Proceedings of the Federated Conference on Computer Science and Information Systems pp. 39–48
DOI: 10.15439/2017F225
ISSN 2300-5963 ACSIS, Vol. 11

Determining the significance of features with the use of Sobol method in probabilistic neural network classification tasks

Piotr A. Kowalski[†‡]
[†]Faculty of Physics and Applied Computer Science,
AGH University of Science and Technology,
al. A. Mickiewicza 30, 30-059 Cracow, Poland,
Email: pkowal@agh.edu.pl
[‡]Systems Research Institute,
Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw, Poland,
E-mail: pakowal@ibspan.waw.pl

Maciej Kusy[*]
[*]Faculty of Electrical and Computer Engineering,
Rzeszow University of Technology,
al. Powstancow Warszawy 12, 35-959 Rzeszow, Poland,
Email: mkusy@prz.edu.pl

***Abstract*—In this article, the problem of determining the significance of data features is considered. For this purpose the algorithm is proposed, which with the use of Sobol method, provides the global sensitivity indices. On the basis of these indices, the aggregated sensitivity coefficients are determined which are used to indicate significant features. Using such an information, the process of features' removal is performed. The results are verified by the probabilistic neural network in the classification of medical data sets by computing model's quality. We show that it is possible to point the least significant features which can be removed from the input space achieving higher classification performance.**

I. Introduction

GLOBAL sensitivity analysis (GSA) embraces a group of algorithms which determine the influence of the input of the model to the model's output. This gives the possibility of estimating how the model output variance is influenced by relative impact of a single input variable and the interactions between them. In GSA, the influence on the output of the model can be assessed by means of regression methods, screening approaches [1], and the variance-based techniques, e.g., Sobol method [2], [3], the Fourier amplitude sensitivity test (FAST) [4], or the extended (EFAST) [5].

In literature, we can find a lot of contributions devoted to applications of GSA to feature selection. For example, in [6], the Sobol method is applied in optimization of shell and tube heat exchangers; the non-influential geometrical parameters which have the least effect on total cost of tube heat exchangers are identified. In turn, in [7], a new GSA based algorithm for the selection of input variables of neural network is proposed. The algorithm ranks the model's inputs according to their importance in the variance of the network output. In reference [8], one can find the use of the standardized regression coefficients, Morris screening and EFAST methods in assessing the most relevant processes occurring in wastewater treatment systems. The aforementioned methods are applied to a complex integrated membrane bioreactor where various interactions among the input factors are detected. The authors of current work utilize the GSA methods in the domain of neural network structure reduction. In [9], we present how the structure of the probabilistic neural network (PNN) can be optimized by means of Sobol, FAST and EFAST methods.

It is important to note that, in addition to GSA based techniques, many other approaches exist which can be utilized for feature selection. For example, ReliefF algorithm, proposed by Kira and Rendell in [10], computes the weights for data set features. This shows how well the feature values distinguish among patterns which are near to each other, taking into account the output class. On the basis of the weight values, the feature significance can be established. Similarly, Breiman's random forest algorithm [11], within its training process, invokes variable importance procedure. This procedure provides a ranking of the overall relevance of features. On the other hand, the extended version of Naïve Bayes classifier, presented in [12], determines the importance of features in classification process by means of weights of a normalized neural network. The weights are obtained by the backpropagation-like technique applied to the model training. The appropriate connection between the network and the classifier is implemented. The attribute clustering algorithms are also utilized to construct informative subset of available features from high dimensional data. The authors of [13] propose such a solution along with an attribute similarity measure which is useful for identifying groups of features that are likely to be selected for reduction purposes.

In this study, we propose the algorithm for determining the significance of input features. This significance is obtained using Sobol method. For the analysis, the UCI machine learning

The work was supported by Rzeszow University of Technology, Department of Electronics Fundamentals Grant for Statutory Activity (DS 2017).

repository (UCI–MLR) data sets [14] are used. The algorithm is tested in the classification problems conducted using PNN; the correctness of operation is verified by computing the learning and test qualities.

This article is organized as follows. In section II, the Sobol sensitivity analysis fundamentals are provided. Section III, introduces the PNN model highlighting its architecture and training algorithm. In section IV, the algorithm for determining the significance of input features is proposed. Section V presents numerical verification results achieved by the proposed algorithm. In section VI, we shortly summarize our work.

II. Sobol sensitivity analysis

Sobol method is based on decomposition of the model output variance into summands of variances of the input parameters in increasing dimensionality [2], [15]. It establishes the contribution of each input variable and the interactions between them to the overall variance in the output of the model. This is achieved by computing the first-order, second-order, higher-order and the overall sensitivity indices. Below, we show how to determine this contribution of variables according to the Sobol approach.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ be the set of mutually independent input parameters in which $x_i \in I^N$ where I denotes [0,1] interval and I^N is the N–dimensional unit hypercube. The model output, whose sensitivity to the parameters $x_1, x_2, \ldots, x_N$ is to be determined, is an integrable function $f(\mathbf{x})$ defined in I^N

$$f(\mathbf{x}) = f_0 + \sum_{s=1}^{N} \sum_{i_1<i_2<\cdots<i_s}^{N} f_{i_1 i_2 \cdots i_s}(x_{i_1}, x_{i_2}, \ldots, x_{i_s}). \quad (1)$$

It can be seen that the overall number of summands in (1) is 2^N. Equation (1) can be rewritten in the following form

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^{N} f_i(x_i) + \sum_{i=1}^{N} \sum_{j=i+1}^{N} f_{ij}(x_i, x_j) + \ldots + f_{12\cdots N}(x_1, x_2, \ldots, x_N). \quad (2)$$

Formula (1) is called ANOVA-representation of $f(\mathbf{x})$ if the integral of each summand over each of its own variables is zero

$$\int_0^1 f_{i_1 i_2 \cdots i_s}(x_{i_1}, x_{i_2}, \ldots, x_{i_s}) dx_k = 0 \quad (3)$$

for $k = i_1, i_2, \ldots, i_s$ where both $\{i_1, i_2, \ldots, i_s\}$ and s run from 1 to N.

Some important remarks can now be inferred. First of all, the integration of (1) over I^N yields

$$\int_0^1 f(\mathbf{x}) d\mathbf{x} = f_0. \quad (4)$$

which allows for computing the term f_0. Further, after integrating (1) over all variables excluding x_i one obtains

$$\int_0^1 f(\mathbf{x}) \prod_{k \neq i} dx_k = f_0 + f_i(x_i), \quad (5)$$

which provides

$$f_i(x_i) = \int_0^1 f(\mathbf{x}) \prod_{k \neq i} dx_k - f_0. \quad (6)$$

Similarly, integrating (1) over all variables excluding x_i and x_j defines the term $f_{ij}(x_i, x_j)$ as follows

$$f_{ij}(x_i, x_j) = \int_0^1 f(\mathbf{x}) \prod_{k \neq \{i,j\}} dx_k - f_i(x_i) - f_j(x_j) - f_0. \quad (7)$$

The procedure is performed until last term $f_{12\cdots N}(x_1, x_2, \ldots, x_N)$ is determined.

