Proposed Integrated Wire/Wireless Network Intrusion Detection System

Soukaena Hassan Hashem[1]

[1]Computer Science Department, University of Technology, Baghdad-Iraq

e-mail: soukaena.hassan@yahoo.com

Received: 22/10 /2013

Accepted**:** 20/2/2014

***Abstract* -** This research proposes “Integrated Network Intrusion Detection System (INIDS)” which is NIDS for wire/wireless networks. INIDN consider features of the three layers; transport and Internet layers for wire and data link layer for wireless. The proposal is a Data Mining (*DM)-based INIDS*, which trained over a labeled wire and wireless datasets (each transaction labeled normal, intrusion name or unknown), INIDS is a hybrid IDS (anomaly and misuse). INIDS, train and construct two separated proposed models these are, Wire-NIDS and Wireless-NIDS then integrate the two models to build the final INIDS. Wire-NIDS use *NSL-KDD dataset*; use Principle Component Analysis (PCA) as a feature extraction, and use Support Vector Machine (SVM) with Artificial Neural Network (ANN) as classifiers. Wireless-NIDS use proposed Wdataset *dataset*, use Gain Ratio (GR) as feature selection, and use Naïve Bayesian (NB) as a classifier. The results obtained from executing the proposed INIDS model showing that Wire-NIDS and Wireless-NIDS classifier accuracy and detection rate is generally higher with the subset of features obtained by PCA (8 from 41) and GR (8 from 17) than with all sets of features. Proposed confusion matrix of INIDS gives less confusion in detection rates with reduced features.

Keywords: IDS, SVM, ANN, NB, PCA, and GR.

1. Introduction

An intrusion can be defined as “any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource”. Also, 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[1].

In addition to Wire networks also WLANs suffer from a lot of vulnerabilities, some of these vulnerabilities are inherited from the usual wired networks and some are new due to the broadcast connection medium. These vulnerabilities include confidentiality, integrity and availability vulnerabilities. Through the WLAN evolution, many security improvements have been added to the IEEE 802.11 standards such as: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA) and IEEE 802.11i (WPA2). These techniques can only protect data frames to satisfy the confidentiality and the integrity security issues. The management and control frames are still unprotected [2]. Intrusion detection (ID) is a technique of monitoring systems for evidence of intrusions or inappropriate usage. The detection of intrusions is an important component of infrastructure protection mechanisms and it analyzes the occurring events in the aim to identify intrusive behavior and establish a response plan [3]. IDS can be classified according to IDS’s environment as a network-based IDS (NIDS) that is a dedicated computer, special hardware platform, with detection software installed that captures packets in a promiscuous mode, 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 [4]. Intrusion detection techniques are classified into two broad categories: misuse detection and anomaly detection. Misuse detection works by searching for the traces or patterns of well-known attacks. Clearly, only known attacks that leave characteristic traces can be detected that way. Anomaly detection, on the other hand, uses a model of normal user or system behavior and ages significant deviations from this model as potentially malicious. This model of normal user or system behavior is commonly known as the user or system profile. Strength of anomaly detection is its ability to detect previously unknown attacks [5].

2. Data Mining and Intrusion Detection

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. Classification is a method of categorizing or assigning class labels to a pattern set under the supervision of a teacher (i.e. learning). Classification 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 constructor (algorithm) that discriminates 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 [6].

A "*Support Vector Machine Classifier*" SVM is a useful technique for data classification. Even though it looks that

Neural Networks are simpler to use than SVM, however, sometimes wrong results are gained. A classification task usually includes training and testing data which consist of some data examples. Each example in the training set contains one base values and several attributes. The goal of SVM is to introduce a model which predicts base value of the data example in the testing set which is given only the attributes. Classification in SVM is supervised learning. Known typecast help indicate whether the system is performing in a right way or not. This information points to a coveted response, validating the precise of the system, or be used to help the system learn to do correctly. In theory, Bayesian classifiers have the lower error rate in comparison to all other classifiers. Bayesian classifiers are also useful in that they supply a theoretical warrant 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 hypothesis is called *class conditional independence* [7].

Feature selection is intended to suggest which features are more important for the prediction, to find out and get rid of irrelevant features that reduce classification accuracy, discover relations between features and throw out highly correlated features which are redundant for prediction. This function can be broken into two groups; feature extraction or feature transformation, and feature selection. Feature extraction refers to the process of creating a new set of combined features (which are combinations of the original features). Principle Component Analysis (PCA) is a useful statistical technique for feature extraction that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension. On the other hand, feature selection is different from feature extraction because it does not produce new variables. Feature selection, also known as variable selection, feature reduction, attribute selection, feature ranking, or variable subset selection, is a widely used dimensionality reduction technique, which has been the focus of much research in machine learning and data mining. Gain Ratio (GR) from information theory is a useful technique for selecting features with the highest GR value [6].

