EASR

Engineering and Applied Science Research

https://www.tci-thaijo.org/index.php/easr/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Optimal data division for empowering artificial neural network models employing a modified M-SPXY algorithm

Wirote Apinantanakon*[1)], Khamron Sunat[1)] and Joel Alan Kinmond[2)]

[1)]Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand
[2)]27 Patrick Street, Trenton, Ontario k8v 4B, Canada

Received 21 April 2019
Revised 15 June 2019
Accepted 18 June 2019

Abstract

Data splitting is an important step in artificial neural network (ANN) models, which is found in the form of training and testing subsets. In general, a random data splitting method is favored to divide a pool of samples into subsets, without considering the quality of data for the training step of a neural network. The drawback of poor data splitting methods is that they poses ill effects to the performance of the neural network when the data involves complex matrices or multivariate modeling. In order to overcome this drawback, the current paper presents our proposed M-SPXY method. It is based on a modified version of Sample Set Partitioning, which relies on a joint X-y distances (SPXY) method. The proposed method has resulted in better performance, compared to the modified Kennard-Stone (KS) method, using Mahalanobis distances (MDKS). In our experiments, the proposed approach was employed to compare various data splitting methods using data sets from the repository of the University of California in Irvine (UCI), processed through an Extreme Learning Machine (ELM) neural network. Performance was measured in terms of classification accuracy. The results indicate that the classification accuracy of the proposed M-SPXY process is superior to that of the MDKS data splitting method.

Keywords: Extreme learning machine, SPXY, Neural network, Subset selection, Mahalanobis distance, Classification

1. Introduction

In the development of Artificial Neural Network (ANN) models, one major requirement is to obtain high performance in terms of precision accuracy in predictions, especially when applied to unseen data. Accuracy is evaluated using ANN models with training and testing data and comparing their performance [1]. In general, accuracy is achieved by inputting available subsets of training data into network models. This is done to learn, and eventually use the subset of testing data to measure the capability of the models. ANN models do not provide an exact method with which to subset the data. However, they do have significant impact on the ability of a model to ensure good generalization [2]. Models are developed and trained to predict outputs of unknown target functions, which represents a finite subset of input data, and the corresponding required outputs. After completion of the learning step, ANN models have the ability to correctly predict outputs from other available input data and to generalize unseen data. However, some problems occur during the process, as ANN models may simply memorize the training examples. In this case, they are not able to provide correct outputs for patterns that are not in the training data sets, leading to poor generalization. This problem is usually found in the structure or parameters of network models, in which improper data splitting has occurred, commonly known as over-training or over-fitting the data [3]. According to many published research reports, the performance of ANN models depends on their design and implementation. However, the performance of ANN models may also depend the quality of input data [4]. A common technique used to address the difficulties of representing data is the use of appropriate although sophisticated sampling methods [5-6] that solve the problems through proper data splitting. Therefore, it is important to find an appropriate data splitting technique, which supports ANN models in a way that can be generalized.

In the area of data splitting, several studies have proposed techniques which select a representative subset from a large pool of data. Wu et al. [7] used the random subsets selection method (RS) for dividing available data into representative training subsets. The random selection approach provides subset training data through random division, without considering the quality of the data. The drawback of RS is that it impacts classification accuracy when ANN models encounter complex data. Another data splitting method is referred to as cross-validation [8]. This is a conventional method that avoids over training and obtains a confidence level for generalization. The basic premise of cross-validation is to divide the dataset into two subsets: a training subset and a performance evaluation subset. Whereas, the widely used *hold-out cross-validation*

*Corresponding author. Tel.: +6681 977 2516
Email address: wirotta@gmail.com
doi: 10.14456/easr.2019.31

method [9] divides all available datasets into three disjoint subsets. One set is used for training. Another set is used to avoid over-training by evaluating the model during training. The last set is used to gain a confidence estimation of the model's performance. Moreover, the advantage of *hold-out cross-validation* is that the proportion of data in each of the three subsets can be varied. However, the question at hand becomes how to separate the data into the three subsets.

In the past decade, Kennard et al. [10] proposed a data splitting technique a called the Kennard-Stone (KS) algorithm. The KS algorithm aims at covering the multidimensional space using Euclidean measurements to maximize the distances between the instrumental response vectors (*X*) of the selected samples. Although the KS algorithm may be used with alternative partitioning methods, a problem exists within the multivariate calibration context. This lies in the fact that the statistics of the dependent variable (*y*) are not taken into account, thereby affecting the performance of the KS algorithm. An improved version of KS, developed by Galvão et al., [11], proposed a method for calibration and validation partitioning, known as the SPXY (Sample Set Partitioning based on joint *X*-*y* distances). The presentation of SPXY considers the variability of both *X* and *y* dimensions. The assumption of SPXY is that the inclusion of *y*-information directly influences the selection process, resulting in a more effective distribution of calibration samples within a multidimensional space. The Kennard-Stone Algorithm was further developed as the Mahalanobis Distance (MDKS) model [12], proposed in 2012. It has been shown superior to both the KS and SPXY methods. MDKS was extended from the KS Algorithm by modifying the KS distances, replacing Euclidean distance (ED) with the Mahalanobis Distance (MD) [13]. The proposed MDKS can eventually establish adequate training subsets, as well as enhance the performance of the ANN model development process. It is especially effective in detecting outliers [14].

At present, the SPXY method is still favored in many real world applications [15-16]. In this paper, we propose an M-SPXY method for data splitting, to select appropriate training subsets for ANN models through the deployment of an Extreme Learning Machine. The M-SPXY method originates from the SPXY method. Both have variability in the *X* and *y* spaces of sample data and measure of the Mahalanobis distances is suitable. The results of the experiments were derived from a comparison of the average accuracies of M-SPXY and MDKS methods in testing processes.

