INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2015, VOL. 61, NO. 3, PP. 289–300
Manuscript received February 4, 2015; revised September, 2015. DOI: 10.1515/eletel-2015-0038

Dimensionality Reduction for Probabilistic Neural Network in Medical Data Classification Problems

Maciej Kusy

***Abstract*—This article presents the study regarding the problem of dimensionality reduction in training data sets used for classification tasks performed by the probabilistic neural network (PNN). Two methods for this purpose are proposed. The first solution is based on the feature selection approach where a single decision tree and a random forest algorithm are adopted to select data features. The second solution relies on applying the feature extraction procedure which utilizes the principal component analysis algorithm. Depending on the form of the smoothing parameter, different types of PNN models are explored. The prediction ability of PNNs trained on original and reduced data sets is determined with the use of a 10-fold cross validation procedure.**

***Keywords*—probabilistic neural network, dimensionality reduction, feature selection, feature extraction, single decision tree, random forest, principal component analysis, prediction ability.**

I. Introduction

DATA sets are composed of input vectors where each one contains the same number of attributes, usually referred to as features. In classification tasks, the number of features depends on the considered problem. For example, it is enough to measure the age, body mass index and blood pressure for a patient in a simple hypertension diagnosis test. The number of features in such a case is therefore equal to three. However, in real medical datasets, the number of features is often larger, e.g. 22 in [1], 30 in [2] or 33 in [3]. For the DNA microarray databases, in the raw data this number may even reach 60,000, since they store the gene expression in a mass [4].

In classification problems, the use of all data features may contribute to computational complexity and, additionally, to the decrease of the generalization ability of the machine learning models utilized for prediction purposes. This results from a possible use of some irrelevant information. In such a case, one should conduct a search for the subset of features which contribute to the decrease of the prediction ability and remove this subset from all input vectors. In general, this issue is treated as the problem of dimensionality reduction for the input vectors. The dimensionality reduction can be solved in two ways, i.e. by applying feature selection or with the use of feature extraction.

Feature selection selects a subset of features out of an entire set of attributes. As a consequence, a lower dimensionality input space is obtained. Within the process of feature selection, no data transformation takes place – the original values of selected features are retained. The existing feature selection approaches for supervised classification problems can be divided into two main groups: filter approaches and wrapper approaches. In filter approaches, the process of feature selection is separated from the learning algorithm of the classifier. The important features are chosen by measuring the general statistics of the training data such as the correlation between individual features and output class labels. Below, four state-of-the-art filter based approaches are shortly described:

- FOCUS algorithm [5] – finds the minimum combination of features which are associated with a single class (the approach called "min-features bias"). FOCUS starts with an empty feature set and carries out breadth-first search until it finds the optimal subset of features with respect to generalization ability. In [5], FOCUS is compared with ID3 [6] and FRINGE [7] algorithms where it exhibits a higher generalization ability using fewer training vectors.
- Fast correlation based filter approach [8] – identifies relevant features and redundancy among these features without pairwise correlation analysis. For this purpose, the concept of predominant correlation between a feature and the class is introduced. Predominant features are used as the subset of new data features.
- Relief algorithm [9] – computes the weights of features which reflect how well their values distinguish between instances that are near to each other, taking into account the output class. The justification is the fact that a good feature should have a different value for vectors of the opposite class and it should have the same value for vectors from the same class. Relief is created for two-class classification problems with discrete and continues features.
- ReliefF algorithm [10] – is the extension of Relief. In contrast to Relief, where only two nearest vectors of different classes are found for a given instance, ReliefF searches k-nearest neighbours and is able to deal with missing data and multi-class classification problems. Moreover, ReliefF can also be applied in regression problems (RReliefF [11]).

It is worth to add that the decision tree algorithms have also been applied to the selection of feature subsets for use by machine learning models. In [12] and [13], C4.5 and greedy decision tree algorithm are utilized to determine the features of the input vectors which are then classified by means of the k-nearest neighbour classifier and the Bayesian network, respectively.

This work was supported in part by Rzeszów University of Technology Grant No. U–596/DS.

M. Kusy is with the Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, Powstańców Warszawy 12, 35-959 Rzeszów, Poland, e-mail: mkusy@prz.edu.pl.

The second method commonly applied in the field of feature selection is known as the wrapper approach. In the wrapper approach, a predefined classifier evaluates the quality of selected features in the way that the search in the space of possible feature subsets is conducted and the accuracy of the classifier is estimated on each subset. The classifier with the highest performance determines which features are selected as the final set to train a model. Performance assessments are usually determined by cross-validation [14], [15]. In this way, the wrapper approach finds features which are better suited to a learning algorithm of a predefined classifier. However, for n features data set, there are 2^n possible feature subsets, therefore the approach is computationally expensive (in comparison to filter methods). The hill-climbing, best-first, branch-and-bound, simulated annealing or genetic algorithms are frequently used strategies for feature subset selection. Among all of these, greedy search strategies are computationally advantageous and robust against overfitting [15]. These are solved by forward selection (where the search starts with the empty set of features) or backward elimination (where the search starts with the full set of features). In the former solution, features are progressively added into the subsets while in the later one, the least promising features are progressively removed.

It is observed that the filter methods are computationally efficient in comparison with the wrapper approaches since they are independent on a predefined classifier. Though, they do not take into consideration the biases of the classifier. The wrapper approaches, in turn, use the classifier to assess the performance of selected features, but the evaluation of this performance is computationally expensive since it must be performed many times. There is a solution created to bridge the gap between filter and wrapper methods. It is called the embedded approach [6], [16]. It interacts with the classifier and integrates the feature selection stage into the model training process [17]. Additionally, it is usually faster than the wrapper approach.

Feature extraction, in contrast to feature selection, relies on the construction of new features which are a linear combination of the original features. The new feature space is created with a lower dimensionality. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are the most popular techniques used for feature extraction. For example, in [18], PCA and LDA are applied for feature extraction in the recognition of over 3,276 color images of the faces of 126 subjects. The recognition is performed by using the nearest-neighbor algorithm with the Euclidean distance measure. In [19], a comparative study of PCA and LDA (and additionally Independent Component Analysis) methods is conducted in the classification of the FERET data set using the nearest neighbor classifier. [20] presents the impact of various feature extraction methods (including PCA) on the performance of the k-nearest neighbour, Naïve Bayes and C4.5 classifiers. The experiments are conducted on the UCI benchmark data sets. The authors of [21] utilize PCA for feature extraction of electroencephalogram signals. The principal components are used as inputs for radial basis function neural network.

The use of relevant features in the training data is particularly significant in classification tasks when probabilistic neural network (PNN) is applied. The complexity of the original PNN model proposed by Specht [22] is not influenced by the dimensionality of the data set, however, various modifications of this network have already been proposed. PNN has a single training parameter, the smoothing parameter (σ), which has to be optimized in order to make the network achieve the highest prediction ability. σ can take the value of **(a)** a scalar, **(b)** a vector of the length equal to the dimensionality (n) of the input vector or **(c)** a matrix, whose size is equal to $n \times G$, where G denotes the number of classes among the data. As one can observe, in the cases **(b)** and **(c)**, the number of features influences the complexity of PNN and its training process. The rejection of irrelevant features can therefore "simplify" the structure of this classifier, improve its generalization ability and shorten the computational time needed to complete the classification task. The problem of feature selection for PNN has not been profoundly explored up to this date. There is a study where informative patterns are selected from four data sets by means of three filter approaches [23]: the chi-square statistic, the ReliefF method, and the correlation-based feature selection method. The solution is applied in the molecular classification of cancer [24].

