Data dimensionality reduction based on genetic selection of feature subsets

K.M. Faraoun [1], A. Rabhi[2]
[1] Evolutionary Engineering and Distributed Information Systems Laboratory, EEDIS
UDL Unixversity- SBA, 22000 - Algeria
Kamel_mh@yahoo.fr
[2]Laboratoire des mathématiques, UDL University,
SBA, 22000 - Algeria
rabhi_abbes@yahoo.fr

Abstract. In the present paper, we show that a multi-classification process can be significantly enhanced by selecting an optimal set of the features used as input for the training operation. The selection of such a subset will reduce the dimensionality of the data samples and eliminate the redundancy and ambiguity introduced by some attributes. The used classifier can then operate only on the selected features to perform the learning process. A genetic search is used here to explore the set of all possible features subsets whose size is exponentially proportional to the number of features. A new measure is proposed to compute the information gain provided by each features subsets, and used as the fitness function of the genetic search. Experiments are performed using the KDD99 dataset to classify DoS network intrusions, according to the 41 existing features. The optimality of the obtained features subset is then tested using a multi-layered neural network. Obtained results show that the proposed approach can enhance both the classification rate and the learning runtime.

Keywords: Features selection, genetic algorithms, patterns classification.

(Received October 10, 2006 / Accepted January 03, 2007)

1 Introduction

Pattern recognition relies on the extraction and selection of features that adequately characterize the objects of interest. The task of identifying the features that perform well in a classification algorithm is a difficult one, and the optimal choice can be non-intuitive; features that perform poorly separately can often prevail when paired with other features [1]. The filter approach [2] to feature selection tries to infer which features will work well for the classification algorithm by drawing conclusions from the observed distributions (histograms) of the individual features. However, the histograms give little insight into the separation between polyps and non-polyps. The correlation structure of the data is responsible for the success of the joint classifier, and a good classification scheme will attempt to utilize this structure.

Another technique, known as wrapper feature selection [3], uses the method of classification itself to measure the importance of a feature or features set. The goal in this approach is maximizing the predicted classification accuracy. This approach, while more computationally expensive, tends to provide better results than the simpler filter methods.

Recent work in the field of pattern recognition explores the use of evolutionary algorithms for feature selection, and genetic algorithms (GAs) are one type of evolutionary algorithm that can be used effectively as engines for solving the features selection problem. Features selection using genetic algorithms has been studied and proven effective in conjunction with various other classifiers.

Most of the existing works are focused on the wrapper mode using different classifier method (Neural networks, SVM, K-NN…..), and the same binary chromosomes representation is generally used. A binary string represents the set of all existing features, with a value of 1 at the i^{th} position if the i^{th} feature is selected, and 0 otherwise. The advantage of this representation is that a standard and well understood GA could be used without any modification. Unfortunately, the model of chromosome is only appropriate for data that have small and medium features. It caused an exponential nature of subsets that exist as the number of features increases. If the number of features is large, it becomes difficult to evaluate all possible combinations of features.

When the major works related to features selection agree that wrapper mode give better results than filter one, this is not always true especially for very large datasets. Training a neural network or an SVM on 100000 samples for each chromosome during each generation of the genetic process became impracticable even on dedicated machines. This approach is useful when the number of training sample is limited according to the features one (the data space dimension). For this reasons, this paper try to show that filter features selection approach can achieve acceptable performances for large datasets when a good fitness function and chromosomes representation are chosen. Another advantage of the proposed approach is it's independence from the classification method implemented, the selected sub set of features can be used as input of any classification schema.

In the present work, we implement a novel chromosomes representation with appropriates genetic operators in order to enhance the convergence of the genetic search. The proposed chromosomal schema is closer to the real perception of the subsets combinations, when the relevant operators can be more logically interpreted than binary ones. A new fitness function schema is also proposed using a new information gain measure elaborated for subsets of features. The use of the GA offers a practical approach to feature selection for DoS attacks taken from the KDD99 dataset. We try to evolve a population of possible feature combination to find the one who give the better discrimination between the DoS attack classes. The genetic search in our work is guided using the proposed fitness measure that compute the information gain of a given features subset. The result of the search is then an optimal features subset, which will be used with a multi-layer neural network to learn the class's discrimination using only the information provided by its features.

The remaining of this paper is organized as follow: first, a theoretic background about GAs and features selection is presented with some related works in the section 2. A detailed explanation of the proposed approach is then given in section 3 with the different elements of the genetic process and the neural network classifier. The section 4 summarizes the obtained results and performances. The paper is finally concluded with a summary of the most important points and future works.

2 Theoretic Background

2.1 Genetic Algorithms

Genetic Algorithms (GAs) are a family of computational models inspired by evolution. Computational studies of Darwinian evolution and natural selection have led to numerous models for computer optimization [4, 5]. GAs comprises a subset of these evolution-based optimization techniques focusing on the application of selection, mutation, and recombination to a population of competing problem solutions. GAs are parallel iterative optimizers, and has been successfully applied to many optimization problems, including pattern recognition and classification tasks. Being a directed search rather than an exhaustive search, population members cluster near good solutions; however, the GA's stochastic component does not rule out wildly different solutions, which may turn out to be better. This has the benefit that, given enough time and a well bounded problem, the algorithm can find a global optimum. This makes them well suited to feature selection problems (they can find near optimum solutions using little or no a priori knowledge).