Assuming that $f(\mathbf{x})$ is square integrable over I^N, all terms $f_{i_1 i_2 \cdots i_s}$ in (1) are also integrable. Thus

$$\int_0^1 f^2(\mathbf{x}) d\mathbf{x} - f_0^2 = \sum_{s=1}^{N} \sum_{i_1<i_2<\cdots<i_s}^{N} \int_0^1 f_{i_1 i_2 \cdots i_s}^2 dx_{i_1} \cdots dx_{i_s}. \quad (8)$$

The left side of (8) is called the total variance of $f(\mathbf{x})$

$$D = \int_0^1 f^2(\mathbf{x}) d\mathbf{x} - f_0^2 \quad (9)$$

while

$$D_{i_1 \cdots i_s} = \int_0^1 f_{i_1 i_2 \cdots i_s}^2 dx_{i_1} \cdots dx_{i_s} \quad (10)$$

are the partial variances for each term in (1). Using (8)–(10) we receive

$$D = \sum_{s=1}^{N} \sum_{i_1<i_2<\cdots<i_s}^{N} D_{i_1 \cdots i_s}, \quad (11)$$

which means that

$$D = \sum_{i=1}^{N} D_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} D_{ij} + \ldots + D_{12\cdots N}. \quad (12)$$

The sensitivity indices are defined as the following ratios

$$S_{i_1 \cdots i_s} = \frac{D_{i_1 \cdots i_s}}{D}, \quad (13)$$

where

$$S_i = \frac{1}{D} \int_0^1 f_i^2(x_i) dx_i \quad (14)$$

are the first-order sensitivities computed for the variables x_i, $i = 1, \ldots, N$; the sensitivities S_i measure how particular x_i variables affect the output of the model, i.e., the variance of $f(\mathbf{x})$. Similarly, the second-order sensitivity

$$S_{ij} = \frac{1}{D} \int_0^1 \int_0^1 f_{ij}^2(x_i, x_j) dx_i dx_j \quad (15)$$

is used to determine the second-order contribution from interaction between x_i and x_j to the output variance. The sum of all sensitivity indices for x_i defined as

$$S_{T_i} = S_i + \sum_{j \neq i} S_{ij} + \ldots + S_{12\ldots N} \tag{16}$$

measures the overall effect of this parameter on the output of the model. All $S_{i_1 \cdots i_s}$ indices are nonnegative and their sum is equal

$$\sum_{s=1}^{N} \sum_{i_1 < i_2 < \cdots < i_s}^{N} S_{i_1 \cdots i_s} = 1. \tag{17}$$

A Monte Carlo algorithm is used for an estimation of global sensitivity indices.

III. Probabilistic neural network

PNN is a feedforward network initially proposed by Specht in [16], [17]. It is very popular with the scientists in the field of machine learning. PNN is frequently utilized in many applications, e.g.: medical diagnosis and prediction [18], [19], [20], [21], image classification and recognition [22], [23], [24], multiple partial discharge sources classification [25], interval information processing [26], [27], phoneme recognition [28], email security enhancement [29], intrusion detection systems [30] or classification in a time-varying environment [31].

The operation of PNN is based on a Bayes decision rule. In this section, we shortly highlight the structure of the model and its training algorithm.

A. Structure of the network

PNN is organized into four layers. The input vector variables $\mathbf{x} = [x_1, \ldots, x_N]$ form the neurons in the first input layer. All given training data, after some activation, are used to create the neurons in the second layer, called the pattern layer. Pattern neurons forward produced output to the next summation layer, where each summation neuron acquires inputs from the pattern neurons representing the same class. In particular, in the summation layer, there exist $g = 1, \ldots, G$ neurons and each gth neuron sums the signals from the neurons of the gth class. The last output layer yields the classification outcome on the basis of the highest value obtained from all G summation neurons.

Different approaches may be utilized to activate pattern neurons of PNN. In this paper, the product kernel involving all input variables is considered

$$K(\mathbf{x}) = \mathcal{K}(x_1) \cdot \mathcal{K}(x_2) \cdot \ldots \cdot \mathcal{K}(x_N), \tag{18}$$

where each multiplicand takes the following Cauchy form

$$\mathcal{K}(x_i) = \frac{2}{\pi(x_i^2 + 1)^2}. \tag{19}$$

Such a form of kernel function allows us to define summation neuron output as follows

$$f_g(\mathbf{x}) = \frac{1}{P_g \det(\mathbf{h})} \sum_{p=1}^{P_g} \frac{1}{s_p^N} K\left(\frac{\left(\mathbf{x} - \mathbf{x}_g^{(p)}\right)^T \mathbf{h}^{-1}}{s_p} \right), \tag{20}$$

where: P_g stands for the number of cases in the gth class ($g = 1, \ldots, G$); $\mathbf{h} = \mathrm{diag}(h_1, \ldots, h_N)$ denotes the vector of smoothing parameters; s_p is the modification coefficient; $\mathbf{x}_g^{(p)} = [x_{g,1}^{(p)}, \ldots, x_{g,N}^{(p)}]$ is the pth training vector of the gth class. The formula (20) is also referred to as the kernel density estimator (KDE) for the gth class in the context of PNN operation.

Using (18) and (19), the gth summation layer neuron produces the following signal

$$f_g(\mathbf{x}) = \frac{1}{P_g \det(\mathbf{h})} \sum_{p=1}^{P_g} \frac{1}{s_p^N} \prod_{i=1}^{N} \frac{2}{\pi\left(\left(\frac{x_i - x_{g,i}^{(p)}}{h_i s_p} \right)^2 + 1 \right)^2}. \tag{21}$$

The final output layer of PNN determines the class assignment for the sample vector $\mathbf{x}$ based on the Bayes decision rule [17] for all f_g KDEs

$$G(\mathbf{x}) = \underset{g=1\ldots G}{\mathrm{argmax}}\ f_g(\mathbf{x}), \tag{22}$$

where $G(\mathbf{x})$ provides the predicted class label. The structure of the PNN model is illustrated in Fig. 1.

B. Training algorithm

The training algorithm of PNN consists in the appropriate choice of the smoothing parameter h_i and the computation of the modification coefficients.