In this paper, the problem of dimensionality reduction of the input vectors in medical data classification tasks conducted by PNN is studied. The problem is solved in two ways. The first solution relies on performing feature selection which is based on a single decision tree (SDT) and a random forest (RF) algorithm. In the case of the SDT approach, a single decision tree is created and the tree nodes are used as data features. In the RF approach, the variable importance procedure is utilized to determine the set of features. The second solution is achieved by utilizing one of the feature extraction procedures, i.e. the PCA algorithm. PCA is applied to the input vectors and the principal components are used as new features. Three PNN models are explored, for which the smoothing parameter is determined according to **(a)**, **(b)**, and **(c)** possible forms. The classifiers are tested in six classification tasks by assessing their prediction ability on original and reduced training sets.

This paper is composed of the following sections. Section II discusses the principle of operation of the probabilistic neural network. In Section III, SDT, RF and PCA algorithms are shortly described in the context of feature selection and extraction. Here, the proposed dimensionality reduction approaches are also presented. Their use for data classification problems solved by means of the PNN model is justified. Section IV provides the profound comparative analysis of the classification performance of PNN models trained on original data sets and data sets where the features are determined by means of SDT, RF and PCA based approaches. In Section V, a short experimental study concerned with the results of feature selection and extraction in medical classification problems performed by various machine learning algorithms is outlined. The paper is concluded in SectionVI.

II. PROBABILISTIC NEURAL NETWORK

Probabilistic neural network is a feedforward model. In the first layer, PNN is composed of n neurons which represent features x_{ij} $(i = 1, \ldots, l,\ j = 1, \ldots, n)$ of an input vector $\mathbf{x}_i \in \mathbb{R}^n$. The second layer, called the pattern layer, consists of as many neurons as training vectors. These neurons are activated by means of radial basis functions computed between training vectors and a test vector. Pattern neurons feed the signal to the next summation layer. There are G neurons in the summation layer, where G represents the number of classes. Each gth summation neuron $(g = 1, \ldots, G)$ acquires the inputs measured over all the vectors of the gth class. Therefore, l_g pattern neurons constitute the input for the gth summation neuron. Finally, the output layer determines the category for the vector $\mathbf{x}$ in accordance with the Bayes's theorem on the basis of the outputs of all the summation layer neurons

$$G^*(\mathbf{x}) = \arg\max_g \{y_g(\mathbf{x})\}, \tag{1}$$

where $G^*(\mathbf{x})$ denotes the predicted class for the vector $\mathbf{x}$ and $y_g(\mathbf{x})$ is the summation layer signal defined as follows

$$y_g(\mathbf{x}) = \frac{1}{l_g (2\pi)^{n/2} \prod_{j=1}^{n} h_j^{(g)}} \sum_{i=1}^{l_g} \exp\left(-\sum_{j=1}^{n} \frac{\left(x_{ij}^{(g)} - x_j\right)^2}{2\left(h_j^{(g)}\right)^2} \right), \tag{2}$$

where $h_j^{(g)}$ is an element of the smoothing parameters' matrix $\mathbf{H} = \{h_j^{(g)}\}_{G \times n}$. The architecture of the probabilistic neural network is illustrated in Figure 1.

There are different ways of representing the smoothing parameter in (2) for PNN:

- $h_j^{(g)} = \sigma$; the smoothing parameter takes the form of a scalar which is used for all the neurons in the pattern layer (the network denoted as PNN1);
- $h_j^{(g)} = \sigma_j$; the smoothing parameter is related to each jth input feature so that all the pattern neurons of the network are activated by $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_n]$ (the network denoted as PNN2);
- $h_j^{(g)} = \sigma_j^{(g)}$; the smoothing parameter is determined for each jth feature of the input vector and for each gth class. Hence, the hidden neurons associated with the gth class are activated by $\boldsymbol{\sigma}^{(g)} = [\sigma_1^{(g)}, \ldots, \sigma_n^{(g)}]$, $g = 1, \ldots, G$ (the network denoted as PNN3).

For reading convenience, the indices g and j will skipped when referring to $h_j^{(g)}$. Henceforth, the smoothing parameter will be abbreviated to h.

We can observe that the h parameter computed for each feature and class creates the most general form of the PNN classifier. However, this type of network is the most demanding computationally since $G \times n$ matrix of the smoothing parameters must be stored. PNN training procedures, such as the conjugate gradient [25] or reinforcement learning [26], are very sensitive to the representation of h, particularly when PNN2 and PNN3 are considered. Therefore, the reduction of data dimensionality may contribute to finding the final solution in a smaller number of steps of a given training algorithm.

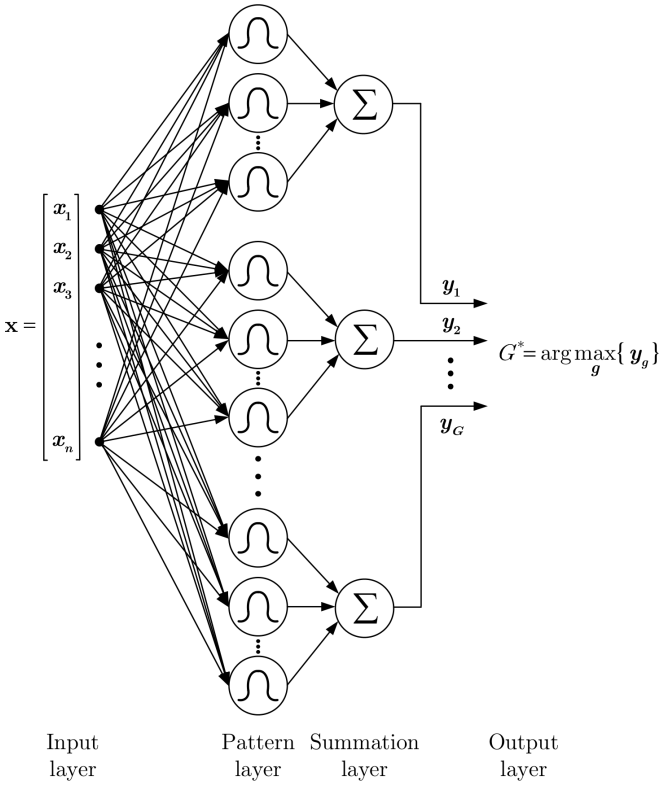

Fig. 1. The structure of the probabilistic neural network.

Furthermore, obtaining an optimal subset of features can create a more general network in terms of prediction ability.

III. APPROACHES USED FOR DIMENSIONALITY REDUCTION

In this section, the theory of single decision tree, random forest algorithm and principal component analysis is shortly described. Thereafter, the approaches for dimensionality reduction based on SDT, RF and PCA are introduced. This section concludes with the motivation for the current work.

A. Single decision tree

SDT, originally introduced by Hunt [27] and then independently in [28] and [29], is a hierarchical structure which, by means of a graph, is used to aid in a decision process. As the learning algorithm, SDT is treated as a predictive model – it maps the input data into desired targets. If the desired targets take the form of classes to which the data belong, SDT is a classification tree.

In this work, the C4.5 implementation [30], [31] of the decision tree is utilized for feature selection. It is shortly highlighted below.

SDT is composed of three types of elements: nodes, branches and leafs. Each node represents a split, i.e. a data partitioning based on the values of a selected feature. The splits are represented in a different way for discrete and continuous features. For a discrete feature, there is a single branch for each possible value (or a group of values). In the case of continuous feature, a threshold Z is computed which creates a binary split

for this feature. In general, the root, the node at the top level of the tree, is chosen based on class impurity. The remaining features are available for selection in lower-level nodes. The branches represent the values of particular features while the leafs are the class labels.