3. Related work

In ***[8], Ahmad I. et al*** proposed a Genetic Algorithm (GA) to search the principal feature space. GA and Multilayer Perceptron (MLP) are used for classification purpose. Consequently, this method provides optimal intrusion detection mechanism which is capable to minimize the number of features and maximize the detection rates. The goal in ***[9] Suebsing A. et al*** work is to effectively apply Euclidean Distance for selecting a subset of robust features using smaller storage space and getting higher Intrusion detection performance. Experimental results show that the proposed approach based on the Euclidean Distance can improve the performance of a true positive intrusion detection rate especially for detecting known attack patterns. In ***[10],***

Vaarandi R. proposed a novel unsupervised DM based approach for IDS alert classification. With this strategy, knowledge is mined from IDS logs and processed in an automated way, in order to build a caution classifier. The classifier is then used in real-time for discerning important IDS warns from frequently occurring false positives and events of low significance. In ***[11], Vaarandi R. et al.*** extended their previous work (Risto, 2009) on IDS alert classification. Their method first applies a frequent item-set mining algorithm to past IDS alert logs in order to discover patterns that describe redundant alerts. After that, data clustering methods are used for finding detailed sub-patterns for each detected pattern. Finally, the detected knowledge is explained and used for real time classification of IDS alerts, in order to characterize critical alerts from irrelevant ones. In ***[12], Mohammad M. N. et al.*** proposed intelligent DM IDS and its core part a composite detection engine with anomaly, and misuse detection features. The two detection engines work serially to detect the user's activity in turn. The system collects the data of DB audit system in real time, analyzes the audit data, judges that it is a normal behavior, abnormal behavior or aggressive behavior and responds to the result obtained by the operation behavior, and finally reports the result to the manager in a comprehensible form. In ***[13], Guojun Z. et al.*** presented a cooperative IDS based on IPv6 to address this challenge. Such a system consists of four parts: data flow tracking and analysis, capturing packets and rules matching, disaster recovery, and blocking. The technique of cooperative ID is introduced into the system for realizing the coordination control among parts. The system has a perfect detection rating. In ***[14], Baig M. N. et al.*** presented a model for feature selection that uses the information gain ratio measure as a means to compute the relevance of each feature and the k-means classifier to select the optimal set of MAC layer features that can improve the accuracy of intrusion detection systems while reducing the learning time of their learning algorithm. Experimental results with three types of neural network architectures clearly show that the optimization of a wireless feature set has a significant impact on the Efficiency and accuracy of the intrusion detection system. In ***[15], Neelakantan N. P. et al.*** show that 802.11 network, the features used for training and testing the intrusion detection systems, consist of basic information related to the TCP/IP header, with no considerable attention to the features associated with lower level protocol frames. The resulting detectors were efficient and accurate in detecting network attacks at the network and transport layers, but unfortunately, not capable of detecting 802.11-specific attacks such as de-authentication attacks or MAC layer DoS attack. In ***[16], Gupta V. et al.*** analyze attacks that deny channel access by causing pockets of congestion in mobile ad hoc networks. Such attacks would essentially prevent one or more nodes from accessing or providing specific services. In particular, focus on the properties of the popular medium access control (MAC) protocol, the IEEE 802.11x MAC protocol, enable such attacks. They show that conventional methods used in wire-line networks will not be able to help in prevention or detection of such attacks.

4. The Proposed Model of INIDS

From the survey of traditional and current NIDS, there are number of significant drawbacks. These critical issues must be taken in consideration to construct our proposed model. These drawbacks are:

1. There is no integrated wire and wireless NIDS work with attributes of the three layers of TCP/IP (transport, network and data link).
2. There is no available standard dataset for wireless intrusion detection as KDD'99 which presents the benchmark dataset for training and testing in wire networks.
3. A huge number of false positive occurs when normal attack is mistakenly classified as malicious and treated accordingly; also a huge number of false negatives. This is the case where IDS does not generate an alert when an intrusion is actually taking place.
4. Most NIDS are either anomaly detection models or Misuse detection models. Each one of them has its drawbacks.

The proposed INIDS aims to construct IDS to protect both wire and wireless. This means it will deal with three layers; transport, Internet and data link layers. By constructing two IDS separately; one for wire (Wire-NIDS) and the other for wireless (Wireless-NIDS), and then integrating them to construct the INIDS, see figure (1), as in the following consequences steps;