There are three major design decisions to consider when implementing a GA to solve a particular problem. A representation for candidate solutions must be chosen and encoded on the GA chromosome, an objective (fitness) function must be specified to evaluate the quality of each candidate solution, and finally the GA run parameters must be specified, including which genetic operators to use, such as crossover, mutation, selection, and their possibilities of occurrence.

The process of fitness-dependent selection and application of genetic operators to generate successive generations of individuals is repeated many times until a satisfactory solution is found. In practice, the performance of genetic algorithm depends on a number of factors including: the choice of genetic representation and operators, the fitness function, the details of the fitness-dependent selection procedure, and the various user-determined parameters such as population size, probability of application of different genetic operators, etc.

2.2 Feature Subset Selection

The term feature subset selection is applied to the task of selecting those features that are most useful to a particular classification problem from all those available [6]. The main purpose of feature subset selection is to reduce the number of features used in classification while maintaining acceptable classification accuracy. Less discriminatory features are eliminated, leaving a subset of the original features which retains sufficient information to discriminate well among classes [7]. For classical pattern recognition techniques, the patterns are generally represented as a vector of feature values. The selection of features can have a considerable impact on the effectiveness of the resulting classification algorithm. Consider a feature set, $F = \{f_0, f_{1,} \ldots, f_N\}$. If f_0 and f_1 are dependent, that is they always move together, then one of these could be discarded and the classifier has no less information to work with. This has the benefit that computational complexity is reduced as there is smaller number of inputs. Often, a secondary benefit found is that the accuracy of the classifier increases. This implies that the removed features were not adding any useful information but they were also actively hindering the recognition process. The problem of feature selection can be seen as a case of feature weighting, where the numerical weights for each of the features have been replaced by binary values. A value of 1 could mean the inclusion of the corresponding feature into the subset, while a value of 0 could mean its absence. In a domain where objects are described by d features, there are 2^d possible feature subsets. Obviously, searching exhaustively for the best subset (using any criteria to measure the quality) is futile. For this reason, the genetic algorithms has been identified as the best tools to explore such search space, and produce pseudo-optimal solutions that are sufficient to produce acceptable results.

2.3 GAs for features selection: Related Work

Existing work in the field of pattern recognition explores the use of evolutionary algorithms for feature selection [8,9,10], and genetic algorithms are one type of evolutionary algorithms that can be used effectively as engines for solving the feature selection problem. The features selection using genetic algorithms has been studied and proven effective in conjunction with various classifiers, including k-nearest-neighbours, and neural networks [9,11].

In [1] Yang and Hanovar investigated combinations of genetic algorithm and neural network. Eads et al.[13] and Sepulveda-Sanchis et al.[14] combined genetic algorithm and SVM. Liu and al. in [15] combined the parallel genetic algorithm with classification method proposed by Golub and al. In [16] a combination of SVM and GAs features selection is proposed for gene expression classification. Boudjeloud and Poulet [17] have used the Calinski index value as a fitness measure to evaluate the efficacy of each chromosome representing a dimensions combination.

Besides selecting feature subsets, GAs can extract new features by searching for a vector of numeric coefficients that is used to transform linearly the original features [18, 19]. In this case, a value of zero in the transformation vector is equivalent to avoiding the feature. Raymer et al. [20] combined the linear transformation with explicit feature selection flags in the chromosomes, and reported an advantage over the pure transformation method.

More sophisticated Distribution Estimation Algorithms (DEAs) have also been used to search for optimal feature subsets. DEAs explicitly identify the relationships among the variables of the problem by building a model of selected individuals and use this model to generate new solutions. However, in terms of accuracy, the DEAs do not seem to outperform simple GAs when searching for feature subsets [21, 22]. Anther idea proposed in [23] is the use of a measure of class separability to select features; it has been used generally in machine learning and computer vision.

With respect to the existing works concerning the use of genetic algorithm for features selection, our proposed work present two main difference: firstly, the dataset used for benchmarking is very large, and any wrapper approach will fail to deal with such problem in a reasonable runtime, secondly, a new fitness function is proposed, and it seem to be the more appropriate for multi-category classification problems as will be shown by experimental results.

3 Methodology

As explained above, the main goal of this work is to use genetic algorithm to select the best feature subset that discriminate the DoS attack classes, using the KDD99 dataset. The selected subset is included in the set of 41 existing attributes that handle the information in the dataset. In the remaining, we will first give some details about the used dataset and the required codification of the different features. Then, the details of the proposed approach are explained, with the main components of our classification system.

3.1 The dataset details

The KDD 99 intrusion detection datasets are based on the 1998 DARPA initiative, which provides designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies [24]. To do so, a simulation is made of a factitious military network consisting of three ‘target’ machines running various operating systems and services. Additional three machines are then used to spoof different IP addresses to generate traffic. Finally, there is a sniffer that records all network traffic using the TCP dump format.

