An Evolutionary Improvement of the Mahalanobis – Taguchi Strategy and Its Application to Intrusion Detection

Dimitris Liparas and Evangelia Pantraki

Department of Informatics
Aristotle University of Thessaloniki
54124, Thessaloniki, Greece
{dliparas,pantraki}@csd.auth.gr

Abstract. The Mahalanobis - Taguchi (MT) strategy is a statistical methodology combining various mathematical concepts and is used for diagnosis and classification in multidimensional systems. MT is a very efficient method and has been applied to a wide range of disciplines so far. However, its feature selection phase, which uses experimental designs (orthogonal arrays), is susceptible to improvement. In this paper, we propose a methodology that incorporates MT and a Genetic Algorithm (MT-GA), with the latter being used both for optimizing the feature selection step of MT and for determining the most suitable training set. As an application domain for testing the proposed methodology, we utilized Intrusion Detection Systems (IDS). IDS play an increasingly important role in network security technology nowadays and more and more research is being directed towards building effective diagnostic models. We test the effectiveness of MT-GA by applying it to a well-known intrusion detection dataset and by comparing its performance to that of the typical MT strategy and of other classifiers. The results indicate the benefits of using MT-GA.

Keywords: Mahalanobis – Taguchi strategy, Genetic algorithm, Data mining, Intrusion detection systems.

1 Introduction

The Mahalanobis-Taguchi (MT) strategy, proposed by Genichi Taguchi, is a rather new methodology used for diagnosis and classification in multidimensional systems [35]. It combines various mathematical and statistical concepts like Mahalanobis distance (MD), Gram-Schmidt orthogonalization process and experimental designs. Principally, the MT strategy develops a univariate measurement scale from a group of "normal" or "healthy" observations. The scale is then used to signify the level of abnormality of a set of "abnormal" cases in reference to the "normal" population. The scale is assessed for its ability to effectively recognize the abnormal cases in the way that larger values of the scale indicate abnormality. The selection of an optimal subset of the most important variables from the original variable set is a very essential part of the MT strategy.

L. Iliadis, M. Papazoglou, and K. Pohl (Eds.): CAiSE 2014 Workshops, LNBIP 178, pp. 16–30, 2014.

Feature selection as a procedure can influence the classification process in various ways. [40] indicate that the choice of features used to represent patterns that are presented to a classifier affects, among other things, the accuracy of the learned classification algorithm, the time needed for learning a classification function, the number of examples needed for learning and the cost associated with the features. Within the framework of the typical MT strategy, feature selection is achieved with the use of experimental designs (orthogonal arrays-OA) and a measure called signal-to-noise (S/N) ratio. Through the selection of the most useful variables the final measurement scale is formulated. Although the primary goal of the methodology is the construction of a measurement scale, the strategy can also serve as a binary classifier with the determination of a proper abnormality threshold.

Despite the fact that the MT strategy has been successfully applied to an extensive range of disciplines, such as manufacturing and financial crisis forecast (see for example [5,19]), [39] have provided a review, analysis and criticism of MT strategy. Among other things, they illustrate that the use of OA and experimental design methods does not ensure that the optimal feature subset is obtained. Moreover, they suggest that the MT strategy could be modified to include a better dimensionality reduction algorithm. In a response discussion to [39], [1] express doubts about whether the OA method is the most suitable for feature selection within the MT strategy. A solution to these concerns and suggestions would be the development of a hybrid methodology, consisting of the MT strategy and a feature selection technique as a replacement for the OA-S/N ratio method. [24] have proposed an approach that uses binary particle swarm optimization for the feature selection problem.

In this study, we propose the use of a Genetic Algorithm (GA) for optimizing the procedure of selecting the most useful variables. In addition, we proceed further in the application of the GA by formulating a GA-based refinement of the training set as well. In other words, the GA is used both for selecting the optimal variable subset and for extracting the best training subset, in terms of classification accuracy. In order to explore the potential of the proposed hybrid methodology, we opted to apply it to the intrusion detection domain.

With the enormous growth of computer networks usage and the huge increase in the number of applications running on top of it, network security is becoming increasingly more important [36]. Intrusion detection systems (IDS) are devices that monitor network or system activities for known or potential threats and make reports on any intrusion attempts. A system that possesses all the features of an IDS, but also has the ability to try and stop intrusions, is called intrusion prevention system (IPS). Because of the increasing dependence on information systems and the potential impact of intrusions against those systems, IDPSs (Intrusion detection and prevention systems) have become a necessary addition to the security infrastructure of nearly every organization [29].