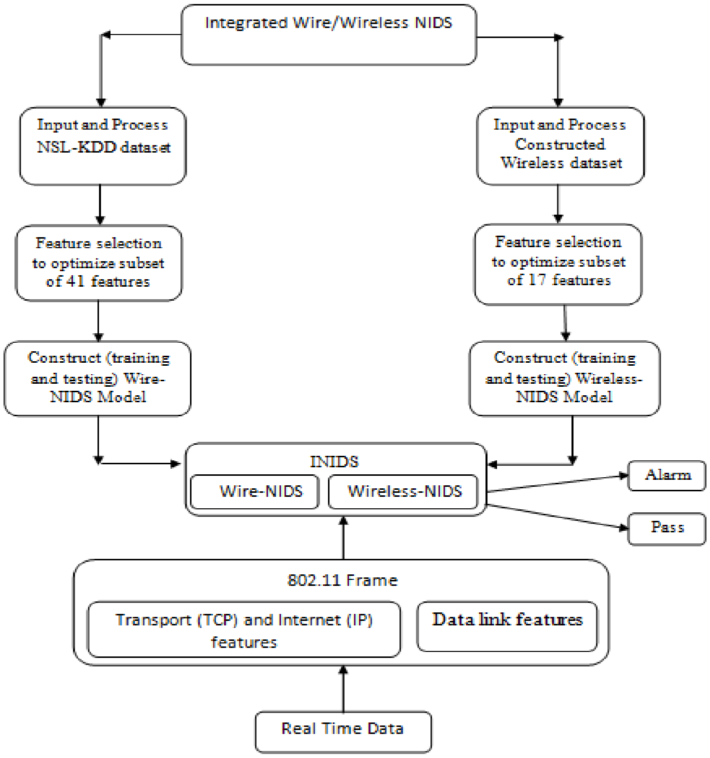

Integrated Wire/Wireless NIDS
Input and Process NSL-KDD dataset
Input and Process Constructed Wireless dataset
Feature selection to optimize subset of 41 features
Feature selection to optimize subset of 17 features
Construct (training and testing) Wire-NIDS Model
Construct (training and testing) Wireless-NIDS Model
INIDS
Wire-NIDS
Wireless-NIDS
Alarm
Pass
802.11 Frame
Transport (TCP) and Internet (IP) features
Data link features
Real Time Data

Figure (1) General block diagram of the proposal

1. Wire-NIDS Model is DM-based IDS for training and testing that will depend on *NSL-KDD dataset*, using PCA as a feature extraction and SVM as classifier.
2. Wireless-NIDS Model is DM-based IDS for training and testing that will depend on W data set *wireless data set* collected on WLAN 802.11, using GR as feature selection and NB as a classifier.
3. Integration of both Wire-NIDS and Wireless-NIDS is done by the following strategy:
   - Enter real-time data, frames of WLAN 802.11.
   - Separate the packet (transport layer and internet layers) from data link layer.
   - Process packet with Wire-NIDS and process data link with Wireless-NIDS.
   - Apply the logical truth table of (AND operation) to decide the frames (connections with three layers) to pass or trig alarm recorded as intrusion, see table (1).

Table (1) Truth table of final INIDS according to Wire-NIDS and Wireless-NIDS

Action	**Wire-NIDS**	**Wireless-NIDS**	**INIDS**
Action1	Pass	Pass	Pass
Action2	Pass	Alarm	Alarm
Action3	Alarm	Pass	Alarm
Action4	Alarm	Alarm	Alarm

4.1 The Proposal of Wire-NIDS Model

The proposed Wire-NIDS, see algorithm (1), is a hybrid because it trends to detect intrusions with both techniques of misuse and anomaly, and is multilevel because it trends to detect intrusions with two levels for more accurate intrusions type detections. The first level of the proposed Wire-NIDS will apply anomaly ID technique. It should first learn the characteristics of normal activities and abnormal activities of the system, and then the Wire-NIDS detect traffic that deviate from normal activities. So, the result of the first level is detecting the traffic as either normal or intrusions. If normal, then it passes, otherwise it enters the intrusion traffic to the second level to detect the class of intrusion. The second level of the proposed Wire-NIDS will apply Misuse ID technique and 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. The results of this level are detecting the type of intrusions.

For training and testing of the proposed Wire-NIDS in experiment, NSL-KDD data set is used. It has solved some of the inherent problems of the KDD'99. 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 protocols) each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. These features are grouped into four categories: basic features, content features, time-based traffic features and host-based traffic features. The simulated attacks fall into one of the following four categories: *Denial of Service Attack (DoS), User to Root Attack (U2R), Remote to Local Attack (R2L), and Probing Attack*. A total of 22 training known attack types and additional 17 unknown types are summarized.

The numbers of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable. Our proposed Wire-NIDS model indicates that feature reduction technique is capable of reducing the size of dataset. The time and space complexities of most classifiers used are exponential function of their input vector size.

Algorithm (1): Proposal of Wire-NIDS Model

Input: Input NSL-KDD dataset

Output: Wire-NIDS model classify real-time connection either pass (normal) or alarm (intrusion with specifying the name of intrusion)

Process:

1. Preprocess and normalize the selected records of NSL-KDD dataset.
2. Divide dataset into two parts Training dataset and Testing dataset.
3. Use feature extraction with PCA, see algorithm (1.1), to optimize dataset records with intrinsic features.
4. Training Wire-NIDS Model with Training Dataset
   - Train both datasets (dataset with all features and dataset with PCA features) with anomaly classifier, SVM see algorithm (1.2), to classify records either normal or Intrusions.
   - If records are intrusion, train them with misuse classifier, ANN see algorithm (1.3), to classify intrusions to their fourth classes and their types in the classes.
5. Test Wire-NIDS Model with Testing Dataset
   - Keep two copies of testing datasets first one without class labels and the second with class label.
   - Test both testing datasets without class labels (dataset with all features and dataset with PCA features) with anomaly classifier, SVM, to classify records either normal or Intrusions.
   - If records are intrusion, test them with misuse classifier, ANN, to classify intrusions to their fourth classes and their types in the classes.
   - Compare the results of testing the Wire-NIDS on testing datasets without classes with the original testing dataset with classes to calculate the measures of performance of the proposed Wire-NIDS. These measures are (accuracy and detection rate).
6. Evaluate the proposed system by testing the proposal with real-time data (online network connections).The results expected are
   - Pass (if the connection was normal).
   - Alarm (if the connection was intrusion).In our proposal the alarm which detects the class of intrusions also detects the type of intrusion in this class.

