Nonlinear Feature Transformation and Genetic Feature Selection: Improving System Security and Decreasing Computational Cost

Saeid Asgari Taghanaki, Mohammad Reza Ansari, Behzad Zamani Dehkordi, and Sayed Ali Mousavi

Intrusion detection systems (IDSs) have an important effect on system defense and security. Recently, most IDS methods have used transformed features, selected features, or original features. Both feature transformation and feature selection have their advantages. Neighborhood component analysis feature transformation and genetic feature selection (NCAGAFS) is proposed in this research. NCAGAFS is based on soft computing and data mining and uses the advantages of both transformation and selection. This method transforms features via neighborhood component analysis and chooses the best features with a classifier based on a genetic feature selection method. This novel approach is verified using the KDD Cup99 dataset, demonstrating higher performances than other well-known methods under various classifiers have demonstrated.

Keywords: Intrusion detection system, feature transformation, feature selection, genetic algorithm.

Manuscript received Apr. 10, 2012; revised Aug. 24, 2012; accepted Oct. 1, 2012.

Saeid Asgari Taghanaki (corresponding author, phone: +98 9358643778, s.asgari@iaumajlesi.ac.ir) is with the Department of Computer and Electrical Engineering, Majlesi Branch, Islamic Azad University, Isfahan, Iran.

Mohamad Reza Ansari (rezaansari@yahoo.com) is with the Department of Computer and Electrical Engineering, Semirom Branch, Islamic Azad University, Isfahan, Iran.

Behzad Zamani Dehkordi (bzamani@iust.ac.ir) is with the Department of Computer and Electrical Engineering, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran.

Sayed Ali Mousavi (sa.mousavi@pmc.iaun.ac.ir) is with the Department of Mechanical Engineering, Najafabad Branch, Islamic Azad University, Isfahan, Iran.

http://dx.doi.org/10.4218/etrij.12.1812.0032

I. Introduction

With the development of communication and the interchanging of information, the Internet has also provided a superior opportunity to disorder and impair data that was previously considered safe. While we are benefiting from the convenience that the novel technology has brought us, computer systems and networks are facing an increasing number of security threats, some of which are initiated externally and others internally [1]-[4]. An intrusion detection system (IDS) is an efficient tool for system security and policy, of which there are two types: misuse detection and anomaly detection. The misuse detection IDS works with familiar patterns, and the anomaly detection IDS recognizes abnormalities against the normal network functions. In a hybrid detection system, these detection systems are united. A number of approaches based on soft computing have been proposed for detecting network intrusions. Soft computing consists of many concepts, including neural networks, artificial intelligence, fuzzy logic, genetic algorithms (GAs), information, and probabilistic reasoning [5]. Soft computing techniques are frequently used in combination with rule-based expert systems in the form of "if-then" rules. Although several approaches based on soft computing have been suggested recently, the probability of using one of the techniques for intrusion detection remains low [6]. Intrusion detection could be regarded as a data analysis procedure. Network behaviors can be classified as either normal or abnormal. Because of the high volume of real network traffic, considering the quantity of records and features, it is very difficult to process all the traffic

information before making decisions. Numerous learning methods have been designed for feature selection, feature extraction, or construction, but empirical studies and theoretical analyses have proven that many of them fail when the number of samples and redundant features is high [7]. Feature selection and feature transformation are commonly used techniques in data preprocessing. Feature transformation is a process that creates a new collection of features.

In this process, the high-dimensional data is transformed into a meaningful representation of reduced dimensionality [8]. Feature transformation is important in many areas since it mitigates dimensionality and other undesired properties of high-dimensional spaces [9]. Finally, feature transformation provides simplicity and accuracy regarding classification, visualization, and compression of high-dimensional data. Traditionally, feature transformation was done using linear techniques, such as principal component analysis (PCA) [10], factor analysis [11], linear discriminant analysis (LDA) (which is one of the most favored supervised linear dimensionality reduction techniques) [12], and classical scaling [13]. In detection approaches based on data mining methods, different features are used. These features are obtained by static and dynamic analysis. Performance of a pattern recognition system depends strongly on the employed feature selection method. Due to the increasing computational cost of a system with a rising number of features, it is important to implement a system that contains as few features as possible. Regarding many high-dimensional issues, the selection of the effective features and the removal of other features can greatly increase the precision of classification and reduce the intricacy of data processing at different steps. In [14], the authors carried out a simple GA that develops weights for the features of the data collection. After that, the k nearest neighbor algorithm (k-NN) classifier was used to evaluate the fitness function. Xia and others [15] introduced an approach that uses information theory and GAs to recognize abnormal network actions. Chittur [16] planned a genetic algorithm that raises a high detection rate of malicious behavior and classifies a low false positive rate of normal behavior as attack. Lu and Traore [17] planned an approach that uses genetic programming to directly obtain a collection of classification rules from historical network data. Li [18] suggested a GA-based method to recognize anomalous network behaviors. In this approach, quantitative and categorical features need to drive the classification rules by GA. Jian and others [19] developed a rule-based classifier to hinder the abnormal traffic; their classifier shows an acceptable recognition rate if applied to 10% of the training KDD Cup99 dataset, but there is no evidence that this classifier achieves accurate results when applied to a larger dataset. In this paper, the nonlinear and nonparametric feature transformation method introduced in [20], that is, neighborhood component analysis (NCA), is used. NCA chooses a projection that promotes the performance of an NN classifier in the projected space. We need to choose the most important data that can be used to desirably detect network attacks. The feature selection is different from feature transformation. In feature selection, the new features will not be produced; rather, only a subset of primary features is selected and the feature space is reduced. In the feature transformation process, new features will generate and previous features' values will change [7].