For N-dimensional data sets, when the product kernel is used for KDE estimation, one recommends to compute h_i by means of the plug-in method [32], [33]. The h_i parameters are then determined independently for each dimension

$$h = \left[\frac{R(\mathcal{K})}{U(\mathcal{K})^2} \frac{8\sqrt{\pi}\hat{\sigma}^9}{3P} \right]^{\frac{1}{5}} \tag{23}$$

where $\hat{\sigma}$ denotes the estimator of the standard deviation and for the Cauchy kernel in (19), $R(K) = 1$ and $U(K) = 5/4$. The calculation of $\hat{\sigma}$ is solved iteratively using second-order level approximation [34], [33].

As presented in both (20) and (21), KDE for the gth class depends on the value of the modification coefficient s_p. For PNN, it is computed separately for each class and is related to the pth training vector. The modification coefficient is defined as follows [34]

$$s_p = \left(\frac{\hat{f}(\mathbf{x}^{(p)})}{\tilde{s}} \right)^{-c}, \tag{24}$$

where

$$\tilde{s} = \left(\prod_{p=1}^{P} \hat{f}(\mathbf{x}^{(p)}) \right)^{\frac{1}{P}}, \tag{25}$$

where c is the non-negative constant used to determine the modification intensity. In literature, one usually assumes $c = 0.5$ [33].

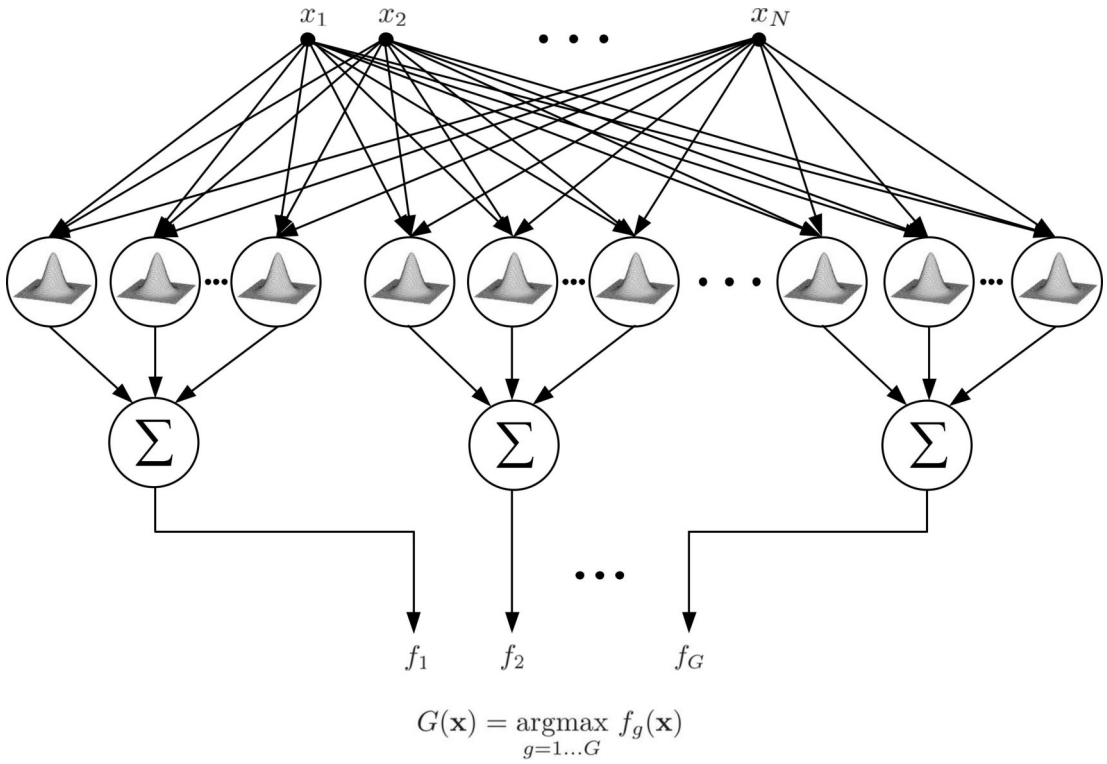

Fig. 1. The architecture of probabilistic neural network.

IV. Algorithm for Determining Significance of Features

This section describes the proposed algorithm for determining the significance of particular features in data set, which in turn, entails the reduction of the PNN's input layer. All components of the this algorithm are set out in Fig. 2 in form of the flowchart. As it can be observed, the flowchart is divided into two parts. The upper part (over the dashed line) concerns a description of PNN topology with all stages of learning process. The bottom part (under the dashed line) shows the application of Sobol method for providing a sensitivity indices what results in establishing the order of data features.

In the first stage of the algorithm, we start from data acquisition ❶. Since the PNN model is utilized, it is assumed (step ❷) that data are distinguished between particular classes. In step ❸, the topology of PNN is created. For this purpose, the number of records, features and classes of the considered data are acquired. Then all training patterns are copied into appropriate neurons (stage ❹) preserving class membership, as it is shown in Fig. 1. This results in obtaining the required structure of PNN ready for training process. Now, as it is presented in subsection III-B, in step ❺, the smoothing parameters h_i are computed for each of regarded classes separately. As a result N smoothing parameters are obtained in each class (which gives $N \cdot G$ in total). In step ❻ of the algorithm, for every gth class, the modification coefficients s_p, $p = 1, \ldots, P_g$, are determined.

In the second stage of the algorithm, the global sensitivity analysis takes place (❼). The application of Sobol method allows us to obtain required information about influence of individual elements of the input vector on particular KDEs $f_g(\mathbf{x})$. Based on the Sobol approach described in Section II, for each input element x_i and each class estimator $f_g(\mathbf{x})$, the first order sensitivity index $S_{i,g}^{(p)}$ (14) for the pth training pattern is computed. After determination of $S_{i,g}^{(p)}$ for all P training patterns, one can calculate aggregated parameters by applying mean square average sensitivity norm

$$S_{i,g}^{\text{mean}} = \sqrt{\frac{\sum_{p=1}^{P} \left(S_{i,g}^{(p)}\right)^2}{P}}. \tag{26}$$

Finally, it is required to define the maximum value S_i in ith row of the matrix $\mathbf{S}^{\text{mean}}$ with the elements aggregated according to (26)

$$S_i = \max_{g=1,\ldots,G} \left\{ S_{i,g}^{\text{mean}} \right\}. \tag{27}$$

In the last step ❽, the algorithm returns the sorted vector with S_i coefficients and the vector which contains the indices corresponding to the sorted coordinates. The first algorithm output item informs us about the aggregated quantitative sensitivity of individual inputs in the PNN's class estimator. These inputs are associated with the features of the considered data set. The second algorithm output item gives us the possibility to indicate the order of features' significance.