Normal connections are created to profile that expected in a military network and attacks fall into one of four categories: User to Root; Remote to Local; Denial of Service; and Probe. In 1999, the original TCP dump files were pre-processed for utilization in the Intrusion Detection System benchmark of the International Knowledge Discovery and Data Mining Tools Competition [25]. To do so, packet information in the TCP dump file is summarized into connections. Specifically, “a connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol” [25]. This process is completed using the Bro IDS, resulting in 41 features for each connection, which are detailed in Table 1.

Basic features of individual TCP connections	duration , protocol_type service , flag , src_bytes , dst_bytes , land , wrong_fragment , urgent
Traffic features computed using a two-second time window	Count, srv_count, serror_rate srv_serror_rate, rerror_rate srv_rerror_rate, same_srv_rate diff_srv_rate, srv_diff_host_rate dst_host_count, dst_h_srv_count dst_h_same_srv_rate, dst_h_diff_srv_rate dst_h_s_src_port_rate, dst_h_srv_diff_h_rate dst_h_serror_rate, dst_h_srv_serror_rate dst_h_rerror_rate, dst_h_srv_rerror_rate
Content features suggested by domain knowledge	Host, num_failed_logins logged_in, num_compromised root_shell, su_attempted num_root, num_file_creations num_shells, num_access_files num_outbound_cmds is_hot_login, is_guest_login

Table 1: The 41 used feature in the KDD99 dataset grouped in three groups

In the present work, only attacks of type denial of service (DoS) are used. This type of attacks is separated in 6 possible classes: Neptune, Smurf, Teardrop, Back, Pod and Land. So the goal of the classification here is to decide to witch class belong a given DoS attack. When using the KDD99 dataset, only 10% of data is used for training, and that contain 391458 DoS attack record. When using GAs, dealing with such record count become very difficult due to computation limitation. So only a subset has been sampled to be used during the training phase. The Table 2 summarize the distribution of different classes used during this work.

The 41 existing features have not all the same data type, some are numeric values and others are symbolic data. This will prevent any classification system to be effective lead to ambiguity of the classification decision. For that, a preliminary codification step is necessary, all the features are converted into numeric values and normalized to the interval [0,1] using the codification schema presented in [26].

Class	Record Count
Neptune	10878
Smurf	11082
Teardrop	2111
Back	879
Pod	222
Land	23

Table 2: Distribution of the attack classes used for training

3.2 The proposed approach

In the following, the details of the proposed method are presented. The Figure 1 presents the general schema of the selection and classification process. Firstly, a population of possible features subset is genetically evolved. The genetic evolution is guided using the proposed fitness criterion, the quality of a given chromosome is proportional to the information gain measure computed using the dataset records retrieved from the training dataset, the obtained solution is then used to train a neural network using the same training set and the resulting network is finally validated using a new test dataset extracted from the KDD99 corrected testing dataset.

3.2.1 Chromosome structure

We implement a novel chromosomes representation with appropriates genetic operators in order to enhance the convergence of the genetic search. The proposed schema is closer to the real perception of the subsets combinations, when the relevant operators can be more logically interpreted than binary ones.

The Figure 2 shows the proposed chromosomes representation. Each gene is ant integer value that represents the index of a given feature from the existing ones (41 in the case of the used KDD99 dataset).

3.2.2 Crossover operator

The crossover operator is introduced in the genetic search to combine the possible characteristics of tow parents in a new tow offspring's. In this works, the crossover is performed by selecting a random position in each parent, and the combine the two ones according to the obtained fragments. The Figure 3 explains the crossover process.

3.2.3 The mutation operator

The mutation operator is used in the genetic algorithms to introduce diversity of in the population. This operator will give new possible solution not explored, and permit to escape to the local minima. We have proposed the operator illustrate in the Figure 4: each gene of the chromosome is replaced by a random value (of a valid feature) according to a mutation rate (generally very small). The resulting chromosome will replace its parent in the new population.