The process of tree growing relies on the appropriate selection of tree nodes. In the C4.5 algorithm, the choice of the nodes is performed according to the criterion of the information evaluation. The information is conveyed by a message and depends on its probability [30]

$$p\left(S, C_g\right) = \frac{\mathit{freq}\left(C_g, S\right)}{|S|}, \tag{3}$$

where $\mathit{freq}\left(C_g, S\right)$ stands for the number of cases in S that belong to class C_g and $|S|$ is cardinality of S; $p\left(S, C_g\right)$ is therefore a proportion of cases in S that belong to the gth class. (3) allows the computation of the amount of information conveyed by the message which is equal to $-log_2\left(p\left(S, C_g\right)\right)$ and can be measured in bits.

In order to build a decision tree using the C4.5 algorithm, the following stages are initially performed:

1) Find the expected information on the membership of the classes in set S by computing the entropy of S

$$\mathit{Info}(S) = -\sum_{g=1}^{G} p\left(S, C_g\right) log_2\left(p\left(S, C_g\right)\right). \tag{4}$$

2) Calculate the expected information measure to partition cases in S in accordance with K outcomes of a feature f

$$\mathit{Info}_f(S) = \sum_{k=1}^{K} \frac{|S_k|}{|S|} \mathit{Info}(S_k). \tag{5}$$

3) Compute the amount of information gained by partitioning S with respect to the feature f

$$\mathit{Gain}(f) = \mathit{Info}(S) - \mathit{Info}_f(S). \tag{6}$$

The formula in (6) determines the gain criterion used to select a feature for a node of the tree.

The above three stages cover the basis of Quinlan's ID3 algorithm. However, as reported in [30], the gain criterion in (6) has a strong bias in favor of features with many outcomes. This bias can be revised by introducing normalization

$$\mathit{SplitInfo}(f) = -\sum_{k=1}^{K} \frac{|S_k|}{|S|} log_2\left(\frac{|S_k|}{|S|}\right), \tag{7}$$

which represents information generated by splitting S into K subsets. Therefore, the gain related to the multiple value features is adjusted. At the final stage of the C4.5 algorithm, in order to provide the proportion of useful information generated by the above splitting, the gain ratio is computed

$$\mathit{GainRatio}(f) = \frac{\mathit{Gain}(f)}{\mathit{SplitInfo}(f)}. \tag{8}$$

The feature with the highest gain ratio is taken as the node of the decision tree.

The entire process of tree growing can be shortly summarized as follows. For all the cases in the set S, the feature f is selected with the highest gain ratio. This feature is set as the node of the tree. On the basis of f, the entire set S is partitioned into $S_1, \ldots, S_K$ subsets. Then again, for each subset S_k, the feature f_k is selected for node representation on the basis of the highest value of (8). The partitioning procedure is performed until some stopping criteria are met, e.g.: all the cases in S_k are from the same class, a minimum size of a node to split is reached (there are fewer input vectors in a group than the specified value), a maximum number of levels in a tree is achieved.

It can turn out that within the process of tree generation some features provide a minor gain of information in terms of prediction ability. In such a case, they are not added to the tree. Moreover, if the tree is constructed with all features included as nodes, some nodes might still be removed from the tree if it is pruned. Then, in either case, the nodes which remain in the tree constitute the subset of the original set of features. Therefore, the tree can be treated as the model, which performs a feature selection process.

B. Random forest

RF algorithm, proposed by Breiman in [32], utilizes the collection of independent decision trees. Within the training process, the trees grow in parallel, not interacting until all of them have been built. Once the training is completed, predictions of single trees are combined to make the overall prediction of RF.

The classification process conducted by RF algorithm can be summarized as follows:

1) Assume T number of trees in RF. It is advised to use large values of T, however, T should be based on the prediction performance of RF.
2) For each t ($t = 1, \ldots, T$), select with replacement a random sample of $s < l$ input vectors from the data set (the process called bagging). The remaining vectors are called out of bag (OOB) vectors.
3) Construct T decision trees for all T input vectors' subsets selected in step 2. Do not prune the trees. Within the tree growing, perform feature bagging, i.e., choose a random subset of features as candidates for a split. In this way, if the features are important in context of target prediction, they will be selected in a majority of T trees. It is recommended to use $\sqrt{n}$ features as candidates for each node split [32].
4) Record the predicted value for a new input vector running it through each tree in RF. Use the predicted classes for each tree as "votes" for the best class.
5) Use the class with the highest number of votes as the predicted category for a new input vector.

It is important to note that except for the classification capabilities, within RF training process, the procedure of variable importance can be invoked. According to Breiman, the procedure is the following. After each tree is constructed, the values of the jth feature in the OOB vectors are randomly permuted. All the OOB vectors are put down the tree. The classification result for each permuted OOB vector is saved. The same scheme is repeated for $j = 1, \ldots, n$ features each

time recording the classification outcome. The measure for variable importance utilizes the difference between the number of votes for the correct class in the OOB vectors where jth feature is permuted and the number of votes for the correct class in the original OOB vectors. The average of this number over all trees in RF is the importance score for feature j.

C. Principal component analysis

PCA, invented in 1901 by Pearson [33] and then independently in [34], is a statistical technique which converts a set of input features into a set of new values by means of linear transformation. The resulting features are called principal components and are linearly uncorrelated. The number of principal components is less than or equal to the number of original features. The principal components are created in the way that each succeeding component is orthogonal to the preceding ones. The first principal component has the highest variance among the data while the succeeding principal component has less variance in its direction.

The main advantage of PCA is the ability to identify patterns of similarities and differences in data. Once these patterns are determined, the data can be compressed by reducing the number of dimensions without much loss of information. PCA is one of the most frequently used feature extraction procedures.

In order to conduct the PCA analysis for the input vectors $\mathbf{x}_i = [x_{i1}, \ldots, x_{in}]$, $i = 1, \ldots, l$, the following steps are performed:

1) Obtain the mean for each jth feature over all vectors
$$\overline{x}_j = \frac{1}{l} \sum_{i=1}^{l} x_{ij}. \tag{9}$$
2) Create new data set $\mathbf{s}$ with mean subtracted from each input dimension
$$\mathbf{s}_j = [x_{1j} - \overline{x}_j, \ldots, x_{lj} - \overline{x}_j] \tag{10}$$
for $j = 1, \ldots, n$.
3) Calculate the covariance matrix
$$\mathbf{c} = \{c_{pq}\}_{n \times n}, \tag{11}$$
where
$$c_{pq} = \frac{1}{l-1} \mathbf{s}_p^T \mathbf{s}_q, \tag{12}$$
for $p, q = 1, \ldots, n$.
4) Compute the eigenvectors of the covariance matrix solving the following equation
$$\begin{cases} \mathbf{c}\mathbf{e}_k = \lambda_k \mathbf{e}_k, \\ \text{subject to constraints:} \\ \|\mathbf{e}_k\| = 1, \quad k = 1, \ldots, n, \end{cases} \tag{13}$$
where $\mathbf{e}_k$ is the kth eigenvector, λ_k is its corresponding eigenvalue. Each eigenvector found after solving (13) is called the principal component.
5) Find the number k, for which λ_k takes the highest value. The k index provides the first principal component $\mathbf{e}_k$, which has the largest variance between the data.
6) Order the eigenvectors by the eigenvalues, highest to lowest. In this way, the set of principal components is obtained in order of significance where each $\mathbf{e}_{k+1}$ is orthogonal to $\mathbf{e}_k$.
7) Choose p principal components ($p \leqslant n$) to store the most significant information: $\mathbf{e} = [\mathbf{e}_1, \ldots, \mathbf{e}_p]$ for which $\lambda_1 > \ldots > \lambda_p$. The principal components with lesser information can be ignored since their eigenvalues take smaller values. Thus, the resulting data has a lower number of features.
8) Derive the new data set composed of the vectors $\mathbf{n}_i = [\sum_{j=1}^{n} e_{1j} (x_{ij} - \overline{x}_j), \ldots, \sum_{j=1}^{n} e_{pj} (x_{ij} - \overline{x}_j)]$.