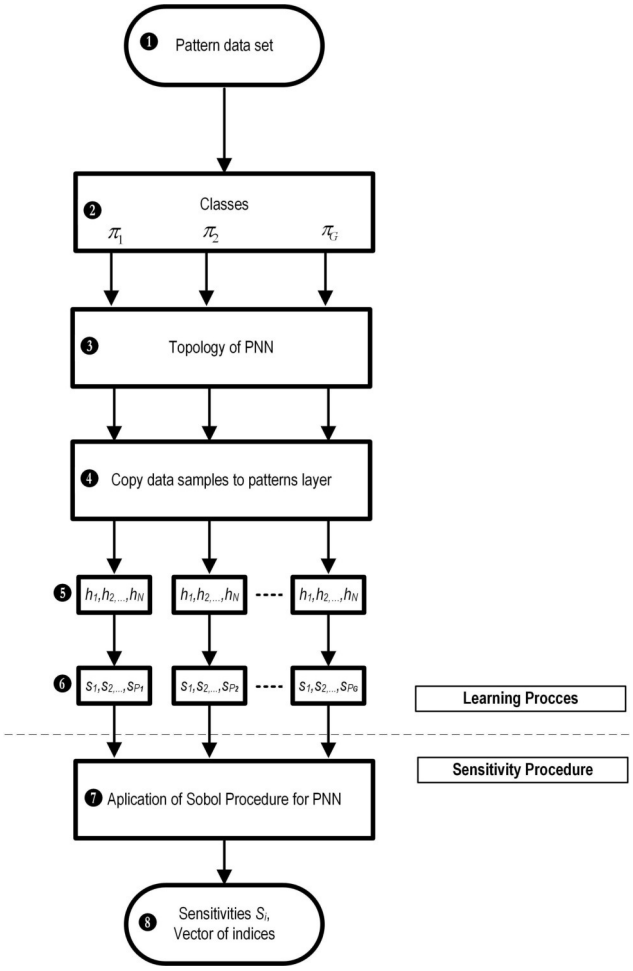

Fig. 2. Flowchart of the proposed algorithm.

The steps ❼ and ❽ as well as the PNN learning stages ❶–❻ constitute the complete algorithm for determining the significance of particular features of data set.

In the current paper, we do not focus on providing a priori general criterion to decide what is the right number of features to remove. Such a criterion could, for example, be based on finding explicit difference between two neighboring S_i elements in the matrix $\mathbf{S}^{\text{mean}}$, as shown in [9]. On the other hand, determining a general threshold of feature significance is difficult to establish since it is dependent on classifier applications. However, if we assume the use of PNN in classification tasks, some solution could rely on iterative reduction of the least significant feature along with simultaneous assessment of the network quality.

V. Numerical results

In this section, numerical verification results of the proposed algorithm are presented. In the first part, we focus on Sobol sensitivity method applied to determine the significance of input features. The second part considers the evaluation of the introduced algorithm in the classification tasks. To make our study more representative, three UCI–MLR medical data sets are taken under consideration. Table I characterizes these data sets. In particular, we present: the number of records with class distribution (M_i), the number of features (N), and the number of classes (C). In the last column of the table, the bibliography reference of each data set is provided.

TABLE I
Characteristics of experimental data sets

Data set	Abbrev.	M_i	N	C	Biblio.
Wisconsin Breast Cancer	WBC	239 – 444	9	2	[35]
Statlog Heart	SH	150 – 120	13	2	[36]
Parkinsons Data	PD	48 – 147	22	2	[37]

A. Significance of data features

This part of paper examines the application of the Sobol method used to determine the significance of the individual features for all data sets presented in Table I. The results of the numerical verification of the algorithm presented in Section IV are shown in three drawings for each data set separately. In particular, for the WBC data set, Fig. 3 contains the sensitivity values S_i for each data feature, Fig. 4 displays the sorted values of S_i in descending order while Fig. 5 illustrates the difference between the particular bins presented in Fig. 4, i.e. $dS_i = S_{i-1} - S_i$ for $i = 2, 3, \ldots, N$. Figures 6, 7, and 8 depict respectively: S_i, sorted S_i and dS_i for the SH data set. Finally, in Fig. 9, Fig. 10, and Fig. 11, we show S_i, sorted S_i and dS_i for the PD data set, respectively.

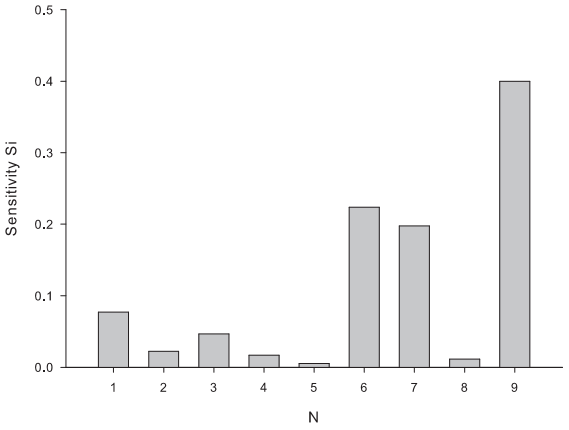

Fig. 3. Sensitivity coefficients for the WBC data set.

In the case of the WBC data set (Fig. 3 and Fig. 4), we can see that the 9th feature is the most dominating since its sensitivity is equal $S_9 = 0.3998$. Then, two features can be distinguished, i.e., x_6 and x_7 for which $S_i \approx 0.2$. The next distinctive group of features constitute x_1 and x_3 where $S_i \in (0.05, 0.1)$. The remaining features, i.e., {2,4,8,5}

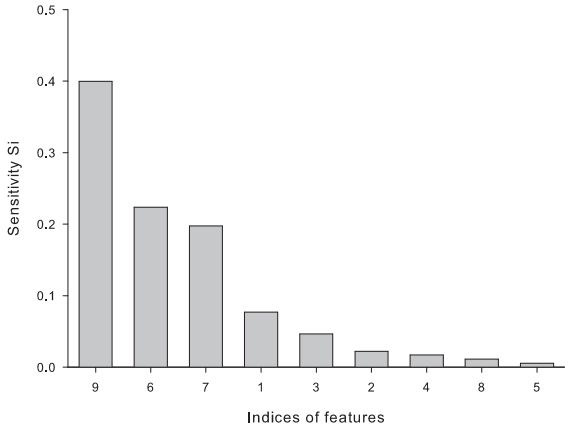

Fig. 4. Sorted sensitivity coefficients for the WBC data set.