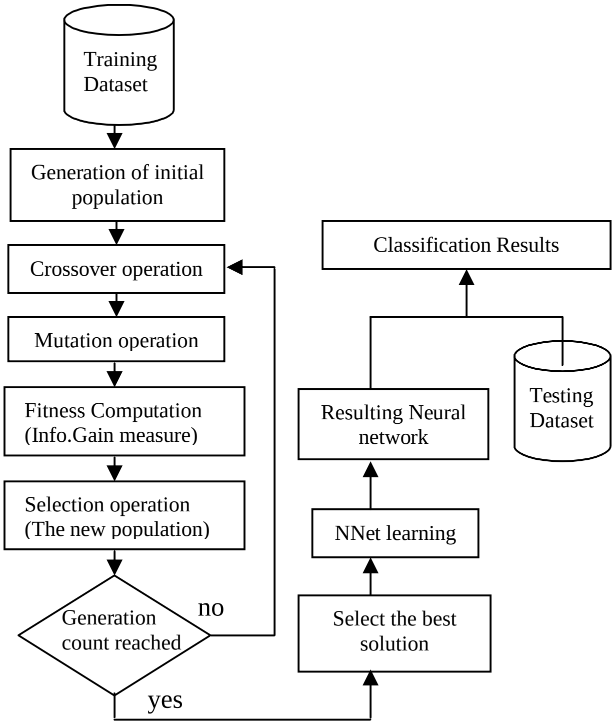

Figure 1: General schema of the proposed method

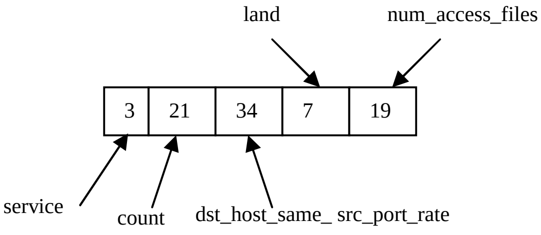

Figure2: Chromosome representation

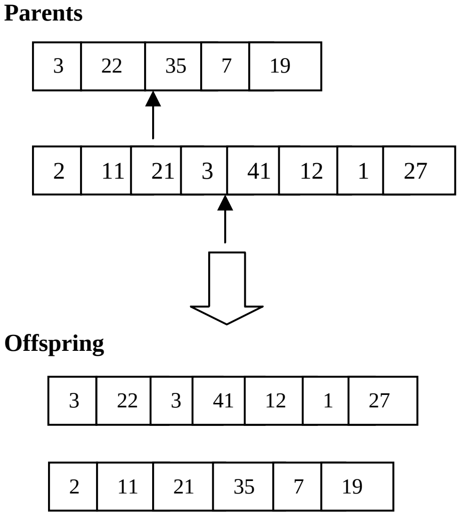

Figure 3: The proposed crossover operator

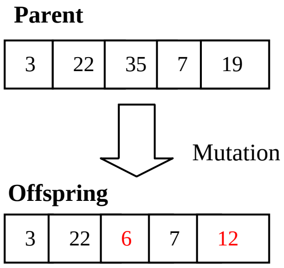

Figure 4: The mutation operator.

3.2.4 The fitness function

As mentioned above, the fitness value of a given features subset is proportional to its information gain. In [27], the computation of the information gain for only one feature according to the classes is proposed like the following:

Let S be a set of training set samples with their corresponding labels. Suppose there are *m* classes and the training set contains s_i samples of class *i* and s is the total number of samples in the training set. Expected information needed to classify a given sample is calculated by:

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log 2(\frac{s_i}{s}) \qquad (1)$$

A feature F with values { f_1, f_2, …, fv } can divide the training set into v subsets { S_1, S_2, …, S_v } where S_j is the subset which has the value f_j for the feature F. Furthermore let S_j contain s_{ij} samples of class i. Entropy of the feature F is:

$$E(F)=\sum_{j=1}^{v}\frac{s_{1j}+s_{2j}+......+s_{mj}}{s}\,.\,I(s_{1j},s_{2j},....,s_{mj})\quad(2)$$

Information gain for F can be calculated like the following :

$$\text{Gain}(F)=I(s_1,s_2,\ldots.,s_m)\text{-}E(F)\qquad(3)$$

The value of the gain as shown in (3) gives the information gain of a feature F with regard to all the classes. If we want to measure the gain of the feature for a given class k, we shall consider the problem as binary classification one. We consider two classes: the class k, and the remaining of classes will constitute one other class. So the new Expected information needed to classify a given sample will be:

$$I(s_k,s_{k'})=-\left(\frac{s_k}{s}\log 2(\frac{s_k}{s})+\frac{s_{k'}}{s}\log 2(\frac{s_{k'}}{s})\right)\quad(4)$$

Were k' denote the complemented class of the class k. The entropy of a feature F according to the class k is:

$$E(F)=\sum_{j=1}^{v}\frac{s_{kj}+s_{k'j}}{s}\,.\,I(s_{kj},s_{k'j})\qquad(5)$$

Information gain for F can be calculated as:

$$\text{Gain}(F)=I(s_k,s_{k'})\text{-}E(F)\qquad(6)$$

This gain measure allow only to compute the gain information value for individual features, we have extend it to allow computation of the gain provided by a collection of p features $(F^1,F^2,\ldots,F^p)$ as presented in the following:

Let $C=(F^1,F^2,\ldots,F^p)$ be a subset of features selected from the n existing ones. We want to compute the information gain provided by C with respect to the labelled dataset S.

If each feature F^i has v_i values $VF^i=\{f_1^i,f_2^i,\ldots.,f_{vi}^i\}$, to each combination $(F^1,F^2,\ldots.,F^p)$ we can associate the set of all the possible combinations values:

$$V_C=\left\{<f^1_{k_1},f^2_{k_2},......,f^p_{k_p}>\,/\,\forall f^1_{k_1}\in VF^1,\forall f^2_{k_2}\in VF^2,.......,\forall f^p_{k_p}\in VF^p\right\}$$
$$\text{when}: 1\le k_1\le v_1, 1\le k_2\le v_2,.........,1\le k_p\le v_p\qquad(7)$$

It is clear that V_C contain $v=\prod_{i=1}^{p}v_i=v_1.v_2.v_3\ldots.v_p$ possible combination, so a subset of features $(F^1,F^2,\ldots.,F^p)$ divide the training set into v subsets $\{S_1,S_2,\ldots.,S_V\}$, were S_j is the subset which has the combination of values of the j^th combination $V_C(j)$ from the set V_C for the each feature F^i from C.