In this paper, we suggest a new approach based on soft computing and data mining techniques, called neighborhood component analysis feature transformation and genetic feature selection (NCAGAFS). Our method uses the advantages of feature transformation and feature selection, which transforms features via NCA and chooses the effective features with a classifier-based genetic feature selection method. The principal issues in developing feature selection techniques are selecting a small feature set to reduce the cost and running time of a given system and reaching an acceptably high detection rate. Several techniques have been developed for choosing a desirable subset of features from a larger set of probable features. The proposed method has been classified as a misuse detection type of IDS because patterns have high importance in NCAGAFS. The proposed method is evaluated using the KDD Cup99 dataset, which includes 41 different features and one label as a class. Empirical results show that the NCAGAFS method has a higher performance than traditional methods.

The reminder of the paper is structured as follows. Section II discusses NCA, GA-based feature selection, and the proposed method, which is a hybrid NCA and classifier-based GA. Section III presents the experiment results obtained using the KDD Cup99 dataset. Finally, conclusions are explained in section IV.

II. Framework of NCA and GAFS

In this section, we elaborate on our novel approach, NCAGAFS. We discuss the two main modules, that is, the NCA module as a feature transformation and the GA module as a feature selection.

1. NCA as Feature Transformation Method

NCA was proposed by Goldberger and others [20]; we explain the details of the method here for clarification. NCA adopts a linear projection of vectors into a space that improves a criterion depending on the leave-one-out accuracy of an NN classifier on a training set. Specifically, NCA takes as input a training set containing vectors $\{x_1, x_2, \ldots, x_N\}$, where $x_i \in R^m$,

and an associated set of labels $\{y_1, y_2, \ldots, y_N\}$, where $y_i \in \mathcal{L}$. The method then adopts a projection matrix A of size $p \times m$ that projects the training vectors x_i into a p dimensional representation, $z_i' = Ax_i$, where an NN classifier is operative at discriminating amongst the classes. The NN classifier uses this projection matrix A to define a Mahalanobis distance metric in the projected space [21].

$$d(x_i, x_j) = (Ax_i - Ax_j)^T (Ax_i - Ax_j). \quad (1)$$

The method makes use of "soft-neighbor" assignments instead of directly using the k-NNs to define a differentiable optimization criterion. Specifically, each point of j in the training set has a probability of p_{ij} to assign its label to a point i that decays as the distance between points i and j increases.

$$p_{ij} = \frac{\exp(-\| Ax_i - Ax_j \|^2)}{\sum_{k \neq i} \exp(-\| Ax_i - Ax_k \|^2)}, p_{ii} = 0. \quad (2)$$

The method attempts to enhance the expected number of points correctly classified in a leave-one-out setting over the training set. First, the value of p_i is defined to characterize the probability of a point i being assigned the correct class label.

$$p_i = \sum_{j=C_i} p_{ij}, \; C_i = \{j \mid y_j = y_i\}. \quad (3)$$

The final improvement criterion $f(A)$ can then be defined simply as the sum of the probabilities of classifying each point correctly.

$$f(A) = \sum_i p_i. \quad (4)$$

This criterion gives rise to a gradient rule that can be used to improve the matrix A (note that x_{ij} is shorthand for $x_i - x_j$).

$$\frac{\partial f}{\partial A} = 2A \sum_i (p_i \sum_k (p_{ik} x_{ik} x_{ik}^T - \sum_{j \in C_i} p_{ij} x_{ij} x_{ij}^T). \quad (5)$$

By using a number of gradient methods, such as conjugate gradient ascent or stochastic gradient ascent, this function can be improved. Note that the function $f(A)$ is not convex, so care needs to be taken when initializing the matrix A to avoid suboptimal solutions.

The above gradient method can have high computational costs. Computing the soft-neighbor probabilities alone requires $O(N^2p)$ cost. However, as many of these probabilities will be very close to zero, we can truncate the gradient calculation. Moreover, we can decrease the amount of calculation required by rearranging terms of the gradient as follows [21].

$$\frac{\partial f}{\partial A} = 2A \sum_i (p_i \sum_k (p_{ik} A(x_{ik}) x_{ik}^T - \sum_{j \in C_i} p_{ij} (Ax_{ij}) x_{ij}^T) \quad (6)$$

In our experiments, we optimize $f(A)$ using conjugate gradient ascent.

2. GA as Feature Selection Method

It is impossible to search all subsets to find an optimal subset since it requires a great number of computational attempts. Different heuristic search strategies have been used, including hill-climbing, branch and bound algorithms, forward selection, backward elimination, and such stochastic algorithms as simulated annealing and GA [22].

The first step of the GA is creating a chromosome. Every chromosome is a collection of genes. In the feature selection problem, each gene shows a feature and a chromosome is a representation of a set of features. To determine whether a specific feature is present or not in the chromosome, one and zero are used. One in a gene position indicates that a particular feature is present, and zero indicates that it is absent. The other question is about quantity and the nature of the feature in a chromosome gained through information [23]. The first population is created randomly by values present in the chromosome. After that, the individuals are evaluated by a fitness function, which is a classification error in our experiment, the false positive and false negative values being measured for each chromosome. The fitness function must examine the efficacy of each individual in a population, so it considers each individual an input and provides a numerical estimation that should show the benefits of the features. The chromosome with the best fitness value is regarded as the elite one.

In the next stage, crossover and mutation are to be applied on the chromosomes that have the highest fitness value. A mutation operator tries to preserve diversity in the population since mutation chooses one position randomly from the chromosome [23].