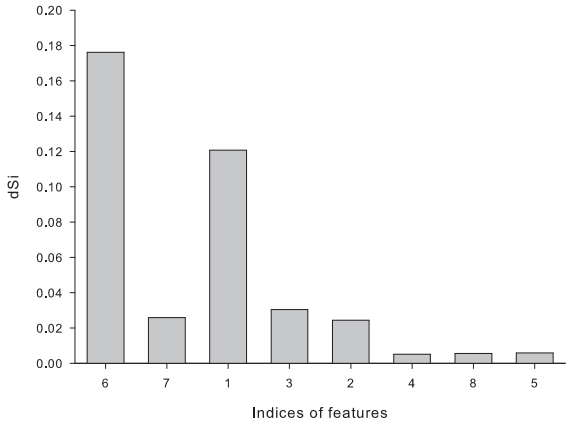

Fig. 5. The differences between sorted sensitivity coefficients for the WBC data set.

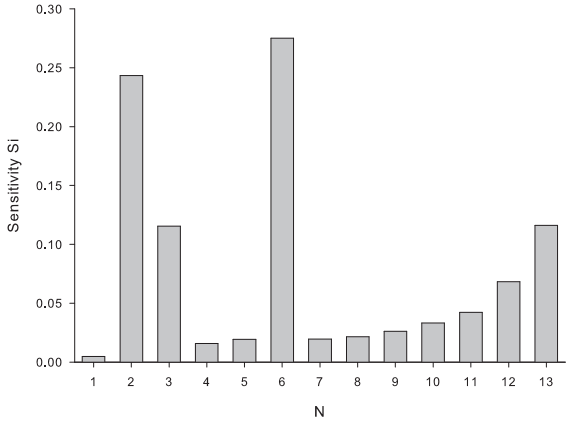

Fig. 6. Sensitivity coefficients for the SH data set.

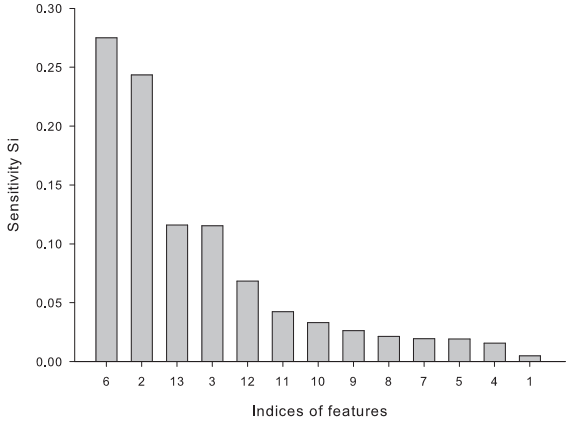

Fig. 7. Sorted sensitivity coefficients for the SH data set.

comprise the collection of less significant inputs because all their sensitivities are less than 0.025. In Fig. 5, one can clearly notice two distinct values for the first and third bar. This indicates the border between the most important feature x_9, two elements group of x_6, x_7, and further features x_1 and x_3. These visible bars may contribute to discovering the cutoff between significant and negligible features for this data set. This fact will be explained in subsection V-B.

In the case of the SH data set (Fig. 6), it is possible to point out two significant features x_6 and x_2 for which S_i equals 0.2751 and 0.2434, respectively. The next group of features create the inputs x_{13} and x_3 with $S_i \approx 0.1150$. Then for the features $\{x_{12}, x_{11}, x_{10}\}$ (what can be observed in Fig. 7), we can remark linear decline of the sensitivity. The remaining features are characterized by a similar value of $S_i \approx 0.02$. Only the last feature x_1 is the least significant what results from $S_1 = 0.0047$. Analyzing Fig. 8, one can see a noticeable peak at 13th feature and much smaller one at x_{12}. These observations indicate two potential borders where the input reduction may occur.

Finally, for the last PD data set considered in this study, which consists of 22 features, one observes that the most important feature index is 10; here $S_{10} = 0.1602$ (see Fig. 9 and 10). Subsequent group of features is characterized by $S_i \approx 0.1$ which includes inputs $\{x_{13}, x_6, x_8, x_{12}\}$. Analyzing the indices of features from the set $\{1, 5, 4, 15\}$ we can see a linear decrease in the sensitivity coefficient values. The next two peaks in the figure belong to features 11 and 14 with similar sensitivity (approx. 0.028). The last group of features comprises the ones for which $S_i < 0.02$.

B. Verification of data features significance in classification task

The results presented in subsection V-A are verified in the classification problems. Firstly, we apply Sobol method globally on the entire data set and determine the order of features' significance. Sorted sensitivity coefficients for the considered WBC, SH and PD data sets are presented in Figures 4, 7 and 10, respectively. Then, the PNN classification

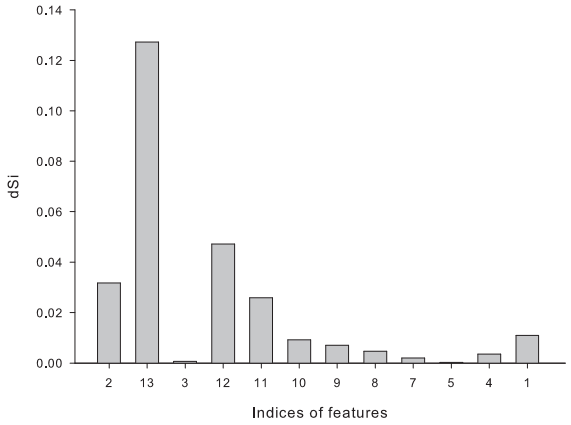

Fig. 8. The differences between sorted sensitivity coefficients for the SH data set.

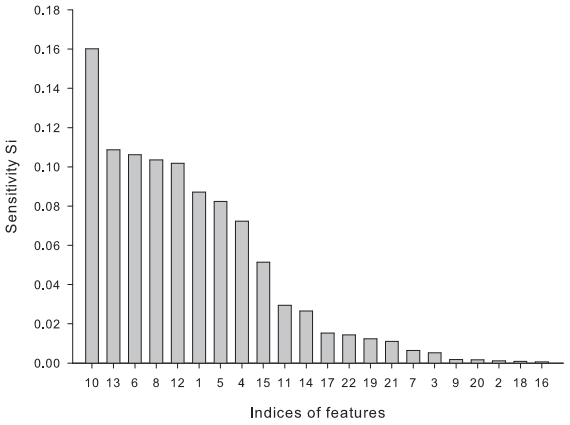

Fig. 10. Sorted sensitivity coefficients for the PD data set.