As the solution, a new set of transformed input vectors $[\mathbf{n}_1, \ldots, \mathbf{n}_l]$ is obtained.

D. Proposed approaches

The dimensionality reduction problem considered in this work is solved using two feature selection approaches which are based on SDT and RF classifiers, and the feature extraction procedure which requires PCA.

The SDT based approach relies on constructing the decision tree for a considered data set. Once the tree structure is obtained, the importance of the features can be read from this model. The root contains the maximum amount of information about the data, while the remaining nodes, appearing in a down-to-bottom direction, give the order of importance of the particular features. In this work all features used for splitting at all nodes of the constructed tree are selected to the training set. The parameters for the SDT model are as follows: the minimum number of rows allowed in a node is set to 5, the minimum size for a node to split is equal to 10, the maximum number of tree levels equals 8. The Gini or entropy split selection algorithms are applied to find the candidates for the split (the algorithm which yields better results for each dataset is finally used). All the parameters are adopted experimentally.

The RF based algorithm simply involves the calculation of the variable importance according to the idea presented in subsection III-B. The number of trees in the forest is set to $T = 300$. As recommended in [32], the number features used as candidates for each node split is equal to $\sqrt{n}$, where n stands for the number of features. As the result, the variable importance procedure gives a ranking of the overall importance of features. In this work, the importance score for the most important feature is scaled to a value of 100. Other features receive lower scores. The most optimal set of features in terms of PNN prediction ability is obtained according to the following procedure. First, a feature with 100 importance score is only used for input vectors' representation; the prediction ability of PNN is evaluated. Then, two features with the highest scores are selected for all vectors; the prediction ability of PNN is computed. Next, a third most important feature is added and the prediction ability of PNN is calculated. The procedure is repeated for all features preserving the ranking of importance. The highest PNN prediction ability determines the optimal subset of features.

The PCA based procedure requires performing PCA for a given database. Transformed input vectors are utilized as

the new input data for PNN. Initially, only a single principal component is involved in the classification task. Therefore, the data dimensionality for PNN is equal to one. In a further step, the PCA algorithm is performed again, additional principal component is added to the input vectors and the classification task is conducted. This procedure is repeated until all n possible components are determined. Therefore, the number of principal components is determined within the set $\{1, \ldots, n\}$. The optimal subset of principal components is selected for which the highest prediction ability of PNNs is achieved. It is assumed that all features are continues.

The conjugate gradient procedure is used for the training process of all PNN models. All simulations are performed in DTREG software [35].

E. Motivation of the study

The idea of introducing the dimensionality reduction for input vectors used to train PNN is motivated as follows:

- In real data classification problems, there is rarely substantial knowledge about relevant features. This results in the existence of irrelevant or redundant features among the input vectors.
- The decrease of the prediction ability of the machine learning models is often observed when they are trained on original input space. For example, the decision tree algorithms such as ID3, C4.5 or instance based learning methods degrade in prediction ability when facing many unnecessary features [36], [37].
- It is possible to improve predictive performance and reduce the risk of overfitting by applying feature selection or extraction procedures.
- Removing unwanted features in large-size data sets contributes to reducing of the classifier's training time.
- The computational complexity of the PNN model depends on the number of features. If the smoothing parameter of this network takes the form of an n-dimensional vector, there are n parameters to be optimized. In the case when the network is equipped with a matrix of the h's, there are $n \times G$ parameters for which the optimal value needs to be found.

It should also be pointed out that probabilistic neural network is a frequently exploited model in the field of data mining. It is applied in medical diagnosis and prediction [38], [39], [40], image classification and recognition [25], [41], [42], earthquake magnitude prediction [43] or classification in a time-varying environment [44].

IV. Results and discussion

In the simulations, six UCI machine learning repository medical data sets are used [45]: Pima Indians diabetes (PID), dermatology (D), diagnostic breast cancer (DBC), Statlog heart (SH), Parkinsons disease (PD), and breast tissue (BT). Table I presents the number of input vectors, features and classes for each considered database. For comparison purposes, a 10-fold cross validation (CV) error (E) is computed for the PNN models trained on original data sets and the data sets with reduced dimensionality.

TABLE I
Repository medical data sets used to test PNN models

Data set	Input vectors	Features	Classes
PID	786	8	2
D	358	34	6
DBC	569	30	2
SH	270	13	2
PD	195	19	2
BT	106	9	6

Tables II, III and IV show the lowest CV error results obtained in all data set classification problems by PNN1, PNN2 and PNN3, respectively. Columns labeled "ALL" indicate the results for PNNs tested on the training vectors with all input features. The "SDT" denoted columns show the outcomes of the networks tested on the data sets where the features are the decision tree nodes. In the columns marked with "PCA", the CV errors provided by the PNN models for the input vectors represented by principal components are set out. Finally, the columns named "RF" present the error values for the networks tested on the data set with features determined on the basis of variable importance procedure. The variables: n, f, p and v denote the number of features of the input vectors, the total number of tree nodes generated by SDT, the number of principal components and the number of important variables provided by RF, respectively. Both p and v are determined on the basis of the highest prediction ability of PNN, as explained in subsection III-D.

As one can observe, the application of the variable importance procedure to select features from the input vectors contributes to the decrease of the CV error value for PNN1, PNN2 and PNN3 models in all data classification tasks. However, in comparison to the remaining solutions, the reduction in the number of features is smallest for the RF based approach in almost all classification cases. This remark suggests that the RF based feature selection approach proposed in the current article may be revised. For example, an interesting idea would be to try multiple subsets of top ranked features not necessarily preserving the ranking of their importance. From Tables II–IV we can also see that the selection of features on the basis of decision tree nodes is an unsuitable approach: there is no CV error decrease for PNN2 and PNN3 models.

Since the RF based approach provides the highest prediction ability results for PNNs, it is worth to pay attention to the outcome format of variable importance procedure. For example, in PNN2 classification problem of BT data, the decrease on E rate is equal to 3.78%. Here, the following set of important variables is obtained:

- I0 (100.00) – impedivity (Ω) at zero frequency;
- PA500 (73.09) – phase angle at 500 KHz;
- DA (65.64) – impedance distance between spectral ends;
- P (63.67) – length of the spectral curve;
- Max IP (61.17) – maximum of the spectrum;
- Area (50.29) – area under spectrum.

The importance score is added in brackets, starting from the most important feature (I0). This result is achieved for $v = 6$

TABLE II
THE LOWEST CV ERROR VALUES (E, IN %) COMPUTED FOR PNN1 TRAINED ON THE INPUT VECTORS WITH ALL FEATURES (ALL) AND THE FEATURES SELECTED USING SDT, PCA AND RF BASED APPROACHES

	ALL		SDT		PCA		RF	
Data set	E	n	E	f	E	p	E	v
PID	24.87	8	**24.35**	2	25.39	5	**22.66**	4
D	4.30	34	**2.51**	7	4.75	7	**2.79**	16
DBC	3.69	30	3.69	5	5.97	4	**3.16**	15
SH	18.85	13	**14.33**	4	**17.77**	8	**14.82**	9
PD	8.21	19	**6.66**	3	**4.62**	15	**5.13**	17
BT	35.85	9	**31.13**	4	**31.13**	3	**31.13**	4

TABLE III
THE LOWEST CV ERROR VALUES (E, IN %) COMPUTED FOR PNN2 TRAINED ON THE INPUT VECTORS WITH ALL FEATURES (ALL) AND THE FEATURES SELECTED USING SDT, PCA AND RF BASED APPROACHES

	ALL		SDT		PCA		RF	
Data set	E	n	E	f	E	p	E	v
PID	22.39	8	24.74	2	**20.96**	6	**21.75**	5
D	0.84	34	1.95	7	**0.00**	19	**0.56**	30
DBC	2.43	30	3.16	5	2.98	18	**1.58**	27
SH	14.33	13	17.04	4	**13.33**	11	**13.70**	9
PD	3.07	19	6.15	3	**0.51**	16	**1.03**	16
BT	27.36	9	32.07	4	**25.47**	4	**23.58**	6