3. Hybrid NCA and GA-Based Feature Selection

In this subsection, we present our proposed method, which integrates NCA as a feature transformation and the GA as the process of feature selection. Then, we use selected transformed features in the classifier to increase the accuracy rate. The main goal of our work is to use information gained by the significance of features and then select a set of significant features using the proposed GA. As we mentioned above, NCA has desirable performance in transformation for a pattern classification system, and, after applying the NCA, the dimension is reduced, data is correlated, and the error rate of classification is decreased. The performance of the method is useful for various datasets, both for dimensionality reduction and metric learning. In general, there are two classes of regularization assumption that are common in linear methods for classification. The first is a robust parametric supposition about the structure of the class distributions (typically enforcing connected or even convex structure); the second is a

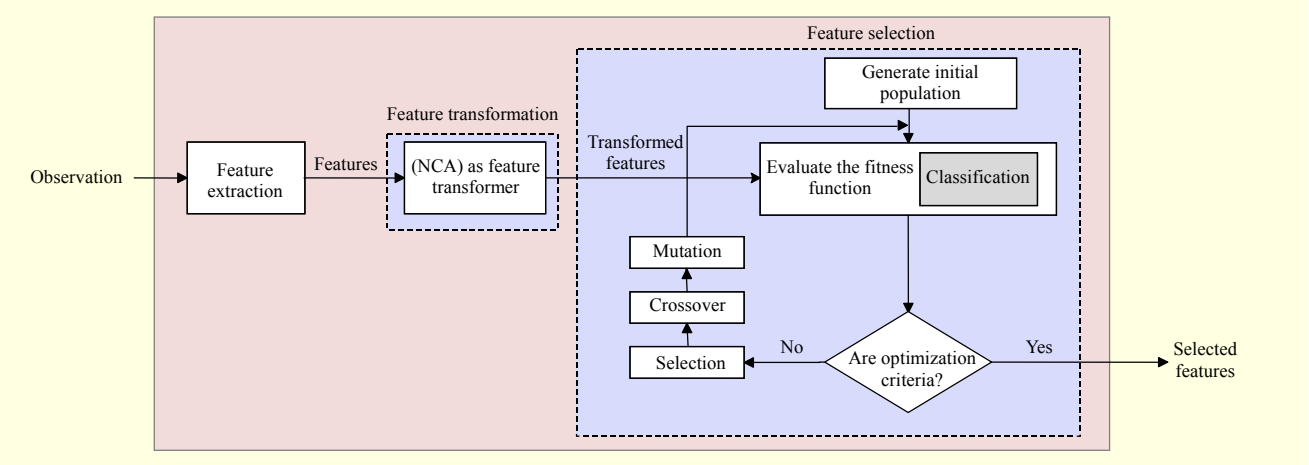

Fig. 1. Detailed block diagram of proposed method.

supposition on the decision boundary (typically enforcing a hyper plane). The NCA method makes neither of these suppositions, relying instead on the robust regularization imposed by restricting ourselves to a linear transformation of the original inputs. In the dataset with *n* features, we can set the value for the output dimension from *n* to 1 feature. However, if we set the output dimension value near 1, the important features could be lost. In this paper, we combine the NCA and GA to keep the best and significant features and not lose the important features. GA finds global maximum optimum and so we can expect GA to choose the best subset of features. GA is an iterative process, and the cycle of production and evaluation is continuous until it reaches the best fitness.

We use NCA as a preprocessor for GA. A detailed block diagram of our method is shown in Fig. 1.

III. Results and Discussions

1. Data Preparation

To experiment and to work with the system classifier, the KDD Cup99 [24] dataset is suitable to use. In KDD Cup99, each TCP/IP connection is particularly explained by 41 different contiguous features, such as duration, protocol type flag, and so on. Attacks fall into four main categories:

- DOS: denial-of-service (for example, syn flood);
- R2L: unauthorized access from a remote machine (for example, to guess passwords);
- U2R: unauthorized access to local super user (root or admin) prominences (for example, several "buffer_overflow" attacks);
- Probe: surveillance and other probing methods (for example, scanning of ports).

In our experiments, we randomly select the records from each category. Table 1 shows detailed information about the number of samples for the normal class and the attack class, the size of training, and the testing dataset.

Table 1. Number and distribution of training and testing dataset used in experiments.

Connection type	Count	Probability	Training dataset in 66% split	Testing dataset in 66% split
DOS	5,000	0.449	3,300	1,700
Normal	2,432	0.243	1,605	827
Probe	1,520	0.151	1,003	517
R2L	1,000	0.099	660	340
U2R	49	0.004	32	17

2. Calculation Criteria

The following calculations are usually suggested to calculate the detection precision of IDS [25]: true positives rate (TP), true negatives rate (TN), false positives rate (FP), and false negatives rate (FN). A true positive shows that the IDS detects precisely a specific attack having happened. A true negative shows that the IDS has not made a mistake in recognizing a normal condition. A false positive results from losing recognition conditions, a limitation of detection methods in the IDS, or unusual conditions caused by particular environmental parameters. It represents accuracy and precision of the IDS. A false negative shows the inability of IDS to detect the intrusion after the occurrence of a particular attack. This might happen due to a lack of information about an intrusion type or because the recognition information regarding such an intrusion event is excluded from the IDS. This shows the perfection of the detection system. However, if the quantity of samples for the

U2R, Probe, and R2L attacks in the training set and test set is very low, these numbers as a standard operation measure are not enough [26]. Hence, if these numbers are regarded as a measure for testing the performance of the system, it could be inexact. Therefore, we use the precision, recall, and F-measure, which are not related to the size of the training and the testing instance. They are mentioned as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}, \tag{9}$$

$$\text{F-measure} = \frac{(1+\beta^2) * \text{Recall} * \text{Precision}}{\beta^2 * \text{Recall} + \text{Precision}}, \tag{10}$$

where β corresponds to the relative importance of precision versus recall and is usually set to 1. On the other hand, we calculate the percent of correction rate for each used classifier, such as BPNN, RBF net, Bayesian net, and so on.

Furthermore, the criteria of the receiver operating characteristic (ROC) area and the precision-recall (PRC) area are calculated. PRC has superior performance over area under curve (AUC) when the class distribution is significantly skewed [27]. In this paper, because we have a large number of selections, transformation methods, and classifiers, if we want to draw curves, the diagrams will be intricate. For this reason, we calculate the weighted average values of the ROC and PRC areas as results.