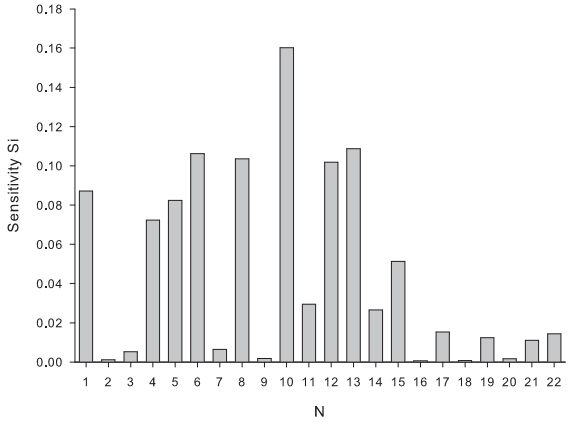

Fig. 9. Sensitivity coefficients for the PD data set.

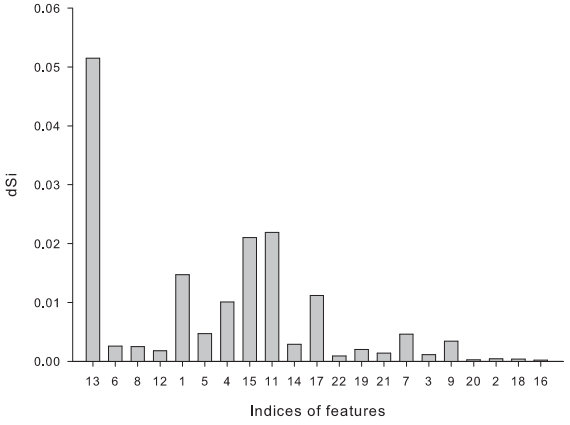

Fig. 11. The differences between sorted sensitivity coefficients for the PD data set.

performance is evaluated using a 10-fold cross validation (CV) procedure. Single classification task is performed by removing the least significant feature. The entire procedure is conducted until a single feature is left. The whole experiment is repeated 30 times. As the result, we provide classification quality computed as the ratio of the number of correctly classified input patterns to the data set cardinality.

For all analyzed data sets, the obtained results are set out in tables and figures. The tables present the following indicators: the current number of features (N), the least significant feature index (LSF), average learning quality along with standard deviation–denoted as q_{cv}^L and $std(q_{\text{cv}}^L)$, and average test quality with standard deviation–denoted as q_{cv}^T and $std(q_{\text{cv}}^T)$. In the case of figures, the plotted bars depict q_{cv}^L (painted gray) and q_{cv}^T (painted white) determined at particular set of selected features.

Table II and Fig. 12 represent the results for the WBC data set. Analyzing the reduction of individual features, the following is observed. First of all, the inequality $q_{\text{cv}}^L > q_{\text{cv}}^T$ holds in the entire range of feature indices. The sensitivity to the reduction in the test set is higher than the one in the learning set. Secondly, by reducing the least significant feature (no. 5) we notice an improvement in the quality of the classification for the test set. However, within the removal of the next least significant features (i.e., 8,4 and 2), a slight quality decrease is noticed: q_{cv}^L drops from 0.9987 (for full data set) down to 0.9946 (data set with 6 features) while q_{cv}^T – from 0.9677 down to 0.9458. Let us proceed further: by removing x_3 and x_1, we achieve the decrease of the test quality to 0.9311. Now the tendency in quality decrease becomes stronger and stronger since discarding the next two features (7 and 6) results in a sudden q_{cv}^T decline (0.8912). Finally, leaving only the most significant 9th feature causes a drastic worsening of the test quality (down to 0.7861). The above conclusions strongly refer to the groups of features with similar sensitivity values.

For the SH data set, the results are presented in Table III and in Fig. 13. Here, the effect of simultaneous features' reduction

TABLE II
SIMULATION RESULTS FOR WBC DATA SET

N	LSF	q_{cv}^L	$std(q_L)$	q_{cv}^T	$std(q_T)$
9	5	0.9987	0.0001	0.9677	0.0023
8	8	0.9973	0.0001	0.9697	0.0019
7	4	0.9972	0.0002	0.9589	0.0024
6	2	0.9946	0.0001	0.9458	0.0032
5	3	0.9861	0.0002	0.9421	0.0030
4	1	0.9691	0.0005	0.9311	0.0026
3	7	0.9245	0.0004	0.8918	0.0019
2	6	0.9079	0.0008	0.8912	0.0046
1	9	0.7876	0.0008	0.7861	0.0015

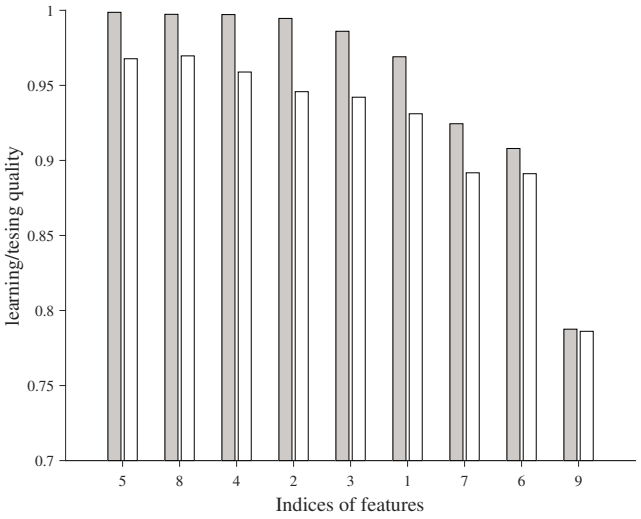

Fig. 12. Simulation results for WBC data set.

TABLE III
SIMULATION RESULTS FOR SH DATA SET

N	LSF	q_{cv}^L	$std(q_L)$	q_{cv}^T	$std(q_T)$
13	1	1.0000	0.0000	0.7781	0.0083
12	4	1.0000	0.0000	0.7859	0.0087
11	5	1.0000	0.0000	0.7819	0.0077
10	7	1.0000	0.0000	0.7478	0.0083
9	8	1.0000	0.0000	0.7626	0.0081
8	9	0.9967	0.0002	0.6763	0.0091
7	10	0.9968	0.0004	0.6726	0.0127
6	11	0.9966	0.0002	0.6419	0.0100
5	12	1.0000	0.0000	0.6041	0.0083
4	3	0.9801	0.0012	0.5537	0.0126
3	13	0.9460	0.0012	0.5807	0.0164
2	2	0.7503	0.0021	0.5500	0.0195
1	6	0.6429	0.0012	0.6204	0.0064