Over the years, a wide range of machine learning techniques have been applied, in order to propose and build efficient IDS. [37] have provided a thorough literature review of the different methodologies employed for intrusion detection purposes. Among others, Neural Networks [28], Naïve Bayes [30], Decision Trees [7] and Support Vector Machines [41] can be listed. Other studies focus on the idea of proposing hybrid (for example [31]) or ensemble classifiers (see for instance [23,8]) that combine several algorithms, in order to improve the performance of an intrusion detection model.

In order to evaluate MT-GA in the intrusion detection domain, we apply it to a benchmark intrusion detection data set. However, our intention is not to limit the use of the method simply as an intrusion detection model, but to provide a way of demonstrating the benefits of using MT-GA, by utilizing intrusion detection as an application domain. The application results show that it compares successfully to other MT-related and intrusion detection methods.

The rest of the paper is organized as follows: In Section 2 we provide the theoretical background of our study. We describe the notions and steps of MT strategy, as well as some basic GA concepts. In Section 3 we present the proposed MT-GA methodology. In Section 4 we provide the experimental results from the application of MT-GA and of other methods to the NSL-KDD intrusion detection data set. Finally, concluding remarks are provided in Section 5.

2 Theoretical Background

In 2.1 and 2.2 we present briefly the MT strategy's key features and operational steps. For more details see ([35]).

2.1 The Mahalanobis Distance

The Mahalanobis distance (MD) forms the basis of MT. In the MT strategy's context, there are two different ways to calculate the MD: the Mahalanobis Taguchi System (MTS) and Mahalanobis Taguchi Gram-Schmidt process (MTGS). Here we work only with the MTGS version of the strategy, because of the fact that it can handle cases of multicollinearity (a term that refers to strong linear correlations between the variables). Hence, in the MTGS version of the MT strategy the MD is calculated as follows:

Assume that a sample dataset consists of k variables and n cases (the size of the sample). Let x_{ij} be the value of the ith variable ($i = 1, ..., k$) on the jth case ($j = 1, ..., n$). These variables are standardized by

$$z_{ij} = (x_{ij} - m_i)/s_i \tag{1}$$

whereby m_i and s_i we denote the sample mean and standard deviation respectively of the ith variable.

The matrix of the standardized values z_{ij} is then transformed by the Gram-Schmidt orthogonalization process ([15]) and the MD is calculated using the derived Gram-Schmidt vectors u_{ij} as in the following equation:

$$MD_j = (1/k)\sum_{i=1}^{k}(u_{ij}^2/s_i'^2) \tag{2}$$

whereby s_i' we denote the standard deviation of the ith variable after the application of the Gram-Schmidt orthogonalization process. MD_j expresses the distance of the jth case from a reference (zero) point.

2.2 Steps of MT Strategy

The MT strategy consists of the following steps:

Step 1: Construction of the measurement scale.
First we must determine the variables that define which cases are considered "healthy" or "normal" and which are not. Then we collect all the necessary data from all the variables in the data set. Next, we standardize the values of the variables. Finally, we compute the Mahalanobis distance (MD) values only for the healthy cases using the Gram-Schmidt orthogonalization process.

Step 2: Validation of the scale.
In the next step, we compute the MD values for the "abnormal" cases (observations that do not belong to the "healthy" group). The values of the variables in these "abnormal" cases are standardized by using the mean values and standard deviations of the corresponding variables from the "healthy" group. The effectiveness of the scale is validated with the use of a simple rule: the MD values for the "abnormal" observations must be higher than those for the "healthy" ones (they must be further from the center of the group of "healthy" cases than any of the "healthy" cases are).

Step 3: Feature selection.
In this step, we use orthogonal arrays (OAs) and signal-to-noise (S/N) ratios in order to determine and select the most important variables from the original variable set.

An OA is an experimental design matrix ([12]). Each row represents an experimental run and contains the levels of various factors in order to study the effects of the factors on a prespecified response variable. Each column of the OA represents a factor of the experiment. In the context of MT strategy, we consider the inclusion or exclusion of each variable as a factor with two levels. Hence, the experiment considers k factors and the level of a factor shows when a variable participates or not in the analysis. For each of these runs, the MD values are calculated for the "abnormal" cases as in step 2, but using only the specified variables. The MD values are then used to calculate the value of a S/N ratio, which is the response variable for each different run. The set of the most

important variables is obtained by computing and evaluating the gain in the values of the S/N ratios. For a variable, the gain in S/N ratio is the difference between the average S/N ratio when the variable is used in OA and the average S/N ratio when the variable is not used in OA. If the gain is positive, then that variable is useful ([34]).