The total time required to build the model is also a crucial parameter in comparing the classification algorithms. To compute the difficulty of solving a computational problem, we measure how much time the classifiers require to solve the problem. We try to reduce the time by two steps, one of them described in the body of the NCA algorithm in subsection II.1, to reduce the computational costs, and the second is in the body of the GA (with feature selection and reduction of data).

3. Experiment Results

Each connection in the KDD CUP99 dataset has 41 features and a label reflecting the name of the attack. The label is used only for the training dataset (Table 1). The selection of this dataset is owing to its generality and content richness, and it enables us to evaluate our experiment results with accessible research in the area of IDSs. In the first step of preprocessing, we discard three symbolic values (for example, udp, private, and SF) out of the 41 features of the dataset [28], [29]. The transformation of these symbolic features to numeric values might vary throughout the database, but, in this research, we are left with a random selection of only 10,001 samples from all five million connection records (Table 1) after the test decides these symbolic features do not have a positive effect on our results. The NCA works based on the Mahalanobis distance, and the *k*-NN classifier works based on Euclidean distances, so the algorithm does not have a powerful calculation of the neighbors if the features are very diffuse. Then, we feed the normal dataset with 38 features to three types of classifiers and compute the values of precision, recall, F-measure, percent of correctly classified, PRC area, and ROC area for each classifier. In the next step, we apply the PCA, LDA, PCA+LDA, NCA, GAFS, and NCA+PCA in normal data and feed this new data to classifiers and compute the evaluation criteria.

Table 2. Number of selected features per each classifier with NCAGAFS and GAFS.

Classifier	Number of selected features with NCAGAFS		Number of selected features with GAFS	
Bayes network	11	1, 5, 14, 15, 19, 20, 25, 27, 30, 34, 37	9	6, 12, 20, 22, 24, 33, 39, 40, 41
BPNN	22	1, 5, 6, 8, 9, 11, 12, 13, 15, 17, 20, 23, 26, 27, 29, 32, 35, 36, 38, 39, 40, 41	9	6, 12, 20, 22, 24, 33, 39, 40, 41
RBF network	9	9, 10, 19, 20, 25, 27, 36, 37, 38	13	5, 8, 9, 13, 16, 22, 24, 27, 30, 33, 39, 40, 41
Naïve Bayes	3	8, 18, 20	15	3, 10, 11, 13, 14, 16, 18, 19, 20, 21, 22, 24, 25, 28, 37
Decision table	11	9, 15, 19, 21, 23, 25, 26, 30, 31, 33, 36	9	6, 12, 20, 22, 24, 33, 39, 40, 41
IBK (*k*-NN)	20	5, 6, 7, 9, 10, 11, 13, 16, 17, 18, 19, 21, 22, 25, 28, 30, 36, 37, 39, 41	9	6, 12, 20, 22, 24, 33, 39, 40, 41

On the other hand, we apply the NCA algorithm to all datasets with 38 features to transform the features. Therefore, the input of the NCA algorithm is the dataset with 38 features with diffuse and discrete instances and the output is the data with more correlation and more proximity without any feature reduction. After that, we adjust the GA parameters for features subset selection. In our experiment, we set these parameters to: population size: 80, number of generations: 100, probability of crossover: 0.6, probability of mutation: 0.033, type of mutation: uniform, and type of selection: rank-based. Choosing an appropriate evaluation function is an essential step for successful application of GAs to any problem domain. The fitness function in our method is the classification error. It means that for each model of classifiers, the fitness function is adjustable, and, in the feature selection step, the GA selects the

Table 3. Abbreviation of implemented methods in our experiments.

Method name	Description	No. of features
Normal	Without any feature transformation and feature reduction	38
PCA1	PCA transformation without any feature reduction	38
PCA2	Feature reduction of PCA	18
LDA	LDA transformation without any feature reduction	38
PCA+LDA	Hybrid PCA and LDA transformation without any feature reduction	38
NCA	NCA transformation without any feature reduction	38
NCA+PCA	Hybrid NCA and PCA (NCA as a transformation method and PCA as a reduction method)	5
GAFS	Genetic feature selection without any transformation	Depend on classifier (Table 2)
NCAGAFS	Proposed method	Depend on classifier (Table 2)

features based on the type of classifiers. In our method, the best fitness value is the lowest error for each feature. Finally, after executing the GA and evaluating the fitness function for each model of classifiers and computation of the best fitness value, important features based on the model of classifier are selected. Table 2 shows the number of selected features for six classifiers in NCAGAFS and GAFS.

Experiment results are evaluated by three types of classifiers, that is, model-based classifiers, distance-based classifiers, and rule-based classifiers. Table 3 shows abbreviations of the implemented methods.

As shown by the following tables, we can clearly see the difference of every evaluation criterion under different transformation and selection methods that are shown in Table 3. Different classifiers are used to evaluate our method and other feature transformation and selection methods. The obtained results prove that our proposed method has higher performance than traditional methods. The results correlating to each case are reported in the following subsections.

A. Bayesian Network Results

A Bayesian network is a probable graphical model that shows a collection of random variables and their conditional dependencies through a directed acyclic graph. Bayes network learning uses several search algorithms and quality evaluations. In our experiments, a simple estimator is used to estimate the conditional probability tables of a Bayes network once the structure has been learned. Additionally, this Bayes network learning algorithm uses a hill-climbing algorithm restricted by an order on the variables. Table 4 shows values of different evaluation criteria and correct classification rate for various feature transformation and feature selection methods. As shown in Table 4, our method performs better than other methods. The highest evaluation criteria values and correct classification rate in each row are marked by the bold and underline formats, and the second ranked values are marked by the bold format only. The proposed method improves the correct classification rate in relation to normal data by about 2.15%.