TABLE IV
THE LOWEST CV ERROR VALUES (E, IN %) COMPUTED FOR PNN3 TRAINED ON THE INPUT VECTORS WITH ALL FEATURES (ALL) AND THE FEATURES SELECTED USING SDT, PCA AND RF BASED APPROACHES

	ALL		SDT		PCA		RF	
Data set	E	n	E	f	E	p	E	v
PID	22.53	8	24.48	2	**21.22**	7	**21.48**	4
D	0.67	34	1.67	7	**0.28**	19	**0.00**	26
DBC	1.32	30	2.81	5	1.41	24	**0.53**	25
SH	9.26	13	15.56	4	9.63	11	**7.78**	10
PD	1.03	19	3.59	3	**0.51**	16	**0.51**	18
BT	16.04	9	28.30	4	**13.21**	8	**13.21**	6

TABLE V
COMPUTATIONAL TIME IN SECONDS ACHIEVED BY PNN1, PNN2 AND PNN3 TRAINED ON THE INPUT VECTORS WITH ALL FEATURES (ALL) AND THE FEATURES WHERE THE LOWEST CV ERROR VALUE IS RECORDED AFTER APPLYING SDT, PCA OR RF BASED APPROACH. THE LAST THREE COLUMNS SHOW THE DIMENSIONALITY REDUCTION COST IN SECONDS.

	PNN1 model			PNN2 model			PNN3 model			Reduction cost		
Data set	ALL	SDT	RF	ALL	PCA	RF	ALL	PCA	RF	SDT	PCA	RF
PID	6.39	**3.83**	**4.38**	29.43	**27.44**	**19.58**	51.30	53.10	**31.97**	0.65	0.12	0.94
D	4.16	**1.72**	**2.47**	42.19	**29.61**	**38.44**	37.81	**29.07**	**35.91**	0.14	0.42	0.19
DBC	13.27	**3.12**	**6.17**	158.50	**28.51**	**125.16**	242.00	**107.82**	**228.21**	1.26	0.50	1.31
SH	1.33	**0.63**	**1.17**	9.86	**7.66**	**4.24**	23.94	**16.10**	**15.55**	0.09	0.16	0.19
PD	1.13	**0.67**	**0.91**	11.95	**8.64**	**9.74**	21.73	**9.30**	**9.51**	0.17	0.24	0.24
BT	0.25	**0.19**	**0.17**	1.36	**0.53**	**0.66**	3.72	4.06	**1.93**	0.11	0.16	0.24

out of total $n = 9$ features. The subset of features outlined above is an optimal one for PNN2 model in terms of prediction ability.

Table V presents the computational time needed to complete the classification tasks by means of PNN1 PNN2 and PNN3, for which the highest E value reductions are obtained. Moreover, the dimensionality reduction cost of performing feature selection ("SDT", "RF") and feature extraction ("PCA") is added. The simulations are conducted in a 64-bit Windows 8.1 Pro operating system with an Intel Core i7 2.4-GHz processor and 8-GB RAM. As in the case of the CV error value decrease, also here, the computational savings always occur where RF based approach is used to select features: in each classification problem, PNN1, PNN2 and PNN3 require shorter running time to complete the task. The time required for performing feature selection or extraction is smaller in comparison to the PNN classification time and, in majority, depends on the size of the data set.

Figures 2, 3, 4, 5, 6, and 7 depict the influence of features selected by means of variable importance procedure on the

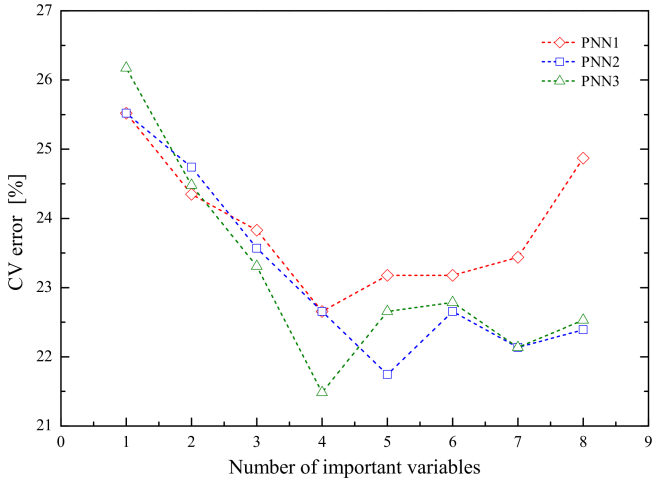

Fig. 2. Plot of the 10-fold cross validation error obtained after applying variable importance procedure in PID data classification task.

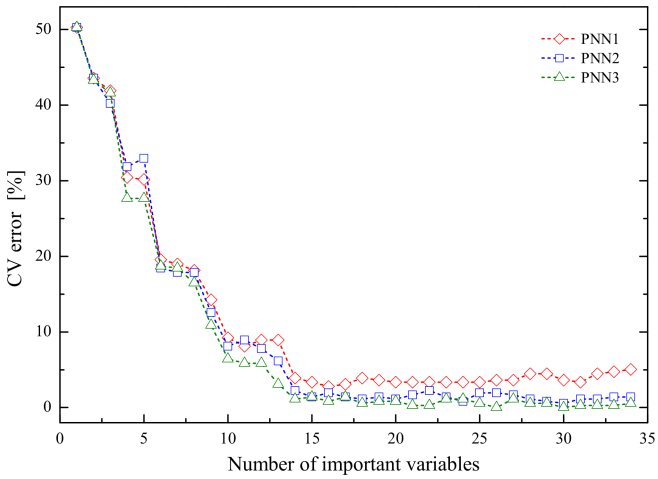

Fig. 3. Plot of the 10-fold cross validation error obtained after applying variable importance procedure in D data classification task.

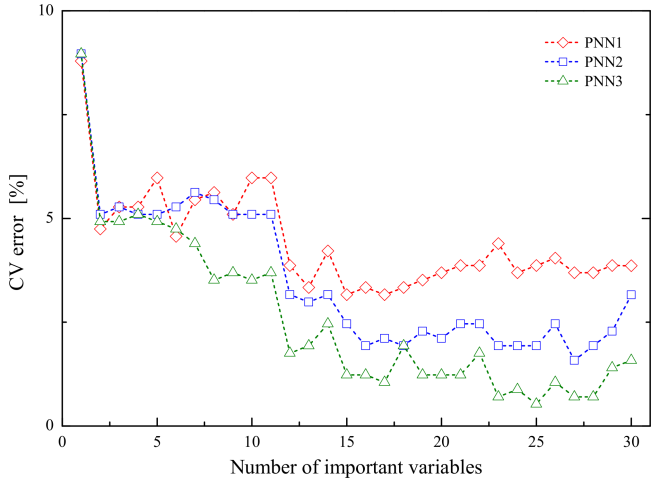

Fig. 4. Plot of the 10-fold cross validation error obtained after applying variable importance procedure in DBC data classification task.

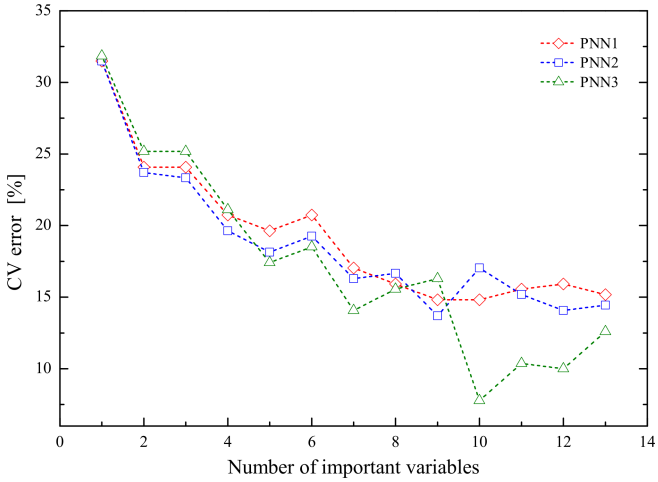

Fig. 5. Plot of the 10-fold cross validation error obtained after applying variable importance procedure in SH data classification task.