Furthermore let S_j contain s_{ij} samples of class i (all the samples that have the values of the j^th combination from V_C for each feature F^i). Entropy of the feature subset $C=(F^1,F^2,\ldots.,F^p)$ can be computed like the following:

$$E(C)=\sum_{j=1}^{\prod_{i=1}^{p}v_i}\frac{s_{1j}+s_{2j}+......+s_{mj}}{s}\,.\,I(s_{1j},s_{2j},....,s_{mj})$$
$$(8)$$

when s_{ij}= Cardinality $(\{V_C(j)\in \text{Class } i\})$

Information gain for C can be calculated as:

$$\text{Gain}(C)=I(s_1,s_2,\ldots.,s_m)\text{-}E(C)\qquad(9)$$

Using the same principal of the expressions (4) and (5), the information gain for each features combination C with respect to a given class k can be deduced as follow:

$$E_k(C)=\sum_{j=1}^{\prod_{i=1}^{p}v_i}\frac{s_{kj}+s_{k'j}}{s}\,.\,I(s_{kj},s_{k'j})$$

and $\text{Gain}_k(C)=I(s_k,s_{k'})\text{ - }E_k(C)$ (10)

The value of the information gain is always bounded by the value of the expected information to classify a given record. It is always preferable to have a fitness function limited in the interval [0,1], this can help to stop the genetic process when the value of the best chromosome reaches the maximum possible one. For this reason, we have defined the normalised information gain measure as the following:

$$\text{NormGain}(C)=\frac{\text{Gain}(C)}{I(s_1,s_2,...,s_m)}\quad(11)$$

So the fitness of a chromosome C is equal to its normalized information gain. The best chromosome is the one who give the maximum information gain according to the dataset.

In the present work, two possible optimization processes has been implemented: we can search the best features subset that maximise the information gain for all the classes, or we can find the best one that maximise the gain for a given class. In the first case, the obtained solution (the subset) must be used to train a neural network that learns to discriminate between classes. In the second case, the evolved neural network learns to discriminate the class used for fitness from the other classes.

3.2.5 Genetic parameters

The selection process used during the genetic evolution is the roulette wheel selection, combined with elitism strategy. In this work, the usual GA parameters are used, with some new introduced ones to control the quality of the obtained solution. The Table 3 summarise the different parameters used.

When the genetic search is achieved, the resulting features subset (optimal solution) is used to train a multi layer neural network. It is clear that the subset with maximum information gain will lead to an optimal learning and so to a best classification rates. The following section details the neural network learning process implemented in this work.

Parameter	**Default value**	**Signification**
Generation Count	50	Maximum number of generations
Population size	100	Number of chromosomes created in each generation
Crossover rate	0.7	Probability of crossover
Mutation rate	0.1	Probability of mutation
Maximum chromosome length	15	Specify the maximum number of features that must be used in each chromosome (features subset size)

Table 3: Parameters set used for the genetic process

3.2.6 Neural network training

Neural networks have been identified since the beginning as a very promising technique of addressing the intrusion detection problem. Many researches have been performed to this end, and the results varied from inconclusive to extremely promising.

In the present work, we use a multi-layered neural network, with the error backpropagation learning algorithm to classify DoS attacks. The network has outputs and inputs according to the used features (input), and the desired classes (outputs).

Two different architectures are used in our implementation, representing the two possible classification goals:

- Discrimination between the 6 different DoS attack classes;
- Discrimination of one given class from the others.

The first architecture is used in conjunction with the fitness function computed using the gain of the expression (9), the number of outputs correspond to the number of the existing classes (6 in the case of DoS attacks), each output neuron is set to 1 if the input sample belong to its corresponding class, the others are so set to 0. For the second case, the expression (10) is used to compute the fitness. The number of outputs is two, corresponding to the desired class, or to the remaining classes. Each output neuron is set to 1 if the input sample belongs to the class k, and to 0 otherwise. The number of hidden layers is fixed to 2 layers in the two architectures, each hidden layer contain 40 connected neurons. The only parameter that controls the learning process in this work is the number of epochs with a default value of 100 epochs.

4 Results and discussion

In the following, the obtained results using the presented approach are presented. Two different tests were performed in the present work:

- The classification of all the DoS attack classes with a single neural network (multi-category classification),
- The discrimination of a single class from all the others (binary classification).

In the two cases, a genetic algorithm is used as explained above to find the most appropriate features subset that discriminate at maximum the desired class. To validate the obtained results, a neural network is trained using the whole existing features, and another using the most single discriminator feature (according to eq. (3)).

4.1 Result of the genetic search

Using the parameters presented in the Table.3, the following results were obtained by taking the best solution after 50 GAs trials. The Table.4 give the best solutions obtained for each class (in the case of binary classification), and for the whole classes (in the case of multi-classification). In the two situations, the fitness of the best solution is indicated with the corresponding information gain. The Table.6 shows the best single feature and its corresponding information gain value according to eq.(3). The Figure.5 shows the evolution of the fitness value during the genetic evolution for each classification case.