B. BPNN Results

The multilayer perceptron is the standard network to use for supervised and nonparametric learning. In our experiment, we put one hidden layer composed of n neurons where

$$n = (\text{number of all features} + \text{number of class})/2.$$

We decide on n because experiments show that adding more neurons inside the hidden layer does not increase the accuracy as expected for this particular problem. The sigmoid function, a gradient descent with momentum, and an adaptive learning rate back-propagation training function, namely, the Levenberg Marquadt method, are used. Based on the results shown in Table 5, the proposed method has a higher recognition rate than other methods. Also, NCAGAFS increases the precision criterion about 0.4. Our method uses NCA as a transformation feature and selects transformed features by the GA, which finds the global optimum, so we expect NCAGAFS to perform better than other methods. The last row of table 5 shows that the proposed method is about three times faster than the normal method. GAFS has desirable modeling time but lower accuracy than NCAGAFS.

C. RBF Network Results

The RBF network is a classifier that implements a normalized Gaussian radial basis function network. This classifier uses the k-means clustering algorithm to create the basic functions and learns either a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that. Symmetric multivariate Gaussians are fit to the data from each cluster. It uses the given number of clusters per class if the class is nominal. It normalizes all numeric attributes to zero mean and unit variance. In our experiments, the RBF network options are:

- number of clusters (basis functions) = 2;
- ridge parameter for the logistic regression or linear regression = 1.0E-8;
- minimum standard deviation for the clusters = 0.1;

Table 4. Bayes net classifier results.

Weighted avg. of 5 classes	Normal	LDA	PCA1	LDA+PCA	PCA2	NCA	NCA+PCA	GAFS	NCAGAFS
Precision	0.878	0.84	**0.886**	0.822	0.871	0.869	0.849	0.827	**0.891**
Recall	0.83	0.798	**0.841**	0.79	0.819	0.816	0.816	0.839	**0.851**
F-measure	0.85	0.816	**0.856**	0.797	0.841	0.838	0.829	0.832	**0.867**
ROC area	**0.983**	0.97	0.979	0.968	0.978	0.975	0.976	0.978	**0.982**
PRC area	**0.983**	0.849	0.886	0.845	0.871	0.881	0.872	0.881	**0.912**
Correctly classified	83.00%	79.76%	**84.06%**	78.97%	81.91%	81.59%	81.56%	83.94%	**85.15%**
Modeling time (s)	2.48	4.29	4.78	0.82	2.82	5.28	1.01	0.39	1.4

Table 5. BPNN classifier results.

Weighted avg. of 5 classes	Normal	LDA	PCA1	LDA+PCA	PCA2	NCA	NCA+PCA	GAFS	NCAGAFS
Precision	0.856	0.82	0.866	0.759	0.855	**0.909**	0.901	0.787	**0.924**
Recall	0.881	0.857	0.882	0.764	0.878	**0.907**	0.876	0.861	**0.908**
F-measure	0.844	0.815	0.843	0.731	0.838	**0.892**	0.845	0.818	**0.89**
ROC area	0.983	0.976	0.983	0.947	0.981	**0.989**	0.978	0.977	**0.988**
PRC area	0.89	0.857	0.896	0.792	0.89	**0.937**	0.894	0.869	**0.932**
Correctly classified	88.15%	85.74%	88.24%	76.41%	87.79%	**90.71%**	87.56%	86.15%	**90.79%**
Modeling time (s)	1037.43	936.66	888.79	104.64	313.39	888.96	78.59	50.64	382.62

Table 6. RBF network classifier results.

Weighted avg. of 5 classes	Normal	LDA	PCA1	LDA+PCA	PCA2	NCA	NCA+PCA	GAFS	NCAGAFS
Precision	**0.797**	0.758	0.755	0.583	0.775	0.717	0.769	**0.789**	0.765
Recall	0.824	0.825	**0.827**	0.565	0.823	0.808	0.809	0.803	**0.826**
F-measure	**0.796**	0.786	0.787	0.519	0.784	0.759	0.76	0.777	**0.79**
ROC area	0.946	0.965	0.96	0.765	**0.967**	0.949	0.963	0.951	**0.969**
PRC area	0.792	0.83	0.82	0.55	0.842	0.831	**0.853**	0.803	**0.879**
Correctly classified	82.44%	82.47%	**82.73**	56.47%	82.26%	80.79%	80.85%	80.27%	**82.62%**
Modeling time (s)	79.01	42.06	25.55	77.37	25.82	68.38	16.23	18.72	58.59

• random seed used by *k*-means when generating clusters = 1.

As the values show in Table 6, NCAGAFS increases correct recognition rate. NCAGAFS produces results similar to those of the PCA1 method, but our method has just nine features as opposed to the 38 features used in the PCA1 method. Therefore, we decrease the computational cost by selecting fewer features and only the significant features.

In some cases, our results are comparable to results of other methods; however, in this paper, we try to achieve multiobjective optimization, which simultaneously increases the accuracy and decreases the computational costs. Therefore, we potentially have more time to build the model than we do if using such methods as RBF.

D. Naïve Bayes Results

The Naïve Bayes classifier is one of the ordinary probable classifiers based on Bayes' theorem that strong (naïve) independence suppositions assume that the presence (or absence) of a specific feature of a class is independent of the

Table 7. Naïve-Bayes classifier results.

Weighted avg. of 5 classes	Normal	LDA	PCA1	LDA+PCA	PCA2	NCA	NCA+PCA	GAFS	NCAGAFS
Precision	0.769	0.742	0.755	0.757	0.773	0.77	**0.813**	**0.816**	0.812
Recall	0.564	0.658	0.58	0.616	0.73	0.666	0.691	**0.715**	**0.754**
F-measure	0.615	0.649	0.611	0.622	0.714	0.683	0.695	**0.725**	**0.777**
ROC area	0.916	0.916	0.881	0.868	**0.927**	0.904	0.907	0.921	**0.941**
PRC area	0.762	0.765	0.748	0.733	**0.788**	0.77	0.783	0.758	**0.824**
Correctly classified	56.44%	65.82%	57.97%	61.65%	**72.97%**	66.59%	69.15%	71.47%	**75.41%**
Modeling time (s)	1.47	1.14	1.94	0.23	0.62	1.23	0.26	0.18	0.1

Table 8. Decision table classifier results.