Step 4: Reconstruction of the scale.
After selecting the subset of the most useful features, the measurement scale is reconstructed using only these variables. The reconstructed scale is used to compute the MD values for any unknown cases, in order to take any corrective actions, if necessary. The final measurement scale can be used for future diagnosis after defining an appropriate threshold.

2.3 Genetic Algorithm

The Genetic algorithm (GA) is a part of the field of evolutionary computing, which is an artificial intelligence area dealing with combinatorial optimization problems. Its theory is based on the process of natural selection and uses procedures inspired by biological evolution (mutation, selection and crossover). The basic idea behind GA is that it tries to evolve a group of candidate solutions (*population*) towards a globally optimal solution. The possible solutions are expressed by coded representations called *chromosomes*. Although there are quite a few encodings for the chromosomes, most usual is the binary one ([38]). In the binary encoding, every chromosome corresponds to bits of 0s and 1s. For example, in the feature selection context, the value 0 shows that a feature does not participate in the computations and the value 1 shows that it does. In this way, a candidate solution can easily be transformed into its representing chromosome.

The process of optimizing the problem's solution is evolved through a number of successive computational runs of the algorithm, called *generations*. In each generation, after all the candidate solutions have been processed, their *fitness* is evaluated by an appropriate function, called *fitness function*. The best solutions (in terms of fitness) are selected, in order to generate the next generation's population. There are many methods for the selection of the fittest solutions, such as roulette wheel selection, tournament selection, elitism and others. Elitism is the process of retaining some of the best solutions unaltered to the next population. This method increases the performance of GA and enables it to converge quicker to the optimal solution. In the next step, genetic operators such as *crossover* and *mutation* are used for the creation of the next population. In crossover, we exchange parts between two or more chromosomes (called *parents*) in order to produce a new chromosome (called *offspring*). A user-defined parameter called *crossover probability* or *crossover rate* indicates the frequency of the crossover operator. There are many crossover techniques, such as one point crossover, two point crossover and others. In mutation, parts of the chromosome are altered with a new chromosome as a result. In binary encoding, for example, bit values of 1 are changed to 0 and vice versa. Again, a parameter called *mutation probability* or *mutation rate* influences the frequency of the mutation operator.

The GA is executed until a stopping condition is satisfied (for instance, the maximum number of generations is reached or the fitness function's value no longer improves). For more details about the basic notions of GA, see ([9]).

In the relevant literature, there are various GA applications for feature selection (see for example [13,22]). Moreover, we find several studies dealing with hybrid GA-based methodologies in intrusion detection (see for example [32,18]).

3 Description of the Proposed MT-GA Methodology

The proposed MT-GA methodology is largely based on US Patent 2006/0230018 A1, published by [11]. The inventors describe a method that, among other things, is suitable for identifying a desired variable subset from a plurality of variables. The basic notion of the method is the application of a genetic algorithm on a set of data records, which have been defined as "normal" data or "abnormal" data based on predetermined criteria. The Mahalanobis distance values for the "normal" and "abnormal" data are calculated using the variable subsets derived from the application of the genetic algorithm. The fitness function evaluates the deviation of the Mahalanobis distance values between the "normal" and "abnormal" data records. The goal of the genetic algorithm is to maximize this deviation. Predetermined parameters, like an improvement rate, may be used to decide whether the genetic algorithm converges to the optimized variable subset.

The approach of the above-described procedure was incorporated into our MT-GA methodology. However, we expanded the idea by not only applying the genetic algorithm to the variables of the training data set (MT-GA$_{variables}$), but to the cases as well (MT-GA$_{variables-cases}$). In this way, we are able to reduce the size of the training set substantially, while at the same time aiming to maximize the classification accuracy of the model. The issue of suitably reducing the number of cases in the training set was addressed in two previous studies ([20,21]). We proposed the use of the MT strategy in conjunction with two clustering algorithms, the two step cluster analysis and the PAM (Partitioning Around Medoids) algorithm. For details about the two step cluster analysis and the PAM algorithm see ([4,16]) respectively. The two clustering algorithms are used in our studies for determining the most suitable training set, in terms of training classification accuracy, with very satisfactory results. In this study, by choosing to apply the GA to the training set's group of cases, we are essentially aiming to expand this notion. The procedures of MT-GA are described as follows.

