as *If-then-else* rules. This classifier works well on noisy data. On the other hand, various empirical studies of "Bayesian classifier" in comparison to Decision Tree and ANN classifiers have found it to be comparable in some domains. In theory, Bayesian classifiers have the minimum error rate in comparison to all other classifiers. Bayesian classifiers are also useful in that they provide a theoretical justification for other classifiers that do not explicitly use Bayes' theorem. For example, under certain assumptions, it can be shown that many ANN algorithms output the *maximum posteriori* hypothesis, as does the NB classifier. NB classifiers assume that the effect of a feature value on a given class is independent of the values of the other features. This assumption is called *class conditional independence*. Both of ID3 and NB classifiers will be used 4 times with each of these 4 sets of features to design the proposed classifiers:

1. All 44 features of training dataset.
2. Subset of features in (1) according to the result of implementation of the *AR* method.
3. Subset of features in (1) according to the result of implementation of the ReliefF measure.
4. Subset of features in (1) according to the result of implementation of the *GR* measure.

Thus, eight classifiers will be obtained: four ID3 classifiers and four NB classifiers. Then the classification of the testing dataset's records will be done with each one of these classifiers, and a comparison among their classification results will be done in order to specify the most accurate classifier among them.

ID3 CLASSIFIERS

With ID3 classifier, a decision tree has been constructed starting with *training dataset* as the "root node" of the tree, then split it into several sub-datasets nodes according to the feature with the "highest GR" value and the columns with this feature in each of these nodes will be removed, and the split process will be repeated with these new nodes. Before deciding to split each new constructed node, computing:

- the "number of classes" in it,
- its "initial entropy",
- specify the "selected and used features",
- and, the "*InfoGain*" and "*GR*" of each feature in the node.

The splitting continues until either all records in the node are labeled with the same class or there is no feature to split the node according to feature's values. After splitting

stage has been stopped, a set of "*top-down"* paths will be constructed from the root node of the tree to each leaf node in it. A path consists of a series of feature-value pairs ending with a class label. This set of paths examined to discover and delete the duplicated once. Then these paths converted to "if-then-else" rules (i.e. classification rules) which will be used then to classify records of the testing dataset.

Algorithm (4) The ID3 [13]
Input: A set of training examples, *S*.
Output: A decision tree.
Steps:

1. Create the root node containing the entire set *S*
2. If all examples are positive, or negative, then stop: decision tree has one node.
   Otherwise (the general case).
3. Select feature F_j that has the largest *GR* value
4. For each value v_i from the domain of feature F_j:
5. add a new branch corresponding to this best feature value v_i, and a new node, which stores all the examples that have value v_i for feature F_j
6. if the node stores examples belonging to one class only, then it becomes a leaf node, otherwise below this node add a new subtree, and go to step 3
7. End

NAÏVE BAYESIAN CLASSIFIERS

In NB classifier a set of probabilities (a priori, conditional, and posteriori) has been found instead of constructing a set of classification rules. Firstly, compute the "a priori probability" of each class (i.e. the frequency of each class in the training dataset). The a priori probability computed just once time for the whole training dataset. Then the following computations will be performed for classifying each record in the testing dataset. The conditional probability $P(a_j|C)$ for every feature's value in the record of the testing dataset is estimated as the relative frequency of records having value a_j as the jth feature in class C. Assuming conditional independence of features, the "conditional *probabilities*" $P(X|C_i)$ of the testing record at each class is computed using equation (1). Finally, the "postpriori probability" P(h|*X*) of the testing record at each class computed using equation (2), the class with the maximum postpriori probability h_{MAP} will be the label for the testing record according to equation (3).

$$P(X|C_i) = \prod_{j=1}^{n} P(a_j|C_i) \quad (1)$$

$$P(h|X) = \frac{P(X|h)P(h)}{P(X)} \quad (2)$$

$$h_{MAP} \equiv arg\ max_{h \in H} = arg\ max_{h \in H} P(X|h)P(h) \quad (3)$$

Algorithm(5) Naive Bayesian
Input: *TrainD* training dataset, *TestD* testing dataset that has not been classified
Output: *TestD* testing dataset that has been classified
Steps:

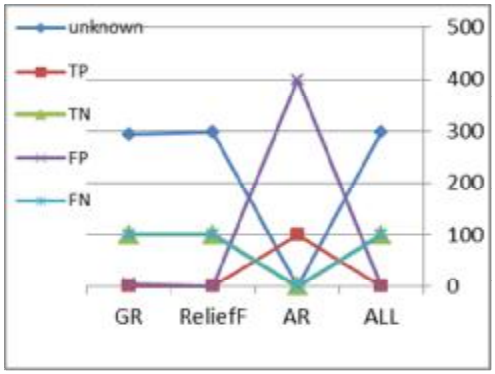
1. Initialize *MaxValue* to a small value
2. For each class C_i in *TrainD* find its a priori probability AP_i
3. For each record *R* in *TestD* do step 4 and 8
4. For each class C_i in *TrainD* repeat steps 5-7
5. Find the conditional probability CP_i of *R* at C_i using equation(2.9)
6. Find the postpriori probability $PP_i$of *R* using equation (2.5)
7. If PP_i greater than *MaxValue*
   then *MaxValue* = PP_i and *class_label* = C_i
8. Assign *class_label* to the class of *R*
9. End

DISCUSSION AND EXPERIMENTS

The proposal has been implemented on the following platform: Windows 7 Ultimate Service Pack1 and 32-bit OS, 4GB RAM, and Intel® Core (TM) 2 Duo CPU with 2.00GHz; and by using Visual Basic 6.0 programming language and Microsoft Access 2003.

After training the two chosen classifiers (ID3 and NB) on training dataset, the two classification models are constructed. Then apply these two models on *testing dataset* records to verify the validation and accuracy of constructed models. The classification results of testing are either true positive (TP) i.e. normal, true negative (TN) i.e. intrusion, false positive (FP) i.e. not normal, false negative (FN) i.e. not intrusion, or unknown i.e. new attack or user behavior.

The results obtained by testing ID3 classifiers which are of classifying the testing dataset records by using the implemented ID3 classifiers, have been illustrated in Figure (2). This results show that the unknown, FP, FN, TP and TN results are conflicting with each other, with very low results of both TP and TN results. While the classification results for classifying the same *testing dataset*'s records with NB classifiers, illustrated in Figure (3), are showing that: TN results are greater than FP, FN and unknown results when using all features set and subsets of features selected by AR, ReliefF, and GR measures, and the TP result with is greater than FP, FN and unknown result with four cases except in two case where FP result with ReliefF measure is greater than TP result with ReliefF and GR measures. Table (1) summarizes these results.



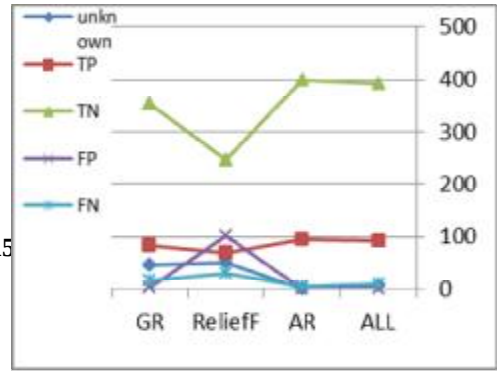


Figure (2) Classification Results of ID3 Classifiers.

Figure (3) Classification Results of NB Classifier.

The correct (TP+TN) and misclassification (FP+FN+unknown) results for each ID3 and NB classifier illustrated in Figure (4) and Figure (5). It is quite obvious from these two Figures that NB classifiers are better than ID3 classifiers since the NB classifiers' correct results are much greater than ID3 classifiers' correct results (a comparison between the classification accuracy of these two sets of classifiers illustrated later in Figure (6)). With ID3 classifiers see Figure (4), misclassification ratios is greater than correct classification ratios for all classifiers, while the inverse for these ratios with NB classifiers as it is depicted in Figure (5). Table (2) summarizes these results.