End.

4.1.1 Feature Extraction by PCA

In the proposal we used Principle Component Analysis (PCA) as feature extraction rather than using any technique of feature selection. We have 41 features. Having such a number of features will be time consuming in training and real-time detecting. So, we need to transform this set of 41 features into small subsets of correlated intrinsic features presenting the basic point in classification.

```
Algorithm (1.1): Suggested-PCA
Input: NSL-KDD training dataset.
Output: PCA set of most frequent and related features.
Process:
    1. Obtain training NSL-KDD'99 transactions.
    2. Represent every transaction Ii as a vector xi.
    3. Compute the mean transaction
         $\Psi = \frac{1}{M}\sum_{i=1}^{M} xi$ ……………………..…...(1)
    4. Subtract the mean transaction
         $\varphi i = xi - \Psi i$ ……………………..…. (2)
    5. Compute the covariance matrix
         $C = \frac{1}{M}\varphi_n \varphi_n^T$=$AA^T$ …………..……(3)
    6. From C, Compute eigenvectors $u_i$ of $AA^T$: Consider matrix $AA^T$ as a $M \times M$ matrix.
    7. Compute the eigenvectors vi of $AA^T$ so that:
         $A^TAv_i \rightarrow \mu_i V_i \rightarrow AA^TAV_i = \mu_i Av_i \rightarrow Cu_i = \mu_i u_i$ where$\mu_i = Av_i$………………..…..(4)
    8. Compute the $\mu$ best eigenvectors of $AA^T$: $\mu_i = Av_i$……………………..…..…(5)
    9. Keep only K eigenvectors, (K features with their values).
End.
```

The complete subject of PCA statistics is based on the idea that you have this huge set of data, and you want to analyze that set of expressions of the relationships between the single points in that set. PCA is applied to the *training dataset* to find the intrinsic features, see algorithm (1.1).The resulted set of features in addition to the original set present all features in the *training dataset* will be used in design (learning) of the classifiers.

4.1.2 SVM and ANN Classifiers in Wire-NIDS Model

After the intrinsic features had been selected, the two popular DM classification algorithms: Support Vector Machine SVM from statistical field and Artificial Neural Networks from soft computing field, will be used in the design of the suggested Wire-NIDS in the shape of two levels. So, both of SVM and ANN classifiers will be trained 2 times, one training with dataset of all 41 features and another with dataset of PCA features to design the suggested Wire-NIDS classifier.

SVM is a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. SVMs attempt to separate data into multiple classes (two in the basic case) through the use of a hyper-plane. Here, it will be used in a more conventional SVM approach. We used one SVM as anomaly detection techniques to identify normal traffic from intrusion traffics. Algorithm (1.2) explains SVM algorithm with Anomaly Intrusion detection learned on NSL-KDD dataset.

```
Algorithm (1.2): Suggested-SVM
Input: NSL-KDD for training and testing.
Output: Results of Anomaly detection on NSL-KDD dataset using SVM.
Process:
    1. Initialize all points in training dataset as (Xi, Yj) where X is a vector of data x1, ……,xn and Y is vector of classes.
    2. Initialize vector of weight W.
    3. Distribute all points (x, y) and extract the hyper plane separator.
    4. If the hyper plane gives optimal separation then depend on hyper plane as classifier model to classify testing dataset and go to End
    5. Else, the following steps must be done
    6. Maximize the hyper plan using equation of Getting Maximum Margin

        MM = 2 / ||w||. …………. (6)

    7. For minimum use the same equation as maximizing

        $\Phi(w) = \frac{1}{2}(w)^t w$ ……….(7)

    8. Initialize Lagrange multiplier $\alpha_i$ vector $\alpha_1 \dots \alpha_n$ using equation
    9.
        $Q(\alpha) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j x_i^T x_j$…………. (8)

    7. Apply classification function using equation
    8.
        $f(x) = \Sigma\alpha_i y_i x_i^T x + b$ …………. (9)

    9. Determine the support vectors xi with non-zero $\alpha_i$ (support vectors are the points that determine the area of hyper plan)
    10. Depend on the hyper plan resulted after determining support vectors as the classifier model to classify testing dataset
End
```