Regarding MT-GA$_{variables}$, we start by setting up the GA with a proper input parameter set (population, number of generations, crossover rate, mutation rate, etc). We then proceed to initialize the GA feature selection procedure with the random selection of a population of candidate solutions and the encoding of the solutions into the equivalent chromosomes. In our case, the binary encoding method was used. Therefore, a chromosome bit of value 1 indicates that the corresponding variable is selected and a bit of value 0 indicates that it is not. After the population of the chromosomes is formulated, we construct the MD measurement scales from the "healthy" cases, using the feature subset derived

from each chromosome. Then the measurement scales are validated by computing the MD values for the training set's "unhealthy" cases. The GA's fitness function evaluates each variable subset. The basic idea behind the fitness function in our method is to minimize the overlapping MD distributions in the "healthy" and "unhealthy" groups. It is almost certain that, for a particular problem, it is very difficult to select an appropriate group of "healthy" cases, able to construct a MD measurement scale that effectively distinguishes between the "healthy" and "unhealthy" groups. As a consequence, we end up with MD distributions with overlapping regions, e.g. many "normal" observations with extreme MD values are identified as "abnormal" and vice versa. Hence, we designed a fitness function aiming to maximize the percentage of the "unhealthy" cases, whose MD values are higher than the maximum MD value of the "healthy" group. The equation for the fitness function is given as follows:

$$Fitnessfunction = (n(u > maxMDh)/n(u)) \tag{3}$$

where
$n(u > maxMDh)$ = number of "unhealthy" cases with MD values higher than the maximum MD value of the "healthy" group's cases
$n(u)$ = total number of "unhealthy" group's cases.

After the evaluation of the solutions, if a predetermined rule for the termination of the GA is not satisfied, the method proceeds to create the next generation's population. This is achieved with the use of the proper selection strategy and the application of the GA operators (crossover, mutation). When the algorithm has converged to the optimal variable subset, the procedure ends.

With the use of the optimal variable subset, we are able to construct the final MD measurement scale from the "healthy" cases and validate it with the "unhealthy" cases. Furthermore, we plot the training set's Receiver Operating Characteristic (ROC) curve ([17]) and based on it, we define an appropriate threshold, meaning that we choose a MD value that provides a reasonable trade-off between sensitivity and specificity (see Section 4.2). With the use of the threshold, we determine the classification accuracy results for the training and test sets.

Finally, regarding the MT-GA$_{variables-cases}$ method, after the GA application to the training set's variables and the determination of the optimal variable subset, we apply exactly the same GA procedure, but this time to the training set's cases. The flowchart of MT-GA$_{variables}$ is depicted in Figure 1.

4 Application of MT-GA Method to Intrusion Detection Data

4.1 Data Set Description − Experimental Design

KDDCUP'99 is an intrusion detection benchmark data set, having been extensively used during the last decade for the evaluation of intrusion detection

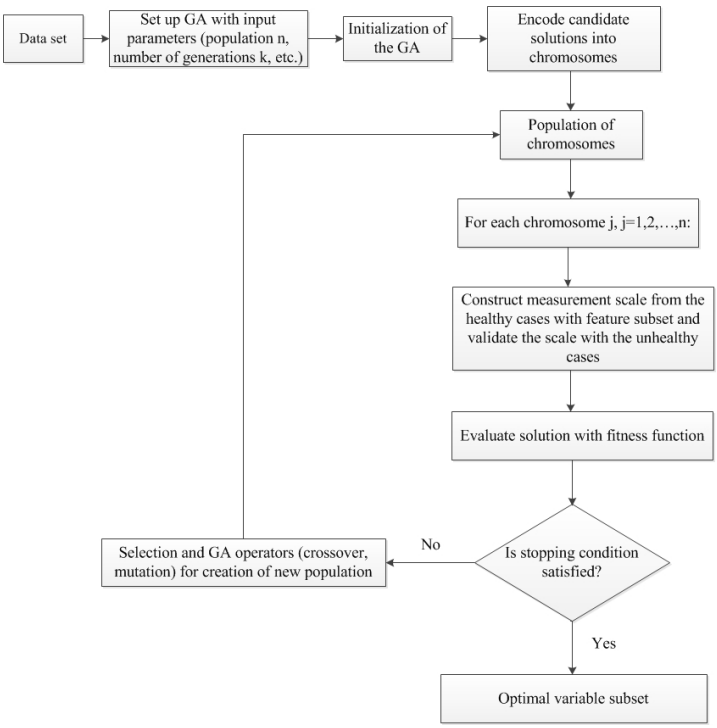

Fig. 1. Procedure of MT-GA$_{variables}$

systems. However, in a recent study [36] some of the underlying problems of KDDCUP'99 that affect the performance and evaluation of the systems have been highlighted. Hence, they proposed a new data set, NSL-KDD, which does not include any duplicate records in the training and test sets and therefore, the classifiers will not be biased towards more frequent records.