CV error values calculated respectively in PID, D, DBC, SH, PD, and BT data classification problems for PNN1, PNN2, and PNN3. It is important to note that the x-axis shows the number of features preserving a ranking of their overall importance. It can be seen from the figures that except for PID classification task performed by PNN1, the larger the set of features generated by the procedure of variable importance, the lower the CV error value.

Figures 8, 9, 10, 11, 12, and 13 illustrate the impact of the number of principal components on the CV error values which are computed respectively in PID, D, DBC, SH, PD, and BT data classification tasks for PNN1, PNN2, and PNN3, where the input space is expressed in terms of p dimensional data set $[\mathbf{n}_1, \ldots, \mathbf{n}_l]$. The x-axis represents the number of p principal components ($p = 1, \ldots, n$) and for each p, the classification task is performed. One can observe a similar pattern in the error changes for the PNN2 and PNN3 models. This error starts with larger values and then it gets smaller along with the increase of the p parameter. In the case of the PNN1 classifier, the CV error decreases for a smaller number of the principal components, but at a higher value of p, it begins to grow again.

The analysis shown in this work is of a particular importance since the probabilistic neural network is very sensitive to the number of features especially when PNN2 and PNN3 models are explored. Feature selection or extraction methods are well known and widely used approaches in the problem of dimensionality reduction for various classifiers. However, this topic has not been studied up to now for different types of PNNs, as the ones considered in this article. Any possible reduction in the number of features is important for this network, since PNN will have a simpler intrinsic structure, less memory will be required to complete the classification process, and the final solution will be achieved in a smaller number of training steps. What is most important, as it is shown in the present study, a higher prediction ability of the model can be obtained.

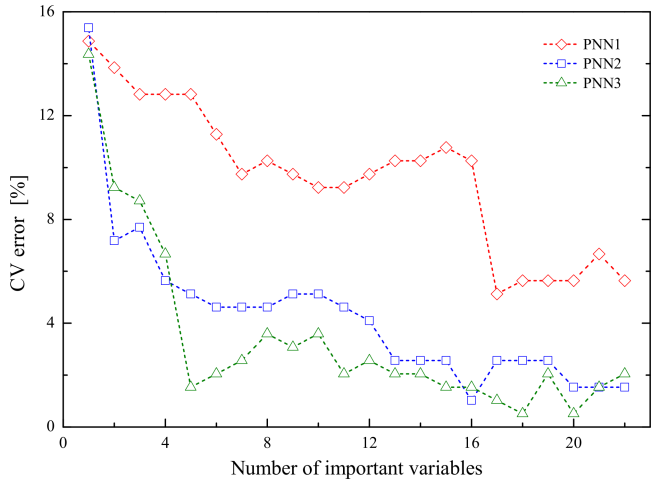

Fig. 6. Plot of the 10-fold cross validation error obtained after applying variable importance procedure in PD data classification task.

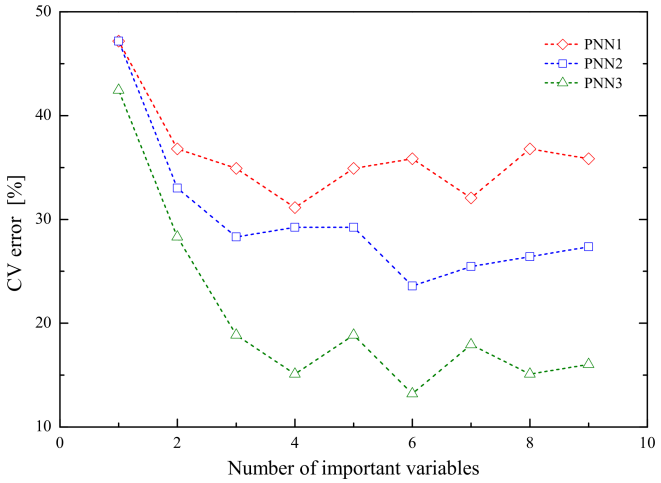

Fig. 7. Plot of the 10-fold cross validation error obtained after applying variable importance procedure in BT data classification task.

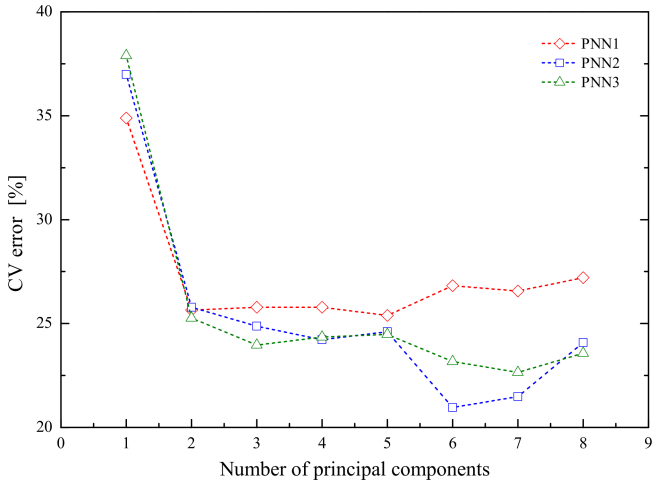

Fig. 8. Plot of the 10-fold cross validation error determined in relation to the number of principal components for PID data classification problem.

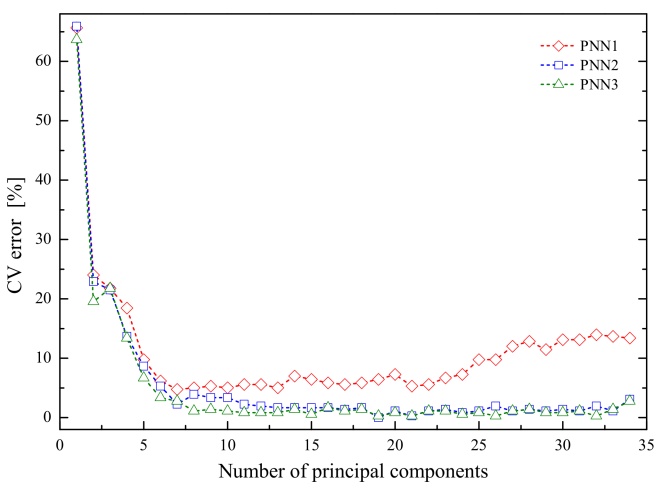

Fig. 9. Plot of the 10-fold cross validation error determined in relation to the number of principal components for D data classification problem.

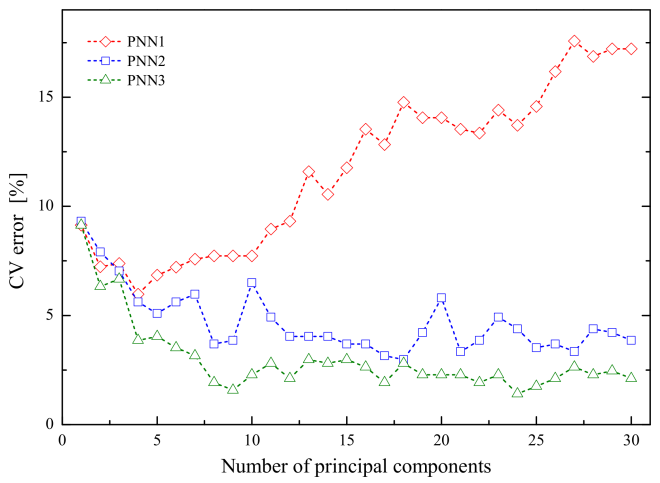

Fig. 10. Plot of the 10-fold cross validation error determined in relation to the number of principal components for DBC data classification problem.