Weighted avg. of 5 classes	Normal	LDA	PCA1	LDA+PCA	PCA2	NCA	NCA+PCA	GAFS	NCAGAFS
Precision	0.858	0.822	0.847	0.831	0.834	**0.867**	0.84	0.84	**0.865**
Recall	**0.877**	0.85	0.876	0.851	0.851	0.876	0.856	0.867	**0.878**
F-measure	0.849	0.818	0.84	0.834	0.838	**0.867**	0.844	0.826	**0.868**
ROC area	**0.981**	0.976	0.98	0.976	0.976	0.978	0.976	0.97	**0.982**
PRC area	0.877	0.859	0.877	0.86	0.863	0.887	**0.888**	0.846	**0.905**
Correctly classified	**87.71%**	85.03%	87.59%	85.09%	85.06%	87.65%	85.59%	86.74%	**87.79%**
Modeling time (s)	36.54	29.98	32.06	3.28	12.55	31.7	1.76	1.42	7.51

presence (or absence) of any other feature, given the class variable. Even if these features relate to every other or depend upon the existence of the other features, this classifier considers all of these features to independently relate to the probability. Table 7 shows the results for the Naïve Bayes method. As shown, NCAGAFS has superior results. We evaluate our method as preprocessing of model-based, decision-based, and distance-based classifiers. The results shown in III.3.A, III.3.B, III.3.C, and III.3.D verify that our method increases the correct recognition rate.

E. Decision Table Results

A decision table as a nonmetric classification method presents a group of rules that can be supposed for building and using a simple decision table majority classifier. Decisions are related to predicates or variables whose possible values are listed among the condition alternatives. Each action is a function to be executed. Many decision tables use a "don't care" symbol to simplify their structure. In digital logic, a "don't care" term is an input-string (a sequence of bits) to a function that the creator does not care about, typically since that input would never happen or because variation in that input would not result in any changes to the output. RMSE is used to evaluate the performance of attribute combinations in our experiments, and a genetic search is used as a search method to find good attribute combinations for the decision table. Results of the decision table classifier are shown in Table 8. We obtain the same results as before. We have negligible improvement, but the computational complexity is reduced. The proposed method decreases the simulation time threefold in comparison to a normal method. This decrement is due to fewer features as well as the correlation between data. We can see that GAFS takes 1.42 seconds but has poor performances in other evolution criteria.

F. Results of k-NN

The *k*-NN is a robust nonparametric method for classifying objects based on closest training samples in the feature space: an object is classified by a majority vote of its neighbors, with the object related to the class with the most current *k*-NNs (*k* is a positive integer number, usually small). If we set k=1, then the object is simply related to the NN classes. Results of the *k*-NN classifier by k=3 are shown in Table 9. In our experiment, we use the LinearNNSearch, which implements the Euclidean distance function as an NN search algorithm. Based on our results, NCAGAFS performs better than other methods. Our

Table 9. *k*-NN classifier results (*k*=3).

Weighted avg. of 5 classes	Normal	LDA	PCA1	LDA+PCA	PCA2	NCA	NCA+PCA	GAFS	NCAGAFS
Precision	0.864	0.874	0.864	0.866	0.866	**0.902**	0.894	0.838	**0.897**
Recall	0.874	0.883	0.875	0.876	0.877	**0.907**	0.9	0.853	**0.905**
F-measure	0.868	0.878	0.869	0.871	0.871	**0.904**	0.897	0.845	**0.9**
ROC area	**0.967**	0.964	0.964	0.963	0.965	0.952	0.948	**0.972**	**0.972**
PRC area	0.861	0.86	0.855	0.855	**0.862**	0.858	0.849	0.851	**0.894**
Correctly classified	87.44%	88.32%	87.53%	87.65%	87.73%	**90.71%**	90%	85.33%	**90.5%**
Modeling time (s)	0.02	0.06	0.02	0.02	0.02	0.02	0.02	0.01	0.02

method has an acceptable performance, which is used as preprocessing for distance-based classifiers. If the comparison of classifiers is done with respect to the time taken to build the model and the identification of correct instances, among other evaluation criteria, then it is concluded that NCAGAFS provides better results. Table 9 shows that the construction time in NCAGAFS is equal to that of normal methods, but the results regarding other criteria are different and this is due to the correlation of data and significant features selection.

G. Dunn Index and SD Index Measure

In this experiment, we use the Dunn index as performance criterion. The Dunn index is a validity index which identifies compact and well-separated classes defined by (11) for a specific number of classes [30], [31].

$$D_{nc} = \min_{i=1,\ldots,nc}\left\{\min_{j=i+1,\ldots,nc}\left(\frac{dist(c_i,c_j)}{\max_{k=1,\ldots,nc} diam(c_k)}\right)\right\}, \tag{11}$$

where nc is the number of classes, and $dist(c_i,c_j)$ is the dissimilarity function between classes c_i and c_j, defined by

$$dist(c_i,c_j) = \min_{x\in c_i, y\in c_j} dist(x,y), \tag{12}$$

and $diam(c)$ is the diameter of the class c, a measure of the dispersion of the class. The diameter of a class c can be defined as

$$diam(c) = \max_{x,y\in c} dist(x,y). \tag{13}$$

It is clear that if the data classes are compact and well separated, the distance between the classes will be too large, and the diameter of the classes will be too small. Thus, based on the Dunn index definition, we can conclude that large values in the index indicate the presence of compact and well-separated classes. The Dunn index measures compression of data classes. As such, higher values of the Dunn index indicate well separated and more compact classes. The SD validity index is defined based on the average scattering of classes and total separation between classes. Average scattering of classes is defined as

$$Scat(nc) = \frac{1}{nc}\sum_{i=1}^{nc}\|\sigma(v_i)\|/\|\sigma(X)\|, \tag{14}$$

where nc is the number of classes, v_i is the center of class I, X is the dataset, σ_i is the variance of the i-th class, and σ is the variance of the dataset [30], [32].