The NSL-KDD data set, as provided in http://nsl.cs.unb.ca/NSL-KDD/, is already split into training and test sets and consists of 25192 (13449 normal/"healthy" and 11743 attack/"unhealthy") training cases and 22544 (9711 normal/"healthy" and 12833 attack/"unhealthy") test cases. Each case is comprised of 41 features and 1 binary class attribute (normal/"healthy" or attack/"unhealthy"). More specifically, from the set of 41 variables/features, 38 are numeric and 3 are categorical. We note that in this study we chose to work only with the group of numeric variables.

Apart from applying the original MT and the proposed MT-GA methods to the NSL-KDD data set, we also employed the version that combines the original methodology with the PAM clustering algorithm (which is used for refining the training data set). In this way, we are able to make a direct comparison of the MT-GA results to the corresponding results of a methodology that employs similar concepts.

Finally, we selected a number of well-known machine learning classifiers, namely Naïve Bayes ([14]), J48 decision tree ([25]), Support Vector Machine (SVM) ([3]), Random Forest ([2]) and Multilayer Perceptron ([27]) and applied them to the NSL-KDD data set as well, in order to form a comparison basis to the MT-GA method. All the algorithms were implemented using functions in the statistical language R ([26]). We note here that for a fair comparison, the 5 machine learning classifiers were applied to the reduced data set that resulted from the application of MT-GA$_{variables-cases}$.

4.2 Accuracy Measures

For the evaluation of the performance of MT-GA, as well as of the other methods employed in this study, we used two basic accuracy measures, namely sensitivity and specificity.

Sensitivity in the study's context estimates the proportion of "unhealthy" or attack cases, which are correctly identified as such (i.e. number of true attack / (number of true attack + number of false normal)).

Specificity on the other hand estimates the proportion of "healthy" or normal cases, which are correctly identified (i.e. number of true normal / (number of true normal + number of false attack)). Moreover, we used the Relative Sensitivity (RS) measure, which was proposed by [33] and is defined as

$$RS = \frac{sensitivity}{specifity} \tag{4}$$

The above metric is used to determine whether a classifier's ability to predict the two different (positive and negative) classes is balanced or not. If the RS value is much higher or much lower than 1, then this is an indication that the method is biased. Generally, we are looking for RS values around 1.

Finally, for each classifier we plotted the ROC curve for the corresponding test set and used the Area Under the Curve (AUC) to measure the discriminability between the test set's different classes and consequently, to evaluate the performance of each method.

4.3 Genetic Algorithm Parameters

One issue concerning the GA's context is the selection of the optimal control parameters for a given problem. The performance of the GA is dependent on the configuration of these parameters. Although there have been some studies addressing this matter (see for instance [6,10]), there is no general theory or rules that one should follow.

In order to determine the best GA setting for our study, we opted to conduct several experiments with different input parameters for each one. Then we selected the setting that gave us the best results (in terms of training accuracy) and fitted best into the problem. The GA parameter setting that we selected to use is the following: population size 100, number of generations 150, crossover

rate 0.8, mutation rate 0.01, one-point crossover and elitism (20% of the population size). The stopping condition for the GA is that the algorithm reaches the maximum number of generations (150) or that the fitness function's value does not improve during the last 50 generations.

4.4 Experimental Results

Table 1 summarizes the results from the application of all MT-related methods described in the previous Sections. The second and third rows contain the number of variables and cases (respectively) in the training set, after the application of each method. We notice that:

- The original MT method achieves a 47.3% reduction in the number of variables (20 out of 38), due to the application of the OA-S/N ratio feature selection method. The same proportion of feature reduction is attained by MT-PAM.
- MT-GA$_{variables}$ reduces the size of the original variable set by 73.6% (10 out of 38).
- Regarding the cases, the application of the PAM clustering algorithm results in only a few discarded cases (54 cases out of a total of 25192).
- The application of GA to the cases, using only the optimal feature subset derived from the initial GA application to the variables of the training set, keeps only 8.6% (2158 cases out of 25192) of the training set's cases, i.e. we achieve a reduction of 91.4% in the cases.

Regarding the accuracy results for the training and test sets (fourth and fifth row of Table 1 respectively): MT-GA$_{variables-cases}$ is very robust when compared to the other methods. It outperforms them both in terms of training (96.1%) and test accuracy (91.8%), while using a greatly reduced training set. In the sixth and seventh row of Table 1 the sensitivity and specificity results