ANN is a system simulating the work of the neurons in the human brain. The neuron consists of some inputs emulating dendrites of the biological neuron, a summation module, an activation function and one output emulating an axon of the biological neuron. The importance of a particular input can be intensified by the weights that simulate biological neuron's synapses. Then, the input signals are multiplied by the values of weights and next the results are added in the summation block. The sum is sent to the activation block where it is processed by the activation function. Thus, we obtain neuron's answer to the input signals "x". Here, it will be used in more conventional MLP (Multi Layer Perceptron) approach as misuse detection techniques to identify the class of intrusion traffics (these are detected intrusion in the first level classifier and sent to the second level to determine its class). Algorithm (1.3) explains ANN algorithm with Misuse Intrusion detection learned on NSL-KDD dataset.

```
Algorithm (1.3): Suggested-ANN
Input: NSL-KDD for training and testing.
Output: Results of Misuse detection on NSL-KDD dataset using ANN.
Process:
    1. Main Assumption for the Training Process of MLP:
        • Learning method: Quasi Newton BFGS and Levenberg-Marquardt
        • Number of Epochs: 1000.
        • MSE (Mean Square error): 0.01.
        • Learning rate: 0.9.
        • Activation function: log-sigmoid.
        • Number of neurons in the Input layer: (41 or according no. of PCA set).
        • Number of neurons in the hidden layer: (21 for 41 input neurons and with no. of PCA set equal to half of this no.).
        • Number of neurons in the output layer: (4 cause no. of intrusions classes are 4).
        • Update of weights – batch mode (after presentation of the entire training data set).
    2. Train and Test on NSL-KDD to construct final ANN misuse NIDS.
End.
```

4.2 The Proposal of Wireless-NIDS Model

The proposed Wireless-NIDS, see algorithm (2), is a DM-based IDS in which both the misuse and anomaly detection techniques depended on the detection of intrusion, where each record in a constructed dataset is labeled as "normal" or "intrusion (specify class of intrusion)" and a learning algorithm is trained over the labeled dataset. Misuse technique is able to automatically retrain IDS 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.

The data used to train and test the classifiers were collected from a wireless local area network. The local network is composed of 5 wireless stations and two one access points. One machine is used to generate normal traffic of HTTP and FTP. The second, third and fourth machines simultaneously transmit data originating from 4 types of WLAN attacks. The fifth station is used to collect and record both types of traffic (normal and intrusive types of intrusion). The attacks used to test the proposed WNIDS are the 4 types of WLAN attacks (de-authentication, chop-chop, fragmentation and duration). There is no real WLAN traffic dataset which can be considered as benchmark to be used in this area of research. Then we select and construct features from MAC layer (Data link layer). To collect WLAN traffics from the constructed WLAN 802.11, we use Wires hark tool.

Algorithm (2): Proposal of Wireless-NIDS Model

Input: Input Wdataset dataset

Output: Wireless-NIDS model classify real-time connection either pass (normal) or alarm (intrusion with specifying the type of intrusion)

Process:

1. Preprocess and normalize the selected records of W data set dataset.
2. Divide dataset into two parts Training dataset and testing dataset.
3. Use feature extraction with Gain-Ratio, see algorithm (2.1), to optimize dataset records with intrinsic features.
4. Train Wireless-NIDS Model with Training Dataset

- Train both datasets (dataset with all features and dataset with GR features) with anomaly and misuse classifier, **Naïve Bayesian Classifier,** see algorithm (2.2), to classify records as either normal or Intrusions and classify intrusions to their fourth classes and their types in the classes.

5. Test Wireless-NIDS Model with Testing Dataset

- Keep two copies of testing datasets; first one without class labels and the second with class label.
- Test both datasets without class labels (dataset with all features and dataset with GR features) with **Naïve Bayesian Classifiers**, to classify records as either normal or Intrusions and classify intrusions to their fourth classes and their types in the classes.
- Compare the results of testing the Wireless-NIDS on testing dataset without classes with the original testing dataset with classes to calculate the measures of performance of the proposed Wireless-NIDS. These measures are (accuracy and detection rate).

6. Evaluate the proposed system by testing the proposal with real-time data (online network connections) the results expected are

- Pass (if the connection was normal).
- Alarm (if the connection was intrusion) in our proposal the alarm detecting the class of intrusions also detects what type of intrusion in this class.

End

In the proposal, we collected 2000 frames. From each frame we extract the following 16 features: the 15 features are clearly found in the header and we add one more feature not found explicitly in frame. This is the last one (casting type); Protocol version, Type, Subtype, To DS, From DS, More Fragmentation, Retry, Power Management, More Data, Protected Wep, Order, Duration, RA, TA, MA, FCS, Casting type.

The data collected were grouped in two sets: training and testing. The first set is used to reach the optimal classifiers. The training set contains the input with its desired output. The testing datasets are necessary to avoid the effect of over fitting. It should be able to predict the output of each entry of the testing data set.

4.2.1 Feature Selection by Gain Ratio

Given entropy *Entropy(S)* as a measure of impurity in a collection of items, it is possible to quantify the effectiveness of a feature in classifying training data. *InfoGain* measures the expected reduction of entropy caused by partitioning the dataset according to feature F_j, in which V is the set of possible values of F_j. The issue with *InfoGain* measure is that it favors features having a high variety of values over those with only a few. *GR* measure overcomes this problem, as explained in Algorithm (2.1), by considering how the feature splits the dataset. S_i is d subsets of records resulting from partitioning S by the d-valued feature F_j. Only the features with the highest *GR* will be selected.

```
Algorithm (2.1) Gain_Ratio
Input: Wdataset training dataset.
Output: GR set of most frequent and related features.
Process:
  1. Initialize a set of features GR_F to $\emptyset$
  2. For each feature F in Wdataset, Do steps 3-6
  3. Find InfoGain of F using

  $InfoGain(S, F_j) = Entropy(S) - \sum_{v_i \in V_{F_j}} \frac{|S_{vi}|}{|S|} . Entropy(S_{v_i})$ …….(10)

  Where
  $Entropy(S) = \sum_{i=1}^{c} -p_i . \log_2 p_i$ …………..(11)

  4. Find Split Information of F using

  $Split\ Information(S, F_j) = \sum_{i=1}^{C} -\frac{|S_i|}{|S|} . \log_2(\frac{|S_i|}{|S|})$ ……………….(12)
  5. If Split Information = 0 Then set it to 1E-25
     (it is a very small value)
  6. Find GR of F using
  $GR(S,F_j) = \frac{InfoGain(S,F_j)}{SplitInformation(S,F_j)}$ …….(13)
  7. If algorithm used as feature selection measure
     Then select half of wdataset features having
     the highest GR values and put them into
     GR_F
  End
```