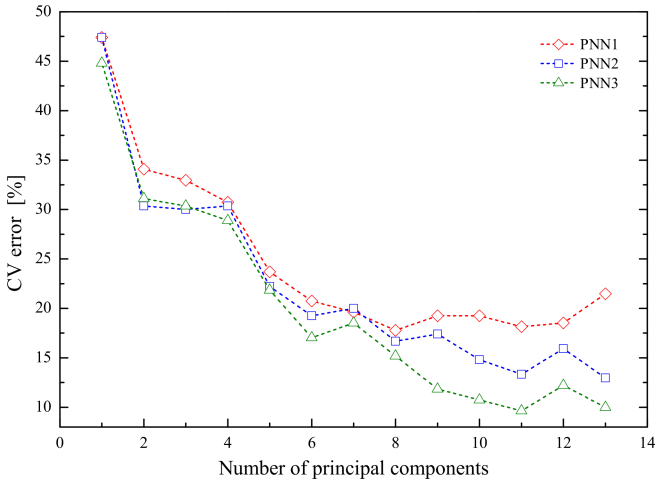

Fig. 11. Plot of the 10-fold cross validation error determined in relation to the number of principal components for SH data classification problem.

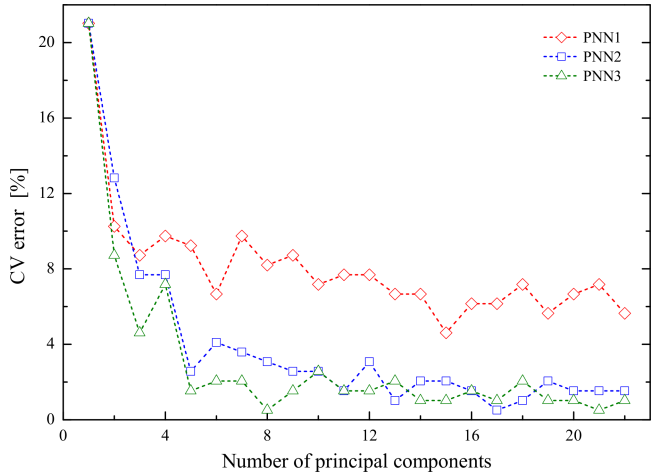

Fig. 12. Plot of the 10-fold cross validation error determined in relation to the number of principal components for PD data classification problem.

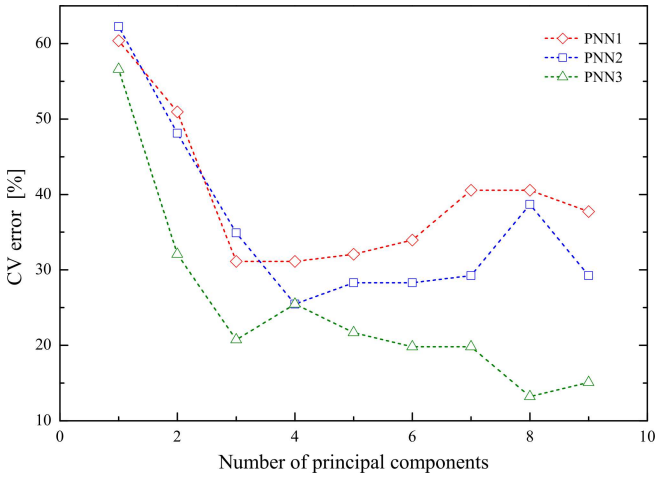

Fig. 13. Plot of the 10-fold cross validation error determined in relation to the number of principal components for BT data classification problem.

V. Related Experimental Studies

Feature selection algorithms have been frequently applied in medical classification tasks solved by various machine learning algorithms. Table VI, VII and VIII present the CV error results published in [14] for ID3, C4.5 and Naïve Bayes (NB) algorithms, respectively, obtained in the UCI medical data classification problems of Wisconsin breast cancer (WBC), Pima Indians diabetes (PID) and sick-euthyroid (S–E). The results are shown for the input vectors with: all features (ALL), features selected using ReliefF (RLF) filter and features selected by means of wrapper approach based on hill-climbing (HC) and best-first search (BFS) algorithms. In Table IX, the errors obtained in the classification problems of lung cancer (L), promoters (P) and arrhythmia (A) data sets received by C4.5 algorithm are set out [8]. The columns labeled with FCB and CSS denote respectively: fast correlation based filter and correlation subset search filter approach. In literature, one can also find studies in which classification algorithms are tested on data sets with extracted features. For example, the

TABLE VI
Classification error results for ID3 algorithm tested on original data sets and the data sets after feature selection according to [14]

Data set	ALL	RLF	HC	BFS
WBC	5.43	6.43	**5.29**	6.15
PID	31.27	36.09	**30.48**	32.56
S–E	3.32	**3.22**	**2.94**	**2.94**

TABLE VII
Classification error results for C4.5 algorithm tested on original data sets and the data sets after feature selection according to [14]

Data set	ALL	RLF	BFS
WBC	4.58	5.58	4.72
PID	28.40	35.82	29.82
S–E	2.27	2.27	**2.29**

authors of [46] display how the PCA transformation influences prediction ability of the $k = 3$ nearest neighbour (3–NN) model. The tests are performed on the UCI medical data sets (PID, DBC and SH). Table X shows error values on the input vectors with original features, transformed features in form of principal components (PCA), and principal components together with original features (PCA + ALL).

The results presented in the Tables VI–X lead to the following observations:

- Except for the S–E data set (results in Table VI), the application of the RLF approach to feature selection does not improve prediction ability of ID3, C4.5 and NB in all classification cases. For the PID database, the increase in the CV error reaches a margin of 4.82%, 7.42% and 11.33% for ID3, C4.5 and NB, respectively. Only in the S–E data classification task performed by all the models,

TABLE VIII
Classification error results for Naïve Bayes algorithm tested on original data sets and the data sets after feature selection according to [14]

Data set	ALL	RLF	HC	BFS
WBC	3.00	4.86	4.43	4.00
PID	24.10	35.43	25.66	**23.97**
S–E	4.36	4.36	**2.65**	**2.65**

TABLE IX
Classification error results for C4.5 algorithm tested on original data sets and the data sets after feature selection according to [46]

Data set	ALL	RLF	FCB	CSS
L	19.17	19.17	**12.50**	**15.83**
P	13.09	**10.36**	**12.27**	**12.27**
A	32.75	34.10	**27.21**	**31.42**

TABLE X
CLASSIFICATION ERROR RESULTS FOR 3–NN ALGORITHM TESTED ON ORIGINAL DATA SETS AND THE DATA SETS AFTER FEATURE EXTRACTION ACCORDING TO [8]

Data set	ALL	PCA	PCA + ALL
PID	26.20	29.40	26.60
DBC	3.20	6.50	3.20
SH	21.90	34.10	**21.20**

both HC and BFS wrapper approaches decrease the CV error value.

- The application of FCB and CSS filters for feature selection (Table IX) increases the prediction ability of C4.5 in all considered classification problems.
- The transformation of the features to the form of principal components, increases the error values in all classification problems performed by 3–NN model. However, the use of principal components along with the original features provides much better results, decreasing the error obtained for the SH data by 0.7%.

A profound experimental study on the different feature extraction techniques is presented in [20]. Two eigenvector-based approaches that take into account the class information are compared with conventional PCA, with random projection and with plain classification without feature extraction. The kNN, NNB and C4.5 classifiers are taken for analysis. The experiments are conducted on 20 UCI datasets. The experiments show that it is difficult to determine which technique is the most appropriate for a selected classifier and/or for a certain problem, but class-conditional feature extraction approaches are often the best ones.