Table 1. Performance comparison for MT-related methods

Results/Method	**MT$_{original}$**	**MT-PAM**	**MT − GA$_{variables}$**	**MT-GA$_{variables-cases}$**
Number of variables (optimal subset)	20	20	10	10
Number of cases (training set)	25192	25138	25192	2158
Training accuracy	93.6%	93.7%	92.3%	96.1%
Test accuracy	84.7%	87.3%	91%	91.8%
Sensitivity	79.6%	84.2%	91.6%	93.4%
Specificity	91.4%	91.5%	90.2%	89.7%
AUC	0.947	0.951	0.949	0.954
Relative Sensitivity	0.87	0.92	1.01	1.04

for all MT-related methods are given. We observe that although the specificity performance of MT-GA$_{variables-cases}$ is a little bit lower than that of the other methods, the improvement in the sensitivity value (MT$_{original}$: 79.6% ; MT-GA$_{variables-cases}$: 93.4%) is evident. Finally, in the eighth and ninth rows of Table 1 we provide the AUC and the Relative Sensitivity results, respectively. The four methods perform very highly in terms of the AUC, yet we do not notice great differences between them. Furthermore, we see that the two MT-GA methods are almost perfectly balanced in their ability to predict the "healthy" and "unhealthy" cases, since they achieve RS values very close to 1.

In Table 2 we compare the results of MT-GA$_{variables-cases}$ (which performed the best among the four MT-related methods) to these of five well-known machine learning methods (namely Naïve Bayes, J48, Random Forest, SVM and Multilayer Perceptron). First of all, we observe that while Random Forest achieves perfect training classification accuracy, it does not perform equally well in the test accuracy measure. As for the other methods, they yield quite promising test classification accuracy results, but still MT-GA$_{variables-cases}$ outperforms them all. In general, the five machine learning methods predict the positive class (specificity values $> 90\%$) with greater effectiveness than the negative class (sensitivity values ranging from 69.6% - 88.7%). This is also obvious from their RS values, ranging from 0.75 to 0.85 (except for J48, whose RS value is very close to 1). Essentially, with the exception of the training accuracy and specificity measures, MT-GA$_{variables-cases}$ yields better results than the selected machine learning techniques in all performance aspects.

Table 2. Performance of MT-GA method vs. other machine learning algorithms

Results/Method	**MT-GA** $_{variables-cases}$	**Naïve Bayes**	**J48**	**Random Forest**	**SVM**	**Multilayer Perceptron**
Training accuracy	96.1%	95.1%	99.7%	100%	98%	99.8%
Test accuracy	91.8%	79.3%	89.9%	81.4%	81.1%	84.2%
Sensitivity	93.4%	69.6%	88.7%	73.3%	72.1%	78.3%
Specificity	89.7%	92.1%	91.5%	92.3%	92.9%	92%
AUC	0.954	0.858	0.884	0.941	0.825	0.936
Relative Sensitivity	1.04	0.75	0.97	0.79	0.77	0.85

In Figure 2, a bar chart depicting the test accuracy of each method, sorted in descending order, is displayed. It is evident that both of the MT methods that use GA perform in this regard better.

In order to test whether the test accuracy of all the methods under comparison has statistically significant differences, we conducted a Cochran's Q-test which is suitable for multiple related binary samples. Here the samples are essentially the results of classification (1=Correct/0=False) of each one of the 9 methods (4 MT-based methods and 5 machine learning classifiers). Since all methods are

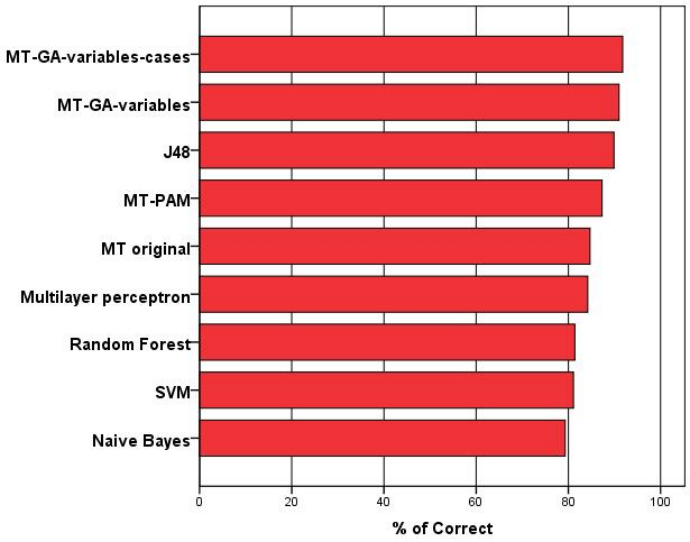

Fig. 2. Percentage of correct classifications of all methods (test accuracy)

Table 3. Results of the pairwise test for significant differences between method's test accuracy (MTGA/V/C: MT-GA$_{variables-cases}$; MTGA/V: MT-GA$_{variables}$)