Class	Best Solution	Fitness value	Info. Gain
Neptune	dst_host_srv_count, srv_serror_rate service, dst_host_diff_srv_rate dst_host_same_srv_rate	0.9999	0.6838
Smurf	src_bytes, root_shell, wrong_fragment, dst_host_srv_serror_rate	1	0.6858
Back	dst_host_count, protocol_type src_bytes	0.9999	0.3312
Teardrop	dst_host_same_srv_rate is_host_login, flag, rerror_rate protocol_type	1	0.1513
Pod	wrong_fragment, service dst_host_srv_serror_rate, serror_rate	0.9999	0.0504
Land	Land, Dst_bytes, Srv_count	0.9999	0.0073
All Classes	src_bytes, root_shell, num_compromized count, is_guest_login, dst_host_rerror_rate	0.9105	1.0586

Table 4: Obtained best features subsets for the DoS attack classes

	Classification rate (using training data)		Classification rate (using testing data)	
	Feature subset	Individual feature	Feature subset	Individual feature
Neptune	97,89 %	95,2 %	97,0 %	92,0 %
Smurf	99,9 %	98,9 %	96,5 %	91,5 %
Back	97,1 %	92,1 %	95,31%	90,31%
Teardrop	99,9 %	98,07 %	99,9 %	93,9 %
Pod	98,82 %	97,01 %	95,45 %	94,45 %
Land	100 %	98.84 %	99,81 %	95,84 %

Table 5: Obtained classification rates for the 6 classes using generated features subsets and individual best features.

It is clear from the tables above that the information gain provided by single feature is not very different from the one obtained with the best features subset but just in the case of individual classes, this is not the case when considering the multi-category classification problem, when the value of the gain provided by the feature subset is clearly better than the individual one (provided by the "count" feature).

Another important remark is that the convergence of the learning error is very fast when using subsets with respect to individual ones, the neural networks are more adapted for multiple inputs than single one.

4.2 Training and Test results

As mentioned above, the obtained feature subset is used to train the neural network. The testing dataset was extracted from the 'Corrected (Test)' set available as testing set with the KDD99 corpus, and used with almost all the IDS implementations and its contain 229853 DoS record.

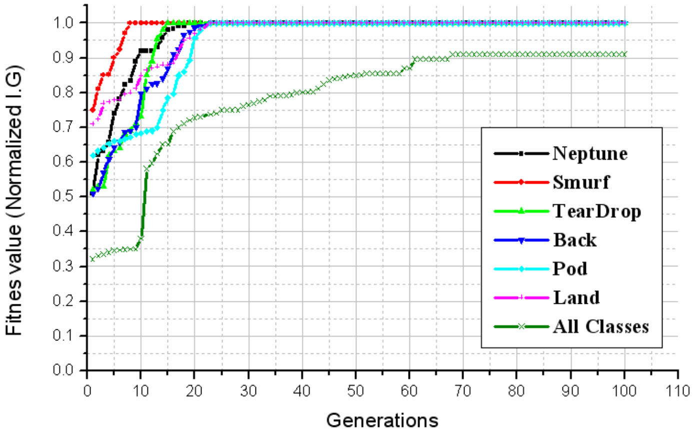

Figure 5: Evolution of the fitness value during genetic evolution for each classification case.

4.2.1 Individual class's discrimination

The first attempt was to discriminate each class from the others (binary classification). The architecture presented in 2.2.6 was used with each class separately based on the features included in the corresponding subset generated genetically. The Table.5 summarizes the obtained classification rates for each class for both training and testing datasets using: the generated pseudo-optimal features subset and the best individual features. We can see that the obtained performances using the features subsets outperform largely those obtained with individual ones, even if they have similar information gain values. This is due to the fact that neural networks are more adapted to multiple outputs than single ones. The Figure.6 illustrates the evolution of the learning error during training epochs.

4.2.2 Multiple classes' discrimination

The second attempt of this work is the most important since it deals with the problem of multiples classes' discrimination (multi-category classification). The proposed neural network architecture for this problem is trained using the inputs provided from the training dataset according to the feature subset generated genetically in the previous step. The goal is to discriminate each class from others using a single network.

Class	Best feature	Normalized I.G (Fitness)	Info. Gain
Neptune	dst_host_srv_serror_rate	0.9924	0.6786
Smurf	src_bytes	0.9769	0.6701
Back	src_bytes	0.9909	0.2853
Teardrop	protocol_type	0.8083	0.1223
Pod	src_bytes	0.9722	0.0491
Land	land	0.9998	0.0073
All Classes	count	0.7681	0.9025