VI. CONCLUSIONS

In this article, the problem of feature selection and extraction for medical data classification tasks conducted by the probabilistic neural network was explored. Feature selection approach was based on (1) utilizing the nodes of a single decision tree as data features and (2) applying the variable importance procedure to select the subset of optimal features. The feature extraction approach was realized by means of PCA performed on the input data sets and using principal components as new features. Three types of PNN models were examined: the network with a single smoothing parameter h for the whole model, the network with the vector of different h values for each data feature, and the network with various h's for each feature and class. The PNN models trained on original data sets were compared with the ones trained on data sets with reduced dimensionality by assessing the models' prediction ability. A 10-fold cross validation procedure was used for this purpose. The results showed that selection of the features by applying the variable importance procedure allowed the increase of the prediction ability for PNN1, PNN2 and PNN3 models in each classification task. Furthermore, the decrease in computational time needed to complete the classification tasks was observed in each classification case. The remaining solutions did not provide such satisfactory results.

REFERENCES

[1] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig, "Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease," *IEEE Transactions On Biomedical Engineering*, vol. 56, no. 4, pp. 1015–1022, 2009.

[2] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Operations Research*, vol. 43, no. 4, pp. 570–577, 1995.

[3] H. A. Guvenir, G. Demiroz, and N. Ilter, "Learning differential diagnosis of Eryhemato-Squamous diseases using voting feature intervals," *Artificial Intelligence in Medicine*, vol. 13, pp. 147–165, 1998.

[4] V. Bolon-Canedo, N. Sanchez-Marono, A. Alonso-Betanzos, J. M. Benitez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Information Sciences*, vol. 282, pp. 111–135, 2014.

[5] H. Almuallim and T. G. Dietterich, "Learning with many irrelevant features," in *Proceedings of the Ninth National Conference on Artificial Intelligence*, 1991, pp. 547–552.

[6] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[7] G. Pagallo and D. Haussler, "Boolean Feature Discovery In Empirical Learning," *Machine Learning*, vol. 5, no. 1, pp. 71–100, 1990.

[8] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," in *Proceedings of the Twentieth International Conference on Machine Learning*, 2003, Washington, USA.

[9] K. Kira and L. Rendell, "A practical approach to feature selection," in *Proceedings of the Ninth International Conference on Machine Learning*, 1992, pp. 249–256.

[10] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," *Lecture Notes in Computer Science*, vol. 784, pp. 171–182, 1994.

[11] M. Robnik-Sikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," *Machine Learning Journal*, vol. 53, pp. 23–69, 2003.

[12] C. Cardie, "Using Decision Trees to Improve Case-Based Learning," in *Proceedings of the Tenth International Conference on Machine Learning*, 1993, pp. 25–32.

[13] M. Singh and G. M. Provan, "Efficient learning of selective Bayesian network classifiers," in *Proceedings of the Thirteenth International Conference on Machine Learning*, Morgan Kaufmann, 1996.

[14] R. Kohavi and G. H. John "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, pp. 273–324, 1997.

[15] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[16] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, pp. 389–422, 2002.

[17] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[18] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[19] K. Delac, M. Grgic, and S. Grgic, "Independent comparative study of PCA, ICA, and LDA on the FERET data set," *International Journal of Imaging Systems and Technology*, vol. 15, no. 5, pp. 252–260, 2005.

[20] M. Pechenizkiy, "The Impact of Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5," in B. Kegl and G. Lapalme (Eds.) AI 2005, *Lecture Notes in Artificial Intelligene*, vol. 3501, Springer-Verlag Berlin Heidelberg, pp. 268–279, 2005.

[21] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr, "Principal Component Analysis-Enhanced Cosine Radial Basis Function Neural Network for Robust Epilepsy and Seizure Detection," *IEEE Transactions On Biomedical Engineering*, vol. 55, no. 2, pp. 512–518, 2008.

[22] D. F. Specht, "Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification," *IEEE Transaction on Neural Networks*, vol. 1, no. 1, pp. 111–121, 1990.

[23] C.–J. Huang and W.–C. Liao, "A Comparative Study of Feature Selection Methods for Probabilistic Neural Networks in Cancer Classification," in *15th IEEE International Conference on Tools with Artificial Intelligence*, 2003, pp. 451–458.

[24] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular Classification of Cancer:

Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, pp. 531–537, 1999.
[25] Y. Chtioui, S. Panigrahi, and R. Marsh, "Conjugate gradient and approximate Newton methods for an optimal probabilistic neural network for food color classification," *Optical Engineering*, vol. 37, pp. 3015–3023, 1998.
[26] M. Kusy and R. Zajdel, "Application of reinforcement learning algorithms for the adaptive computation of the smoothing parameter for probabilistic neural network," *IEEE Transaction on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2163–2175, 2015.
[27] E. B. Hunt, J. Marin, and P. J. Stone, *Experiments in induction*. New York, USA: Academic Press, 1966.
[28] J. H. Friedman, "A recursive partitioning decision rule for nonparametric classification," *IEEE Transactions on Computers*, pp. 404–408, 1977.
[29] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. USA: Chapman and Hall/CRC, 1984.
[30] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, USA: Morgan Kaufmann, 1993.
[31] J. R. Quinlan, "Improved Use of Continuous Attributes in C4.5," *Journal of Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.
[32] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
[33] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, vol. 2, no. 11, pp. 559–572, 1901.
[34] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.
[35] P. H. Sherrod, "DTREG predictive modelling software," 2015. Available: http://www.dtreg.com
[36] D. W. Aha, "Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms," *International Journal on Man-Machine Studies*, vol. 36, pp. 267–287, 1992.
[37] S. B. Thrun et al., "The Monk's problems: a performance comparison of different learning algorithms," Technical report CMU-CS-91-197, Carnegie Mellon University, Pittsburgh, PA, 1991.
[38] I. Maglogiannis, E. Zafiropoulos, and I. Anagnostopoulos, "An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers," *Applied Intelligence*, vol. 30, pp. 24–36, 2009.
[39] D. Mantzaris, G. Anastassopoulos, and A. Adamopoulos, "Genetic algorithm pruning of probabilistic neural networks in medical disease estimation," *Neural Networks*, vol. 24, pp. 831–835, 2011.
[40] R. K. Orr, "Use of a Probabilistic Neural Network to Estimate the Risk of Mortality after Cardiac Surgery," *Medical Decision Making*, vol. 17, pp. 178–185, 1997.
[41] E. Kyriacou, M. S. Pattichis, C. S. Pattichis et al., "Classification of atherosclerotic carotid plaques using morphological analysis on ultrasound images," *Applied Intelligence*, vol. 30, pp. 3–23, 2009.
[42] S. Ramakrishnan and S. Selvan, "Image texture classification using wavelet based curve fitting and probabilistic neural network," *International Journal of Imaging Systems and Technology*, vol. 17, pp. 266–275, 2007.
[43] H. Adeli and A. Panakkat, "A probabilistic neural network for earthquake magnitude prediction," *Neural Networks*, vol. 22, pp. 1018–1024, 2009.
[44] L. Rutkowski, "Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment," *IEEE Transactions on Neural Networks*, vol. 15, pp. 811–827, 2004.
[45] K. Bache and M. Lichman, "UCI Machine Learning Repository," School of Information and Computer Science, University of California, Irvine, CA, USA, Technical Report, 2013. [Online]. Available: http://archive.ics.uci.edu/ml.
[46] M. Pechenizkiy, A. Tsymbal, and S. Puuronen, "PCA–based Feature Transformation for Classification: Issues in Medical Diagnostics," in *Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems*, 2004, pp. 535–540.