	MTGA/V	**J48**	**MT-PAM**	**MT original**	**Multilayer Perceptron**	**Random Forest**	**SVM**	**Naïve Bayes**
MTGA/V/C	**no dif**	√	√	√	√	√	√	√
MTGA/V	–	**no dif**	√	√	√	√	√	√
J48		–	**no dif**	√	√	√	√	√
MT-PAM			–	**no dif**	**no dif**	√	√	√
MT$_{original}$				–	**no dif**	√	√	√
Multilayer Perceptron					–	√	√	√
Random Forest						–	**no dif**	√
SVM							–	√

applied to the same test set, the classification results are considered related. The test gave significance p<0.001, showing that overall, the 9 methods have statistically significant differences in their test accuracy. In order to locate the pairs of methods that are significantly different, we accompanied the Cochran's test with a pairwise test (as implemented in the statistical software SPSS/PASW). This pairwise test uses a correction for the significance of multiple comparisons in

order to control the family-wise error. The pairs of methods that have significant differences (adjusted significance<0.05) are shown with a tick in Table 3.

We notice that, for example, the test performance of Naïve Bayes has significant differences with all the other methods. It is also important to mention that MT-GA$_{variables-cases}$ is not significantly different only from MT-GA$_{variables}$. But despite this we must note that the application of MT-GA$_{variables-cases}$ has the advantage of reducing the size of the training set even more than MT-GA$_{variables}$.

5 Conclusions

In this study, we propose the hybrid MT-GA methodology and we explore its capabilities and potential by using it to build an effective intrusion detection model. By combining a decision-making methodology based on simple statistical methods (MT strategy) with a search heuristic (GA), we build an effective classification model that improves both the classification accuracy and the size of the data set needed for the training of the method (optimal variable subset and reduced number of training instances).

It is important to note that the advantage of MT strategy is that it does not treat "abnormal" cases as if they belong to the same population, different from the population of normal cases. Instead, it implies that each "abnormal" case, which in our study is an intrusion, is a unique case. With all the different types of intrusion events that exist nowadays, the method maps well to intrusion detection problems and its philosophy is in accordance with realistic situations. Another advantage of MT is its ability to construct a scale of abnormality so as to characterize the degree of severity of "abnormal" conditions, which is also quite realistic in the intrusion detection area.

There are various suggestions that can be made for future work. For example, alternative fitness functions can be used for the GA, in order to investigate the behaviour of the algorithm in different settings and conditions. Furthermore, we can integrate the MT strategy with other feature selection techniques and test the performance of each resulting model. It will provide some insight into the capabilities of our proposed methodology, as well as a more direct comparison basis. Finally, the application of MT-GA to data sets that contain noise and require further data cleaning (for example, software engineering data sets) should also be investigated.

References

1. Abraham, B., Variyath, A.M.: Discussion paper to "A review and analysis of the Mahalanobis-Taguchi system". Technometrics 45(1), 22–24 (2003)
2. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
3. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001), Software available at `http://www.csie.ntu.edu.tw/cjlin/libsvm`

4. Chiu, T., Fang, D., Chen, J., Wang, Y., Jeris, C.: A robust and scalable clustering algorithm for mixed type attributes in large database environment. In: Proc. of the Seventh ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, ACM (2001)
5. Cudney, E.A., Paryani, K., Ragsdell, K.M.: Identifying Useful Variables for Vehicle Braking Using the Adjoint Matrix Approach to the Mahalanobis-Taguchi System. Journal of Industrial and Systems Engineering 1(4), 281–292 (2008)
6. DeJong, K.A., Spears, W.M.: An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496, pp. 38–47. Springer, Heidelberg (1991)
7. Depren, O., Topallar, M., Anarim, E., Ciliz, M.K.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Systems with Applications 29, 713–722 (2005)
8. Giacinto, G., Perdisci, R., Rio, M.D., Roli, F.: Intrusion detection in computer networks by a modular ensemble of one-class classifiers. Information Fusion 9, 69–82 (2008)
9. Goldberg, D.E.: Genetic algorithms in search optimization and machine learning. Addison-Wesley, Reading (1989)
10. Grefenstette, J.J.: Optimization of Control Parameters for Genetic Algorithms. IEEE Trans. Systems, Man, and Cybernetics SMC-16(1), 122–128 (1986)
11. Grichnik, A.J., Seskin, M.: Mahalanobis Distance Genetic Algorithm (MDGA) Method and System. US Patent 2006/0230018 A1 (October 12, 2006)
12. Hedayat, A.S., Sloane, N.J.A., Stufken, J.: Orthogonal Arrays. Theory and Applications. Springer, New York (1999)
13. Huang, C.-L., Wang, C.-J.: A GA-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications 31, 231–240 (2006)
14. John, G., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345 (1995)
15. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice-Hall (1992)
16. Kaufman, L., Rousseeuw, P.J.: Clustering by means of Medoids. Statistical Data Analysis Based on the L1-Norm and Related Methods. In: Dodge, Y. (ed.), pp. 405–416. North-Holland (1987)
17. Krzanowski, W.J., Hand, D.J.: ROC Curves for Continuous Data. Chapman & Hall/CRC, London (2009)
18. Lee, C.H., Shin, S.W., Chung, J.W.: Network Intrusion Detection Through Genetic Feature Selection. In: Proc. of the Seventh ACIS International Conference on Software Engineering, Artificial Interlligence, Networking, and Parallel/Distributed Computing, SNPD 2006 (2006)
19. Lee, Y.C., Teng, H.L.: Predicting the financial crisis by Mahalanobis-Taguchi system - Examples of Taiwan's electronic sector. Expert Systems with Applications 36, 7469–7478 (2009)
20. Liparas, D., Angelis, L., Feldt, R.: Applying the Mahalanobis-Taguchi strategy for software defect diagnosis. Automated Software Engineering 19(2), 141–165 (2012)
21. Liparas, D., Laskaris, N., Angelis, L.: Incorporating resting state dynamics in the analysis of encephalographic responses by means of the Mahalanobis-Taguchi strategy. Expert Systems with Applications 40(7), 2621–2630 (2013)
22. Min, S.-H., Lee, J., Han, I.: Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Systems with Applications 31(3), 652–660 (2006)