4.2.2 NB Classifier in wireless-NIDS Model

In NB classifier, see algorithm (2.2), a set of probabilities (*a priori, conditional, and posteriori*) has been found instead of constructing a set of classification rules. Firstly, the "*a priori probability*" of each class (i.e. the frequency of each class in the *trainingdataset*) is computed. The a priori probability is computed just once 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 j^{th} feature in class C . Assuming the *conditional independence* of features, the "*conditional probabilities*" $P(X|C_i)$ of the testing record at each class is computed using equation (14). Finally, the "*postpriori probability*" $P(h|X)$ of the testing record at each class is computed using equation (15).The class with the maximum postpriori probability h_{MAP} will be the label for the testing record according to equation (16).

```
Algorithm(2.2) Naive Bayesian
Input: Wdataset for training and testing
Output: Results of Anomaly-Misuse detection on Wdataset using NB.
Process:
  1. Initialize MaxValue to a small value
  2. For each class $C_i$ in training wdataset find its a priori probability $AP_i$
  3. For each record R in testing wdataset do step 4 and 8
  4. For each class $C_i$ in training wdataset repeat steps 5-7
  5. Find the conditional probability $CP_i$ of R at $C_i$ using

     $P(X|C_i) = \prod_{j=1}^{n} P(a_j|C_i)$ ………………….(14)

     Find the postpriori probability$PP_i$of R using

     $P(h|X) = \frac{P(X|h)P(h)}{P(X)}$ …………………… (15)

     Where $h_{MAP} \equiv arg\ max_{h \in H} = arg\ max_{h \in H} P(X|h)P(h)$…….. (16)

  6. If $PP_i$is greater than MaxValue
        then MaxValue = $PP_i$ and class_label = $C_i$
  7. Assign class_label to the class of R

End
```

5. Discussion and Experimental work

The proposal had been implemented on Windows 7 Ultimate Service Pack1 and 32-bit OS, 16GB RAM, and Intel® Core (TM) 3 Duo CPU with 2.00 GHz; and using Visual Basic.Net and SQL server.

Before explaining the results, we will introduce the number of records taken from NSL-KDD as samples for training and testing, see table (2).Training dataset in the study contained 60,000 records, which were randomly generated from the NSL-KDD for training dataset that consists of 10,050 normal patterns, 40, 050 known DoS patterns, 550 known Probe patterns, 200 known R2L patterns and 150 known U2R patterns. Testing dataset in the paper contained 33,750 records, which were randomly generated (with omitting the records of training) from the NSL-KDD for testing data set that consists of 7,050 normal patterns, 23, 050 known and unknown DoS patterns, 1,050 known and unknown Probe patterns, 2050 known and unknown R2L patterns and 550 known and unknown U2R patterns.

Table (2): No. of Records selected from NSL-KDD for Training and Testing

No. of Records/Type of Records	Training	Testing
DOS	40,050	23,050
Probe	550	1,050
R2L	200	2,050
U2R	150	550
Normal	10,050	7,050

In the proposal, 3000 frames were collected using wires hark tools to construct Wdataset which is conducted by selecting 1750 (1250 for training and 500 for testing) frames with optimal features' values (no missing values, no noise and no redundancy) and then distributing the data collected to training and testing and specifying the number of frames for normal and each attacks in both training and testing as in table (3).

Table (3): No. of Records selected from Wdataset for Training and Testing

No. of Records/Type of Records	Training	Testing
Deauthentication	270	90
Chopchop	230	100
Duration	250	100
Fragmentation	200	90
Normal	300	120

Training which consist of two IDS (Wire-NIDS and Wireless-NIDS) on two training dataset has been done with two sets of features for each IDS. With Wire-NIDS the two sets are (Features (41), Features PCA (8)), and with Wireless-NIDS the two sets are (Features (17), Features GR (8)). The set of intrinsic features obtained by applying PCA on training dataset is the subset {Protocol_type, Service, Flag, count, srv_count,ame_srv_rate, dst_host_srv_count, dst_host_same_srv_rate, st_host_same_src_port_rate}. The best set of features of the GR among normal, four known intrusion and unknown intrusions, is the following subset {Type, Subtype, To DS, More Fragmentation, Retry, Protected Wep, Duration, Casting type}.