Table 6: Obtained results using individual features

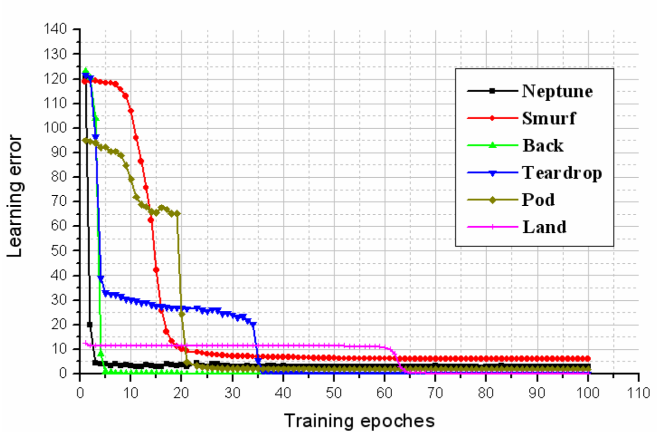

Figure 6: Evolution of the learning error during epochs for the individual DoS classes

The Table.6 show the classification performances obtained when using:

- The single best discriminator feature (taken from Table .4);
- The best features subset generated genetically;
- The whole set of existing features (the 41 KDD99 features set).

It is clear from the Table 6 that the generated features subset can achieves very acceptable performances with respect to other approaches in a reasonably acceptable run time. The Figure.7 shows the evolution of the learning error rate during training epochs in the three situations.

	Rate using Training Dataset	Rate using Testing Dataset	Run time
Single feature	68,54%	61.03%	36 m 22 s
Generated subset	98.73 %	92.42 %	2h 08m 15s
Whole features set	99.01%	93.12%	12h 37m 2s

Table 6: Classification performances using the different inputs cases

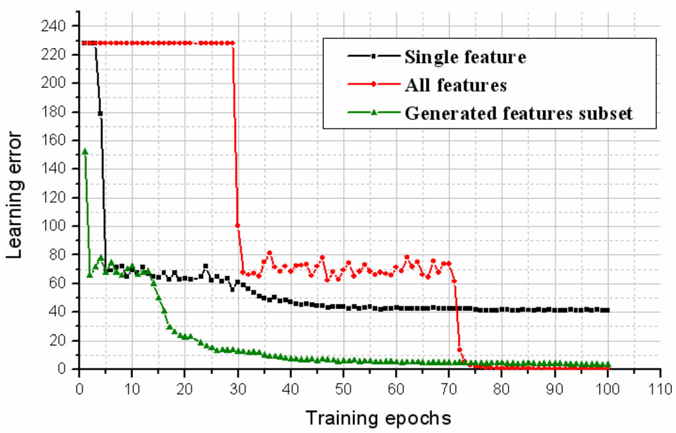

Figure 7: Evolution of the neural network learning error

In order to clearly show the efficacy of the genetic search and the generated solutions, the figure 8 shows the variation of the final learning error for some selected solutions (subsets) with respect to their fitness value. We can see that the error is inversely proportional to the fitness value, this proof that the search process is really guided by the efficacy of the classification system.

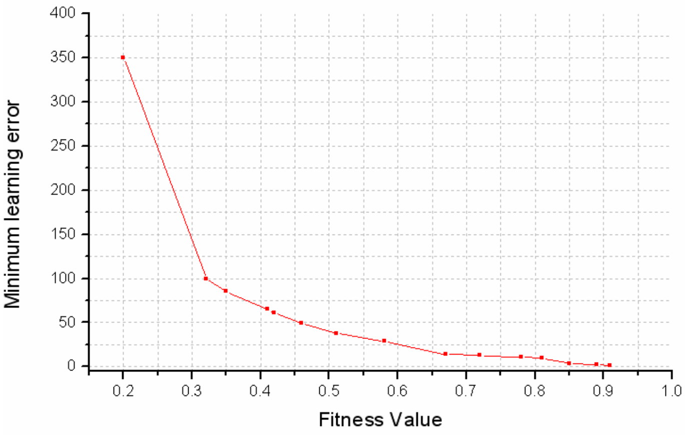

Figure 8: Variation of the minimum learning error with respect to the fitness value

5 Conclusion

In this work, a new approach for selecting best discriminates features subset using genetic algorithms is presented. The goal is to select the best combination that is sufficient to perform a good classification and obtain acceptable rates. This task can not be realised with any iterative or exhaustive approach, so we have use an evolutionary genetic algorithm to explore the huge space of all possible features subsets. To drive this search process correctly to the best solution, a new measure of subset quality is proposed and used as fitness function. The evolved solution is used finally in conjunction with neural network to perform the training process. The obtained result shows that the genetic search can find very acceptable solutions for this problem in an acceptable run time. The quality of the solution is also shown to be sufficient to ensure a good discrimination between the DoS attack classes in the two studied situations: binary and multiple classification problems.

In term of feature works, this approach should be tested with other various datasets with different dimensions. The problem of dataset distribution must also be studied in more depth, as changing the proportionally of each class in the training dataset is shown to change radically the features selection and the classification results.

References

[1]. J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. In IEEE Intelligent Systems, volume 13, pages 44–49, 1998.

[2]. G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problems. In the 11th International Conference on Machine Learning, pages 121–129, 1994.

[3]. R. Kohavi, and G. John. Wrappers for feature subset selection. Artificial Intelligence journal, volume 97. Special issue on relevance, pp 273-324. Dec. 1997.