23. Mukkamala, S., Sung, A.H., Abraham, A.: Intrusion detection using an ensemble of intelligent paradigms. Network and Computer Applications 28, 167–182 (2005)
24. Pal, A., Maiti, J.: Development of a hybrid methodology for dimensionality reduction in Mahalanobis-Taguchi system using Mahalanobis distance and binary particle swarm optimization. Expert Systems with Applications 37, 1286–1293 (2010)
25. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
26. R Development Core Team: R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria (2005), `http://www.R-project.org` ISBN 3-900051-07-0
27. Ruck, D., Rogers, S., Kabrisky, M., Oxley, M., Suter, B.: The multilayer perceptron as an approximation to a Bayes optimal discriminant function. IEEE Transactions on Neural Networks 1(4), 296–298 (1990)
28. Ryan, J., Lin, M.-J., Miikkulainen, R.: Intrusion Detection with Neural Networks. In: Advances in Neural Information Processing Systems, vol. 10. MIT Press, Cambridge (1998)
29. Scarfone, K., Mell, P.: Guide to Intrusion Detection and Prevention Systems (IDPS), Computer Security Resource Center (National Institute of Standards and Technology) (800-94) (2007)
30. Scott, S.L.: A Bayesian paradigm for designing intrusion detection systems. Computational Statistics and Data Analysis 45, 69–83 (2004)
31. Shon, T., Moon, J.: A hybrid machine learning approach to network anomaly detection. Information Sciences 177, 3799–3821 (2007)
32. Stein, G., Chen, B., Wu, A.S., Hua, K.A.: Decision tree classifier for network intrusion detection with GA-based feature selection. Paper presented at the Proc. of the 43rd Annual Southeast Regional Conference, Kennesaw, Georgia (2005)
33. Su, C.T., Hsiao, Y.H.: An evaluation of the robustness of MTS for imbalanced data. IEEE Trans. Knowl. Data Eng. 19(10), 1321–1332 (2007)
34. Taguchi, G., Rajesh, J.: New trends in multivariate diagnosis. Sankhya 62, 233–248 (2000)
35. Taguchi, G., Jugulum, R.: The Mahalanobis-Taguchi strategy - A pattern technology system, p. 234. John Wiley and Sons (2002)
36. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A detailed analysis of the KDDCup 1999 dataset. In: Proc. of 2009 IEEE International Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009), USA (2009)
37. Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., Lin, W.-Y.: Intrusion detection by machine learning: a review. Expert Systems with Applications 36(10), 11994–12000 (2009)
38. Whitley, D.: A genetic algorithm tutorial. Statistics and Computing 4(2), 65–85 (1994), doi:10.1007/BF00175354
39. Woodall, W.H., Koudelik, R., Tsui, K.L., Kim, S.B., Stoumbos, Z.G., Carvounis, C.P.: A review and analysis of the Mahalanobis-Taguchi system. Technometrics 45(1), 1–30 (2003)
40. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. IEEE Intelligent Systems 13(2), 44–49 (1998)
41. Zhang, Z., Shen, H.: Application of online-tradining SVMs for real-time intrusion detection with different considerations. Computer Communications 28, 1428–1442 (2005)