The two proposed systems have been experimented (trained and tested) twice to assess the accuracy of the classifiers. Results of three conducted experiments (Exp1, Exp2, Exp3), which produced the most accurate results, have been presented in this section. Four classification models have been constructed in each of these three experiments. These models have been applied on the same *Testing dataset*, which has been constructed during Exp1, to assess the validation and accuracy of these constructed models on the same testing dataset. The classification results of testing are either **TP** (intrusion), **TN** (normal), false positive (**FP**) (misclassified as intrusion), false negative (**FN**) (misclassified as normal), or **Unknown** (new user behavior or new attack). From classification results the detection rate (**DR**) of the two proposed models was calculated, which is the ratio between the number of TP and the total number of intrusion patterns presented in the testing dataset. It has been calculated using

$$\text{DR} = \frac{TP}{TP+FN+Unknown2} \text{*100\%.........(17)}$$

The false alarm rate (FAR) of an IDS is the ratio between the number of "normal" patterns classified as attacks (FP) and the total number of "normal" patterns presented in the testing dataset. It has been computed using

$$\text{FAR} = \frac{FP}{TN+FP+Unknown1}\text{*100\%......(18)}$$

Values for both DR and FAR for each classifier in the three experiments have been illustrated in table (4a and 4b).

DR are higher with Wire-NIDS classifiers and also with Wireless-NIDS classifiers and FAR often ranging between (0 - 0.06) with classifiers. It is very clear from these that Wire-NIDS and Wireless-NIDS classifiers render better detection and less false alarms with reduced features by PCA and GR.

After integrating the two models to construct INIDS, now we will explain classification accuracy of INIDS model, which is introduced as the ratio between the number of the correctly classified patterns (TP, TN) and the total number of patterns of the testing dataset. The ***accuracy*** of INIDS classifier has been calculated using:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN+unknown}*100\% \qquad \text{……….} \quad (19)$$

Table (4): DRs and FARs of both Wire-NIDS and Wireless-NIDS

(a)

Wire-NIDS			
Feature Selection Measure	Experiment No.	DR	FAR
PCA	1	1	0
	2	0.998	0
	3	1	0
ALL	1	0.996	0.01
	2	0.997	0.02
	3	0.995	0.06

(b)

Wireless-NIDS			
Feature Selection Measure	Experiment No.	DR	FAR
GR	1	1	0
	2	0.999	0
	3	1	0
ALL	1	0.996	0.04
	2	0.990	0.05
	3	0.996	0.04

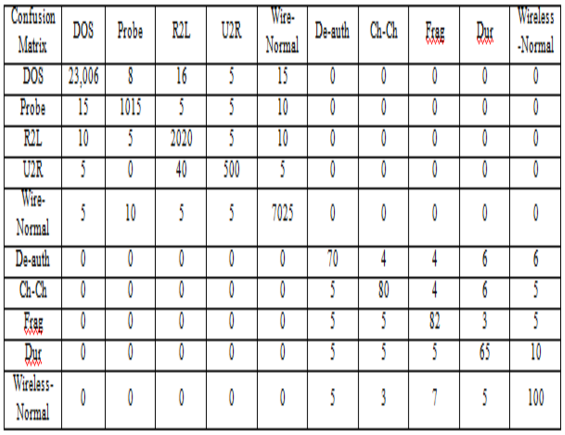
Table (5) summarizes *Accuracy* of INIDS classifiers with both all features set and reduced features subsets by PCA and GR in the three experiments. According to these results, the classifier INIDS was more accurate with subset of features obtained by PCA and GR.

Table (5): Accuracy of INIDS Classifiers

Classifier	Experiment No.	PCA & GR features	ALL features
INIDS	1	1	0.995
	2	1	0.996
	3	0.999	0.997

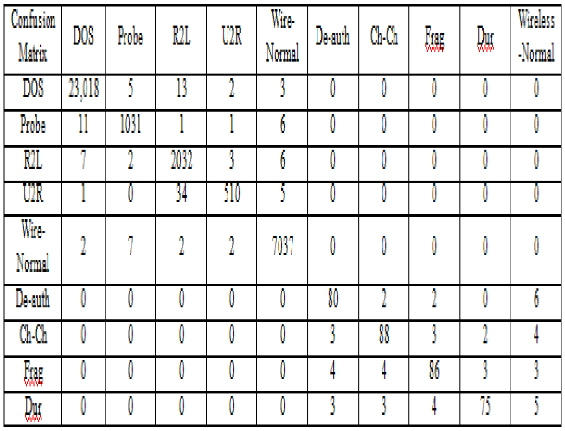
For more precision of accuracy we will explain the overall accuracy of the proposed INIDS according to confusion matrix calculations, will display the confusion matrix for two cases; all features (41 and 17) and reduced features (8 and 8), see table (6) and table (7). See the confusion of classification with (8 and 8) features less than confusion of classification with (41 and 17) features.