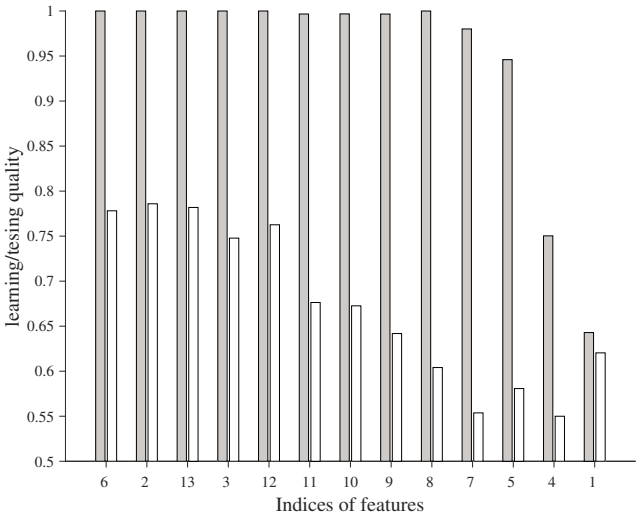

Fig. 13. Simulation results for SH data set.

and the increase of the classification quality can be discerned. This phenomenon occurs when first two features (i.e., 6 and 2) are deleted from data set. The test quality grows form 0.7781 for original data set up to 0.7819 for the reduced one. The removal of four least significant features leads to 0.0155 decrease of q_{cv}^T index. The rejection of the subsequent features results in a significant deterioration in the data set representativeness, therefore the obtained outcomes keep on worsening. The smallest value of test quality is obtained for $N = 2$. However, for the data set with the single feature ($N = 1$), the value of q_{cv}^T is over 7% higher than the one determined when $N = 2$.

Finally, Table IV and Fig. 14 present the results achieved for the PD data set. As shown, discarding ten least significant features yields a slight fluctuation in classification outcomes, since the overall level of quality varies by about 2% here. The reduction of 11 features makes q_{cv}^T decrease below 0.84. The removal of seven subsequent features results in q_{cv}^T changes in the range of 0.87 to 0.83. Discarding 17 least significant features results in a substantial drop in test quality down to a value of 0.7197. At last, when we get rid of $N - 1$ features, the worst outcome is provided, i.e., $q_{\text{cv}} = 0.7015$.

VI. SUMMARY

In this work, the complete algorithm for determining the significance of input features in medical data sets was proposed. It was based on the definition of the global sensitivity indices generated according to the Sobol method. The correctness of the algorithm was verified on the UCI-MLR data classification tasks using the PNN model by computing learning and testing qualities. We showed that it was possible to obtain higher classification performance of PNN after removal of the least significant features. According to medical feedback, the proposed algorithm exhibited proper functioning. Based on the numerical verification, the algorithm had advantageous properties in high-dimensional case ($N = 22$) since no increase in data set cardinality was required to achieve satisfactory

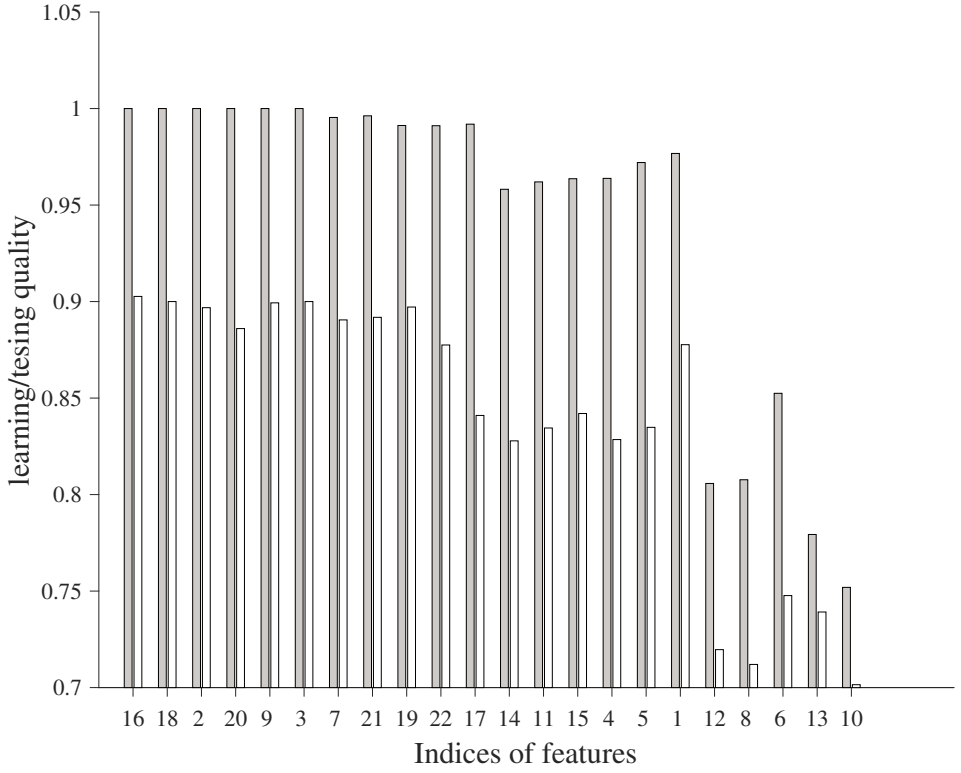

Fig. 14. Simulation results for PD data set.

TABLE IV
SIMULATION RESULTS FOR PD DATA SET

N	LSF	q_{cv}^{L}	$std(q_L)$	q_{cv}^{T}	$std(q_T)$
22	16	1.0000	0.0000	0.9027	0.0046
21	18	1.0000	0.0000	0.9000	0.0065
20	2	1.0000	0.0000	0.8968	0.0090
19	20	1.0000	0.0000	0.8860	0.0069
18	9	1.0000	0.0000	0.8993	0.0082
17	3	1.0000	0.0000	0.9000	0.0050
16	7	0.9954	0.0003	0.8905	0.0075
15	21	0.9962	0.0005	0.8918	0.0075
14	19	0.9912	0.0007	0.8972	0.0084
13	22	0.9911	0.0005	0.8775	0.0093
12	17	0.9919	0.0006	0.8410	0.0092
11	14	0.9582	0.0006	0.8278	0.0069
10	11	0.9620	0.0011	0.8345	0.0069
9	15	0.9636	0.0009	0.8420	0.0112
8	4	0.9638	0.0013	0.8285	0.0091
7	5	0.9720	0.0008	0.8348	0.0110
6	1	0.9767	0.0008	0.8777	0.0062
5	12	0.8058	0.0023	0.7197	0.0105
4	8	0.8077	0.0019	0.7120	0.0121
3	6	0.8525	0.0019	0.7477	0.0110
2	13	0.7793	0.0039	0.7392	0.0176
1	10	0.7519	0.0032	0.7015	0.0118

results. This, in turn, saved us from well known "curse of dimensionality".