Total separation between classes is defined as

$$Dis(nc) = \frac{D_{\max}}{D_{\min}}\sum_{k=1}^{nc}\left(\sum_{z=1}^{nc}\|v_k - v_z\|\right)^{-1}, \tag{15}$$

where $D_{\max}$ and $D_{\min}$ are the maximum and minimum distance between class centers, respectively.

$$D_{\max} = \max(\|v_i - v_j\|)\forall i,j \in \{1,2,\ldots,nc\}, \tag{16}$$

$$D_{\min} = \min(\|v_i - v_j\|)\forall i,j \in \{1,2,\ldots,nc\}. \tag{17}$$

The SD validity index is then defined as

$$SD(nc) = \lambda.Scat(nc) + Dis(nc). \tag{18}$$

In this equation, the first term, *Scat(nc)*, defined in (14), represents the average scattering (or average compactness) of classes. The smaller value of *Scat(nc)* indicates greater compactness of the class. The second term, *Dis(nc)*, is a function of the location of a class's center. It measures the separation between the *nc* classes and increases with the number of classes. A small value of the SD index indicates the presence of compact and well-separated classes. Since the two terms of SD have different ranges, the weighting factor λ is used to balance their overall contribution. Here, we calculate the Dunn and SD indexes for each dataset. Table 10 shows the results of this experiment. The bold and underlined values are the best results achieved on each dataset, the bold values without underlining are the second best results, and the

Table 10. Dunn index and SD index for all methods.

Methods	Dunn index	SD index
Normal	1.44E-10	0.4121
LDA	2.81E-05	546.7815
PCA1	1.44E-10	0.4121
LDA+PCA	3.33E-04	284.6215
PCA2	6.08E-04	0.6107
NCA	0.0031	**0.2338**
NCA+PCA	0.0045	0.3135
GAFS for Bayes net, BPNN, decision table, IBK	8.00E-07	0.5403
GAFS for Naïve Bayes	6.59E-06	0.4836
GAFS for RBF	1.44E-10	0.4131
NCAGF for Bayes net	0.0045	0.3591
NCAGF for BPNN	0.003	0.2714
NCAGF for decision table	0.0039	0.2745
NCAGF for IBK	0.003	**0.2493**
NCAGF for Naïve Bayes	**0.0063**	0.5833
NCAGF for RBF	**0.0046**	0.3352

underlined values without boldface type are the third best values.

IV. Conclusion

In this paper, we introduced a new approach based on the nonlinear and nonparametric method NCA and a classifier-based GA feature selection method to create an efficient and powerful tool for selecting significant features. As we used fewer features than 41 to describe the data, its training time was noticeably decreased and the accuracy of its classification was improved. We compared our result with the results of seven feature selection and feature transformation methods and normal data used to preprocess three models of classifiers (six classifiers). The results showed that the classification method that uses selected features is more accurate than the classification method that uses all of the features. In all of the classifiers used, our method achieved better results than all other known methods achieved. In some cases, the results were notably close; however, our method uses fewer and more important features, thus reducing the modeling time and computational costs.

References

[1] T. Bhaskar et al., "A Hybrid Model for Network Security Systems: Integrating Intrusion Detection System with Survivability," *Int. J. Netw. Security*, vol. 7, no. 2, 2008, pp. 249-260.

[2] P. Kabiri and A. Ghorbani, "Research on Intrusion Detection and Response: A Survey," *Int. J. Netw. Security*, vol. 1, no. 2, 2005, pp. 84-102.

[3] I.-V. Onut A. Ghorbani, "A Feature Classification Scheme for Network Intrusion Detection," *Int. J. Netw. Security*, vol. 5, no. 1, 2007, pp. 1-15.

[4] Z. Zhang et al., "An Observation-Centric Analysis on the Modeling of Anomaly-based Intrusion Detection," *Int. J. Netw. Security*, vol. 4, no. 3, 2007, pp. 292-305.

[5] S.S. Kandeeban and R. Rajesh, "Integrated Intrusion Detection System Using Soft Computing," *Int. J. Netw. Security*, vol. 10, no. 2, Mar. 2010, pp. 87-92.

[6] N. Srinivasan and V. Vaidehi, "Performance Analysis of Soft Computing Based Anomaly Detectors," *Int. J. Netw. Security*, vol. 7, no. 3, 2008, pp. 436-447.

[7] P. Langley, *Elements of Machine Learning*, San Francisco, CA: Morgan Kaufmann, 1996.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Boston, MA: Academic Press, 1990.

[9] L.O. Jimenez and D. Landgrebe, "Supervised Classification in High-Dimensional Space: Geometrical, Statistical and Asymptotical Properties of Multivariate Data," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, no. 1, 1997, pp. 39-54.

[10] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Mag.*, vol. 2, no. 11, 1901, pp. 559-572.

[11] C. Spearman, "General Intelligence Objectively Determined and Measured," *American J. Psychology*, vol. 15, no. 2, 1904, pp. 206-221.

[12] S. Axler, *Linear Algebra Done Right*, New York, NY: Springer-Verlag, 1995.

[13] W.S. Torgerson, "Multidimensional Scaling I: Theory and Method," *Psychometrika*, vol. 17, 1952, pp. 401-419.