[4]. D. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, 1989.

[5]. J. Holland. Adaptation in Natural and Artificial Systems, MIT Press, 1992.

[6]. L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In 16th ICPR, pages 568–571, 2002.

[7]. A. Jain and D.Zongker. Feature selection: evaluation, application and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence 19.pp153 -158, (1997).

[8]. H. Handels, Th.Rob, J.Kreusch, H.Wolff, and S. Popple. Feature Selection for Optimized Skin Tumor Recognition using Genetic Algorithms. Artificial Intelligence in Medicine, 1999. pp:283-297.

[9]. F. Brill, D. Brown, and W. Martin, Fast Genetic Selection of Features for Neural Network Classifiers. IEEE Transactions on Neural Networks, 1992. pp:324-328.

[10]. H. Vafaie, Kenneth D.J., Feature Space Transformation Using Genetic Algorithms. IEEE Transactions on Intelligent Systems, 1998. pp57-65.

[11]. S. Ho, C. Liu, and S. Liu, Design of an Optimal Nearest Neighbours Classifier using an Intelligent Genetic Algorithm. Pattern Recognition Letter, 2002. pp 1495-1503.

[12]. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. J. Smola and K. R. Müller. “Invariant feature extraction and classification in kernel spaces”, Advances in Neural Information Processing Systems, Massachusetts, USA: MIT Press, vol. 12, pp. 526-532, 2000.

[13]. D. Eads, D. Hill, S. Davis, S. Perkins, J. Ma, R. Porter and J. Theiler, “Genetic algorithms and support vector machines for time series classification”, 5th Conference on the Application and Science of Neural Networks, Fuzzy Systems and Evolutionary Computation, pp. 74-85, 2002.

[14]. J. Sepulveda-Sanchis, G. Camps-Valls, E. Soria-Olivas, S. Salcedo-Sanz, C. Bousono-Calzon, G. Sanz-Romero and J. Marrugat, “Support vector machines and genetic algorithms for detecting unstable angina”, Computers in Cardiology, IEEE Computer Society Press, Menphis, USA, 2002.

[15]. J. Liu, H. Iba and M. Ishizuka, “Selecting informative genes with parallel genetic algorithms in tissue classification”, Genome Informatics, vol. 12, pp. 14-23, 2001.

[16]. V. D. Nguyen and D. M. Rocke, “Tumor classification by partial least squares using microarray gene expression data”, Bioinformatics, vol. 8, no. 1, pp. 39-50, 2002.

[17]. L. Boudjeloud and F. Poulet. A genetic approach for outlier detection in high dimensional data sets. In Modelling, Computation and Optimization in Information Systems and Management Sciences, MCO’04, pages 543–550. Le Thi H.A., Pham D.T. Hermes Sciences Publishing, 2004.

[18]. Kelly, J.D., Davis, L.: Hybridizing the genetic algorithm and the K nearest neighbour classification algorithm. In Belew, R.K., Booker, L.B., eds.: Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, Morgan Kaufmann (1991) pp:377-383.

[19]. Punch, W.F., Goodman, E.D., Pei, M., Chia-Shun, L., Hovland, P., Enbody, R.: Further research on feature selection and classification using genetic

algorithms. In Forrest, S., ed.: Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, CA, Morgan Kaufmann (1993) pp:557-564.

[20]. Raymer, M.L., Punch, W.F., Goodman, E.D., Kuhn, L.A., Jain, A.K.: Dimensionality reduction using genetic algorithms. IEEE Transactions on Evolutionary Computation 4 (2000) pp:164-171.

[21]. Inza, I., Larranaga, P., Etxeberria, R., Sierra, B.: Feature subset selection by Bayesian networks based optimization. Artificial Intelligence 123 (1999) pp:157-184.

[22]. Cantu-Paz, E.: Feature subset selection by estimation of distribution algorithms. In Langdon, W.B., Cantu-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A.C., Miller, J.F., Burke, E., Jonoska, N., eds.: GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, San Francisco, CA, Morgan Kaufmann Publishers (2002) pp:303-310.

[23]. Guyon, I., Elissee, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3 (2003) pp: 1157-1182.

[24]. The 1998 intrusion detection off-line evaluation plan. MIT Lincoln Lab., Information Systems Technology Group.
http://www.11.mit.edu/IST/ideval/docs/1998/id98-eval-11.txt, March 1998.

[25]. Knowledge discovery in databases DARPA archive: Task Description. http://www.kdd.ics.uci.edu/databases/kddcup99/task.html

[26]. K.M. Faraoun and A. Boukelif: "Genetic Programming Approach for Multi-Category Pattern Classification Applied to Network Intrusions Detection". International Journal of Computational Intelligence and Applications (IJCIA), ISSN: 1469-0268. Volume 6, Number 1, pp 77-99. Wold scientific publishing (UK) Ltd & Imperial College Press, March 2006.

[27]. Hilmi Günes Kayacik, A. Nur Zincir-Heywood, Malcolm I. Heywood. Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99. Third Annual Conference on Privacy, Security and Trust (PST) October 12-14, 2005, The Fairmont Algonquin, St. Andrews, New Brunswick, Canada.