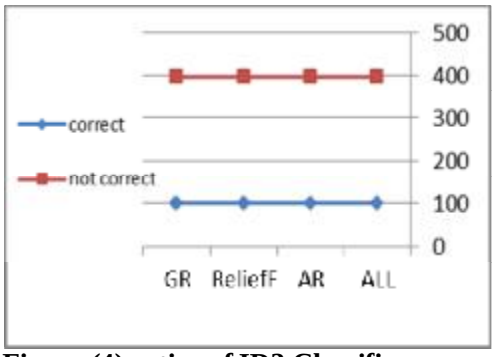


Figure (4) ratios of ID3 Classifiers.

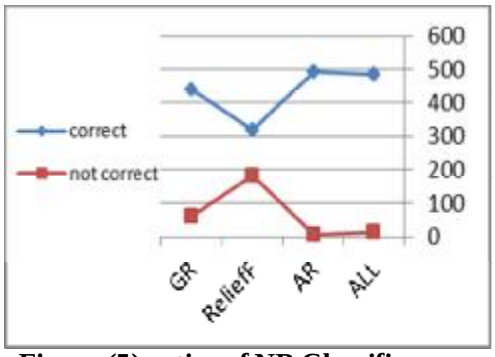


Figure (5) ratios of NB Classifiers.

The accuracy (Acc) of each classifier, see Figure (6), is computed as follows:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN + unknown} \quad \ldots (4)$$

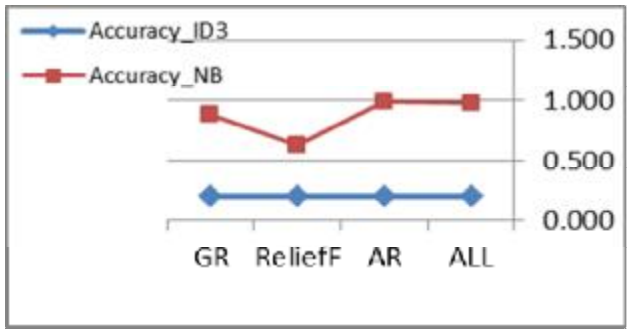


Figure (6) Accuracy ratios of ID3 and NB Classifiers.

Table (3) summarizes *Acc* for both ID3and NB classifiers with the four cases. According to these results, the most accurate classifier is NB classifier with AR measure with Acc is 0.990 followed by NB classifier with all features with Acc is 0.970. Where Table (4) presents the accuracy of classification when using AR method as feature selection measure with different: number of features, min_sup values, and number of iterations.

Table (1) Classification Results of ID3 and NB Classifiers.

Classifier	Feature selection measure	unknown	TP	TN	FP	FN
ID3	All	0.597	0.000	0.200	0.002	0.200
	AR	0.000	0.200	0.000	0.800	0.000
	ReliefF	0.599	0.000	0.200	0.000	0.200
	GR	0.587	0.000	0.200	0.012	0.200
NB	All	0.012	0.182	0.788	0.000	0.018
	AR	0.000	0.190	0.800	0.000	0.010
	ReliefF	0.100	0.140	0.495	0.204	0.060
	GR	0.090	0.168	0.707	0.002	0.032

Table (2) Correct and Misclassification ratios of ID3 and NB Classifiers.

Classifier	Feature selection measure	correct	Not correct
ID3	All	0.200	0.800
	AR	0.200	0.800
	ReliefF	0.200	0.800
	GR	0.200	0.800
NB	All	0.970	0.030
	AR	0.990	0.010
	ReliefF	0.635	0.365
	GR	0.876	0.124

Table (3) Accuracy Ratios of ID3 and NB Classifiers.

	ALL	AR	ReliefF	GR
Accuracy_ID3	0.200	0.200	0.200	0.200
Accuracy_NB	0.970	0.990	0.635	0.876

Table (4) AR measure: NB and ID3 accuracy ratios, and no_feature withdifferent min_sup and no_iteration.