Table (6): confusion matrix of INIDS with (41 features and 17 features)



Confusion Matrix	DOS	Probe	R2L	U2R	Wire-Normal	De-auth	Ch-Ch	Frag	Dur	Wireless -Normal
DOS	23,006	8	16	5	15	0	0	0	0	0
Probe	15	1015	5	5	10	0	0	0	0	0
R2L	10	5	2020	5	10	0	0	0	0	0
U2R	5	0	40	500	5	0	0	0	0	0
Wire-Normal	5	10	5	5	7025	0	0	0	0	0
De-auth	0	0	0	0	0	70	4	4	6	6
Ch-Ch	0	0	0	0	0	5	80	4	6	5
Frag	0	0	0	0	0	5	5	82	3	5
Dur	0	0	0	0	0	5	5	5	65	10
Wireless-Normal	0	0	0	0	0	5	3	7	5	100

Table (7): confusion matrix of INIDS with (8 features and 8 features)



Confusion Matrix	DOS	Probe	R2L	U2R	Wire-Normal	De-auth	Ch-Ch	Frag	Dur	Wireless -Normal
DOS	23,018	5	13	2	3	0	0	0	0	0
Probe	11	1031	1	1	6	0	0	0	0	0
R2L	7	2	2032	3	6	0	0	0	0	0
U2R	1	0	34	510	5	0	0	0	0	0
Wire-Normal	2	7	2	2	7037	0	0	0	0	0
De-auth	0	0	0	0	0	80	2	2	0	6
Ch-Ch	0	0	0	0	0	3	88	3	2	4
Frag	0	0	0	0	0	4	4	86	3	3
Dur	0	0	0	0	0	3	3	4	75	5

6. Conclusion

- Data mining is introduced to help IDS to detect intrusions correctly, and accordingly IDSs have shown to be successful in detecting unknown attacks.
- Using NSL-KDD instead of KDD'99 gives more reliability for detections results since NSL-KDD is less redundant and has less noisy data.
- Concentrating on selecting a minimum number of features produces optimal results in accuracy, causes more times the huge number of feature causing bed performance, since not all features are related in detecting the attacks. So if these features are not omitted they will present the noise in datasets.
- This definitely has an impact on the overall performance of the system. The features are reduced from 41to 8 for wire-NIDS and from 17 to 8 for Wireless-NIDS. The above experiments show that the reduced features increased accuracy, reduced training and computational overheads and simplified the architecture of intrusion analysis engine.
- The proposed integration of the wire and wireless IDS give strong immunity for networking against attacks especially with the new platforms such as cloud computing and mobility.

References

1. Lalli and Palanisamy, "Modernized Intrusion Detection Using Enhanced Apriori Algorithm", International Journal of Wireless & Mobile Networks (IJWMN) Vol. 5, No. 2, April 2013, [IVSL].
2. Barhoo T. S. and ElShami E., "Detecting WLANs' DoS Attacks Using Back propagate Neural Network", Journal of Al Azhar University-Gaza (Natural Sciences), 2011, 13 : 83-92,

3. 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.
4. 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.
5. Huang L. and Hwang M. "Study of Intrusion Detection Systems", Journal Of Electronic Science And Technology, Vol. 10, No. 3, September 2012.
6. Hashem S. H. and Abdulmunem I. A. "A Proposal To Detect Computer Worms (Malicious Codes) Using Data Mining Classification Algorithms", Eng. &Tech. Journal .Vol31, Part (B), No. 2, 2013.
7. Hashem S. H. , "Efficiency Of SVM And PCA To Enhance Intrusion Detection System", Dr. Soukaena Hassan Hashem, Journal Of Asian Scientific Research, 2013, 3(4):381-395.
8. Ahmad I., Abdulah A. B, Alghamdi A. S, Alnfajan K. and Hussain M., "Feature Subset Selection for Network Intrusion Detection Mechanism Using Genetic Eigen Vectors", 2011 International Conference on Telecommunication Technology and Applications, Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore.
9. Bensefia H. and Ghoualmi N., "A New Approach for Adaptive Intrusion Detection", 2011 Seventh International Conference on Computational Intelligence and Security, 2011.
10. Vaarandi R., "Real-Time Classification of IDS Alerts with Data Mining Techniques", MILCOM'09 Proceedings of the 28th IEEE conference on Military communications pp.1786-1792.
11. Vaarandi R., and Podinš K., "Network IDS Alert Classification with Frequent Item set Mining and Data Clustering", The 2010 International IEEE Conference on Network and Service Management, pp. 451-456.
12. Mohammad M. N., Sulaiman N. and Muhsin O. A., "A Novel Intrusion Detection System by using Intelligent Data Mining in Weka Environment", Published by Elsevier Ltd. Procedia Computer Science 3, pp. 1237–1242.
13. Guojun Z., Liping C. and Weitao H., "The Design of Cooperative Intrusion Detection System", IEEE Computer Society, 2011 Seventh International Conference on Computational Intelligence and Security, pp. 764-766.
14. Baig M. N. and Kumar K. K. , "Intrusion Detection in Wireless Networks Using Selected Features", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (5) , 2011, 1887-1893.
15. Neelakantan N. P., Nagesh C. and Tech M.., "Role of Feature Selection in Intrusion Detection Systems for 802.11 Networks", International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN) Volume-1, Issue-1, 2011.
16. Gupta V., Krishnamurthy S., and Faloutsos M., " Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks", This material is based upon work supported by the National Science Foundation under Grant No. 9985195, DARPA award N660001-00-18936.