[14] M.J. Middlemiss and G. Dick, "Weighted Feature Extraction Using a Genetic Algorithm for Intrusion Detection," *Evolutionary Computation*, vol. 3, 2003, pp. 1699-1675.

[15] T. Xia et al., "An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm," *Proc. 24th IEEE Int. Performance Comput. Commun. Conf.*, Apr. 2005, pp. 11-17.

[16] A. Chittur; *Model Generation for an Intrusion Detection System Using Genetic Algorithms*, high school honors thesis, Ossining High School, Ossining, NY, USA, Nov. 2001.

[17] W. Lu and I. Traore, "Detecting New Forms of Network Intrusion Using Genetic Programming," *Computational Intell.*, Malden, MA: Blackwell Publishing, vol. 20, no. 3, 2004, pp. 475-494.

[18] W. Li, "Using Genetic Algorithm for Network Intrusion Detection," SANS Institute, USA, 2004.

[19] G. Jian, L. Da-xin, and C. Bin-ge, "An Induction Learning Approach for Building Intrusion Detection Models Using Genetic Algorithms," *Proc. 5th World Congress Intell. Control Autom.*, June 15-19, vol. 5, 2004, pp. 4339-4342.

[20] J. Goldberger et al., "Neighborhood Components Analysis," *Proc. Adv. Neural Inf. Process.*, Whistler, BC, Canada, 2005, pp. 571-577.

[21] N. Singh-Miller, M. Collins, and T.J. Hazen, "Dimensionality Reduction for Speech Recognition Using Neighborhood Components Analysis," *Proc. Interspeech*, 2007, pp. 1158-1161.

[22] A. Amine et al., "GA-SVM and Mutual Information Based Frequency Feature Selection for Face Recognition" GSCM-LRIT, Faculty of Sciences, Mohammed V University, B.P. 1014, Rabat, Morocco.

[23] S. Sethuramalingam and E.R. Naganathan, "Hybrid Feature Selection for Network Intrusion," *Int. J. Computer Sci. Eng.*, vol. 3, no. 5, 2011, pp. 1773-1780.

[24] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.htm

[25] S. Axelsson, "The Base-Rate Fallacy and the Difficulty of Intrusion Detection," *ACM Trans. Inf. Syst. Security*, vol. 3, no. 3, Aug. 2000, pp. 186-205.

[26] P. Dokas et al., "Data Mining for Network Intrusion Detection," *Proc. NSF Workshop Next Generation Data Mining (NGDM)*, 2002, pp. 21-30.

[27] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recog. Lett.*, vol. 27, no. 8, 2006, pp. 861-874.

[28] G. Liu and Z. Yi, "Intrusion Detection Using PCASOM Neural Networks," *Advances in Neural Networks – ISNN 2006*, J. Wang et al., Eds., Berlin/Heidelberg: Springer-Verlag, 2006, pp. 240-245.

[29] I. Ahmad et al., "Optimized Intrusion Detection Mechanism Using Soft Computing Techniques," *Telecommun. Syst.*, 2011, doi: 10.1007/s11235-011-9541-1.

[30] H. Abbasian et al., "Class Dependent LDA Optimization Using Genetic Algorithm for Robust MFCC Extraction," *Adv. Computer Sci. Eng.*, vol. 6, 2009, pp. 807-810.

[31] J.C. Dunn, "Well-Separated Clusters and Optimal Fuzzy Partitions," *J. Cybern.*, vol. 4, no. 1, 1974, pp. 95-104.

[32] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, "Quality Scheme Assessment in the Clustering Process," *Proc. 4th European Conf. Principles Data Mining Knowledge Discovery*, *LNCS*, vol. 1910, 2000, pp. 265-276.

Saeid Asgari Taghanki received his BSc in 2010 from Islamic Azad University, Shahrekord Branch (IAUSHK), Shahrekord, Iran, in computer software engineering and his MSc in mechatronics trends of human machine interface (HMI) from Islamic Azad University, Majlesi Branch (IAUMajlesi), Isfahan, Iran, with honors (first rank). He has several journal publications and conference contributions on pattern recognition and image processing. He is a lecturer at IAUSHK. His areas of research are pattern recognition, biomedical image processing, data mining, and soft computing.

Mohammad Reza Ansari received his BSc, MSc, and PhD in electrical engineering from the University of Semnan, Semnan, Iran, in 2003, 2007, and 2012, respectively. His main research interests include power systems operation, artificial intelligence, pattern recognition, systems planning, dynamic systems, and system control.

Behzad Zamani Dehkordi received his BSc in 2003 from Isfahan University of Technology (IUT), Isfahan, Iran, in hardware engineering, his MSc in 2006 from the Computer Engineering Department, Iran University of Science and Technology (IUST), Tehran, Iran, and his PhD in artificial intelligence from IUST in 2012. He is working as a research assistant in the ASPL and RCIT labs of the Computer Engineering Department, IUST. He has published many papers on speech recognition and related topics. His interests include feature transformation and kernel-based learning methods.

Sayed Ali Mousavi has been a member of ASME since 2003. He received his PhD in biomechanical engineering from the Eastern Mediterranean University of Cyprus (EMU), Famagusta, Northern Cyprus, in 2010, his MSc in mechanical engineering from EMU in 2009, and his BSc in mechanical engineering from Isfahan University of Technology (IUT), Isfahan, Iran, in 2002. He was a visiting scholar in the Endodontics Department, Loma Linda University, Loma Linda, CA, USA, from 2007 to 2008. During 2010 he served as an honored researcher in the Torabinejad Dental Research Center, Isfahan University of Medical Sciences, Isfahan, Iran. Dr Mousavi was the director of the Robotics Laboratory of Islamic Azad University, Najafabad Branch (IAUN), Najafabad, Isfahan, Iran, in 2011. He has published several articles and papers in books, journal transactions, and conference proceedings. He registered three patents under his name. His research interests are in the areas of biomechanical engineering, mechatronics, and intelligent systems.