Min-sup	No-iteration	No-feature	NB-accuracy	ID3-accuracy

1/60	3	23	0.982	0.062
	4	23	0.982	0.062
	5	23	0.982	0.062
	6	23	0.966	0.03
1/53	3	17	0.99	0.2
	4	17	0.99	0.2
	5	16	0.92	0.2
1/48	3	16	0.92	0.2
	4	16	0.92	0.2
1/40	3	16	0.92	0.2

CONCLUSIONS AND FUTURE WORKS

CONCLUSIONS

- Constructing HybD dataset to include new features, where these features related to host itself not to network traffic give a proposed view for proposed hybrid IDS to detect malware, since these codes have clear affect on host resources performance.
- Dividing HybD dataset into two sub datasets one for training and other for testing, with concentrate on making these two datasets have different records that help significantly in avoid of overfitting problem.
- In the training dataset, the number of records for each class (attacks and normal) equals that of other classes. This is to avoid bias of classification decision to a class with highest records' number.
- As it is obvious from Figure (5), NB classifiers are better than ID3 classifiers since the NB classifiers' accuracies are almost better than ID3 classifiers' accuracies. This is because NB classifier computes the probability for the whole dataset without regard to its volume and the number of features that were used. While with ID3 classifier the classification rules are restricted to permanent set of feature-value pair (s) with one class.
- Deciding AR parameter, (minimum support, number of iteration, and the selected features), was critical, they had been enhanced many times before they took their final value, see Table (4).

FUTURE WORKS

- The proposed hybrid IDS can be employed in servers and critical nodes on computer networks.
- The proposed hybrid IDS can be run online, that will need some modifications on the source code of sniffing program in order to be able to work with 44 features of the HybD dataset.
- Expand the IDS's capability by adding host-based features related to the U2R, R2L and Probing attack categories.

REFERENCES

[1]. Naser M., "A Honey Pot Resources Approach to Divert System Intruders" MSC Thesis, Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, 2006.
[2].Stallings W., "Cryptography And Network Security Principles And Practice" Prentice Hall, Pearson, 2011.
[3]. Schultz M., Eskin E., Zadok E.,"Data Mining Methods for Detection of New Malicious Executables", Security and Privacy, 2001, Proceedings.2001 IEEE, 2001.
[4]. Han J., Kamber M., "Data Mining: Concepts and Techniques" Morgan Kaufmaan Publishers, 2006.
[5]. Naief A., "Proposed System for Intrusion Detection in a Web Site" MSC Thesis, Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, 2006.
[6]. Krutz R., Conley J., Reisman B., Ruebush M., Gollman D., Reese R., "Network Security Fundamentals" John Wiley & Sons, Inc., 2008.
[7]. Kozushko H., "Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems" Independent Study, 2003.
[8]. Gollmann D., "Computer Security" WILEY A John Wiley and Sons, Ltd., Publication, 2011.
[9]. Al-Ajealee S., "Agent-Based Intrusion Detection System" MSC Thesis, University of Technology Department of Computer Science, 2005.
[10]. Rehman R. U., "Intrusion Detection Systems with Snort: Advanced IDS Techniques with Snort, Apache, MySQL, PHP, and ACID" Pearson Education, Inc. Publishing as Prentice Hall PTR, 2003.
[11]. LAROSE D., "Discovering Knowledge In Data An Introduction to Data Mining" WILEY INTERSCIENCE A JOHN WILEY & SONS, INC., PUBLICATION, 2005.
[12]. MITRA S., ACHARYA T., "Data Mining - Multimedia, Soft Computing, and BIOINFORMATICS" A JOHN WILEY & SONS, INC., PUBLICATION, 2003.
[13]. Cios K., Pedrycz W., Swiniarski R., Kurgan L., "Data Mining A Knowledge Discovery Approach" Springer, 2007.
[14]. Garuba M., Liu C., Fraites D., "Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems",IEEE Computer Society, Fifth International Conference on Information Technology, pp. 592-598, 2008.
[15]. Al-Janabi S. Saeed H., "A Neural Network Based Anomaly Intrusion Detection System" IEEE Computer Society, 2011 Developments in E-systems Engineering, pp. 221-226 2011.
[16]. Bensefia H. Ghoualmi N., "A New Approach for Adaptive Intrusion Detection" 2011 Seventh International Conference on Computational Intelligence and Security, pp. 983-987, 2011.
[17]. Haldar N., Abulaish M., Pasha S., "An Activity Pattern Based Wireless Intrusion Detection System" IEEE Computer Society, 2012 Ninth International Conference on Information Technology- New Generations, pp. 846-847, 2012.