The future work will focus on application and simplification of the proposed algorithm on high-dimensional data set classification problems. Other global sensitivity methods will also be considered.

REFERENCES

[1] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.

[2] I. M. Sobol, "Sensitivity estimates for nonlinear mathematical models," *Mathematical Modelling and Computational Experiments*, vol. 1, no. 4, pp. 407–414, 1993.

[3] ——, "Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates," *Mathematics and computers in simulation*, vol. 55, no. 1, pp. 271–280, 2001.

[4] R. Cukier, C. Fortuin, K. E. Shuler, A. Petschek, and J. Schaibly, "Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. i theory," *The Journal of Chemical Physics*, vol. 59, no. 8, pp. 3873–3878, 1973.

[5] A. Saltelli, S. Tarantola, and K.-S. Chan, "A quantitative model-independent method for global sensitivity analysis of model output," *Technometrics*, vol. 41, no. 1, pp. 39–56, 1999.

[6] M. Fesanghary, E. Damangir, and I. Soleimani, "Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm," *Applied Thermal Engineering*, vol. 29, no. 5, pp. 1026–1031, 2009.

[7] E. Fock, "Global sensitivity analysis approach for input selection and system identification purposes–a new framework for feedforward neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1484–1495, 2014.

[8] A. Cosenza, G. Mannina, P. A. Vanrolleghem, and M. B. Neumann, "Global sensitivity analysis in wastewater applications: A comprehensive comparison of different methods," *Environmental modelling & software*, vol. 49, pp. 40–52, 2013.
[9] P. A. Kowalski and M. Kusy, "Sensitivity analysis for probabilistic neural network structure reduction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–14, 2017. doi: 10.1109/TNNLS.2017.2688482
[10] I. Kononenko, "Estimating attributes: analysis and extensions of relief," in *Machine Learning: ECML-94*. Springer, 1994, pp. 171–182.
[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
[12] M. Szczuka and D. Slezak, "Feedforward neural networks for compound signals," *Theoretical Computer Science*, vol. 412, no. 42, pp. 5960–5973, 2011.
[13] A. Janusz and D. Slezak, "Utilization of attribute clustering methods for scalable computation of reducts from high-dimensional data," in *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2012, pp. 295–302.
[14] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[15] A. Saltelli, S. Tarantola, and K.-S. Chan, "A quantitative model-independent method for global sensitivity analysis of model output," *Technometrics*, vol. 41, no. 1, pp. 39–56, 1999.
[16] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.
[17] ——, "Probabilistic neural networks and the polynomial adaline as complementary techniques for classification," *Neural Networks, IEEE Transactions on*, vol. 1, no. 1, pp. 111–121, Mar 1990. doi: 10.1109/72.80210
[18] R. Folland, E. Hines, R. Dutta, P. Boilot, and D. Morgan, "Comparison of neural network predictors in the classification of tracheal–bronchial breath sounds by respiratory auscultation," *Artificial intelligence in medicine*, vol. 31, no. 3, pp. 211–220, 2004.
[19] D. Mantzaris, G. Anastassopoulos, and A. Adamopoulos, "Genetic algorithm pruning of probabilistic neural networks in medical disease estimation," *Neural Networks*, vol. 24, no. 8, pp. 831–835, 2011.
[20] M. Kusy and R. Zajdel, "Application of reinforcement learning algorithms for the adaptive computation of the smoothing parameter for probabilistic neural network," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 9, pp. 2163–2175, 2015.
[21] ——, "Probabilistic neural network training procedure based on q(0)–learning algorithm in medical data classification," *Applied Intelligence*, vol. 41, no. 3, pp. 837–854, 2014.
[22] 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, no. 11, pp. 3015–3023, 1998.
[23] 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, no. 4, pp. 266–275, 2007.
[24] X.-B. Wen, H. Zhang, X.-Q. Xu, and J.-J. Quan, "A new watermarking approach based on probabilistic neural network in wavelet domain," *Soft Computing*, vol. 13, no. 4, pp. 355–360, 2009.
[25] S. Venkatesh and S. Gopal, "Orthogonal least square center selection technique–a robust scheme for multiple source partial discharge pattern recognition using radial basis probabilistic neural network," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8978–8989, 2011.
[26] P. A. Kowalski and P. Kulczycki, "Data sample reduction for classification of interval information using neural network sensitivity analysis," in *Artificial Intelligence: Methodology, Systems, and Applications*, ser. Lecture Notes in Computer Science, D. Dicheva and D. Dochev, Eds. Springer Berlin Heidelberg, 2010, vol. 6304, pp. 271–272.
[27] ——, "Interval probabilistic neural network," *Neural Computing and Applications*, vol. 28, no. 4, pp. 817–834, 2017. doi: 10.1007/s00521-015-2109-3. [Online]. Available: http://dx.doi.org/10.1007/s00521-015-2109-3
[28] K. Elenius and H. G. Tråvén, "Multi-layer perceptrons and probabilistic neural networks for phoneme recognition." in *EUROSPEECH*, 1993.
[29] T. P. Tran, T. T. S. Nguyen, P. Tsai, and X. Kong, "Bspnn: boosted subspace probabilistic neural network for email security," *Artificial Intelligence Review*, vol. 35, no. 4, pp. 369–382, 2011.
[30] T. P. Tran, L. Cao, D. Tran, and C. D. Nguyen, "Novel intrusion detection using probabilistic neural network and adaptive boosting," *International Journal of Computer Science and Information Security*, vol. 6, no. 1, pp. 83–91, 2009.
[31] L. Rutkowski, "Adaptive probabilistic neural networks for pattern classification in time-varying environment," *Neural Networks, IEEE Transactions on*, vol. 15, no. 4, pp. 811–827, July 2004.
[32] P. A. Kowalski and P. Kulczycki, "A complete algorithm for the reduction of pattern data in the classification of interval information," *International Journal of Computational Methods*, vol. 13, no. 03, p. 1650018, 2016. doi: 10.1142/S0219876216500183
[33] M. P. Wand and M. C. Jones, *Kernel smoothing*. Crc Press, 1994.
[34] B. W. Silverman, *Density estimation for statistics and data analysis*. CRC press, 1986, vol. 26.
[35] J. Zhang, "Selecting typical instances in instance-based learning," in *Proceedings of the Ninth International Workshop on Machine Learning*, ser. ML92. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992. ISBN 1-5586-247-X pp. 470–479. [Online]. Available: http://dl.acm.org/citation.cfm?id=141975.142091
[36] G. Brown, *Diversity in neural network ensembles*. University of Birmingham, 2004.
[37] 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, April 2009. doi: 10.1109/TBME.2008.2005954