The remainder of this paper is organized as follows. The materials and methods are introduced in Section 2. The proposed M-SPXY algorithm is presented in Section 3. Section 4 discusses the extreme learning machine. Section 5 describes the experimental design of the study. Section 6 presents the results. Section 7 gives a discussion of the results. The conclusions of the study are drawn in Section 8 of the paper.

2. Materials and methods

2.1 Mahalanobis distance and the selection of calibration samples set

Mahalanobis (MD) is a well-known distance measurement technique applied for various purposes, such as clustering techniques [17] as well as linear, quadratic and regularized discriminating analysis [18]. In multivariate calibration, MD is used for selection of calibration samples from a large set of measurements that are representative of the data [19-20], as well as sample selection and calibration procedures for near-infrared reflectance spectroscopy [21]. MD is a measure between two data points in a space according to a relevant variable. It also correlates features. In the field of multivariate calibration, the MD takes into account correlations in the dataset and calculates the similarity of each of the data points using variance and covariance matrices.

The calibration model is a very useful technique for extracting chemical information [22-24]. Various calibration models, particularly multivariate calibrations, have been applied to many analytical measurements for spectroscopic signals, near-infrared (NIR) data, fluorescence spectroscopy and electrochemistry. The purpose of a calibration model is to provide an appropriate sample set to be predicted what will be measured under a different environmental condition. A calibration model is used when the sample needs to predict values evaluated under a different environment and factors.

In this paper, the selection of an appropriate subset of instances to be included in a training data set is a very important preprocessing step, especially in classification problems. It can improve the quality of the output of a neural network model and classification accuracy. The inspiration of the current study is to combine a calibration technique and the Mahalanobis distance to provide an appropriate subset to train an ELM neural network.

2.2 The SRS algorithm

The Simple Random Sampling (SRS) method is the most commonly used algorithm in data splitting, as it is easy to implement. The SRS selects data randomly, with a uniform distribution. For example, a training subset may be written as:

$$S_{tr} : P\left(x \in S_{tr}\right) = \frac{n_{tr}}{n}, n = |\, S\,|, n_{tr} = |\, S_{tr}\,|. \quad (1)$$

Each sample has an equal probability of selection. The advantage of this algorithm is that it reduces the bias within the model's performance. However, for more complex datasets, the SRS suffers from high variance of model performance as the selection of subsets may not properly cover all data.

2.3 The KS algorithm

The Kennard–Stone (KS) algorithm forms a subset of *M* samples. The technique of the KS algorithm is generally considered to be a candidate for forming the training set, in which the selection of each candidate is chosen sequentially. The stepwise procedure of KS can be summarized as follows. The first step is computing the distance between the pair of samples (*p*, *q*) using the Euclidean distance (ED_x of x-vectors). The ED_x is determined by:

$$ED_x(p,q) = \sqrt{\sum_{j=1}^{N} (x_p(j) - x_q(j))^2} \quad p, q \in [1, M] \quad (2)$$

In equation (2), *x* values are variables in the set of *M* samples. *N* is the dimension of variables, and $x_p(j)$ and $x_q(j)$ are the j^{th} dimension for samples *p* and *q*, respectively. Then, the sample with the largest ED_x is selected. At each subsequent iteration, the algorithm selects the sample that provides the least distance with respect to other samples that

have already been selected. Last, the procedure is repeated until the number of required samples is achieved.

2.4 The SPXY algorithm

SPXY (Sample Set Partitioning based on disjoint *X*-*y*) was extended from the Kennard-Stone (KS) method, covering both the independent variable *X*, and dependent variable *y*. The KS algorithm selects a representative training subset from a pool of *M* samples. SPXY augments the distance of KS, defined in equation (3), with a distance in the dependent variable (*y*) space for the parameter under consideration. The SPXY algorithm employs a distance $ED_y(p, q)$ by calculating each pair of p and q samples as:

$$ED_y(p,q) = \sqrt{\sum_{j=1}^{n}(y_p(j) - y_q(j))^2} \quad p, q \in [1, M] \qquad (3)$$

where $y_P(j)$ and $y_q(j)$ are the j^{th} variables for samples p and q, respectively, and j is the number of variables in y spaces. The distances of $ED_x(p,q)$ and $ED_y(p,q)$ are divided by their maximum values to better understand the distribution of samples for variables x and y:

$$ED_{xy}(p,q) = \frac{ED_x(p,q)}{max_{p,q\epsilon[1,M]}ED_x(p,q)} + \frac{ED_y(p,q)}{max_{p,q\epsilon[1,M]}ED_y(p,q)} \qquad (4)$$

2.5 The MDKS algorithm

A modified Kennard-Stone (KS) algorithm for optimal division of data for developing artificial neural network models (MDKS) was proposed by Saptoro et al. [12]. This algorithm is extended from the original KS algorithm, using the Mahalanobis distance method as selection criteria, rather than the Euclidean distance. A stepwise selection is employed that is similar to that of the KS algorithm. However, the results of the MDKS are superior to those of the KS. For the proposed MDKS, the algorithm replaces equation (2) with equation (5) as follows:

$$MD(p,q) = \sqrt{E'(p,q)^{C^{-1}}(p,q)E(p,q)} \qquad (5)$$

where

$$E'(p,q) = [e_{pq}(1)e_{pq}(2) \dots e_{pq}(u) \dots e_{pq}(F)] \qquad (6)$$

and

$$e_{pq}(u) = ((p,u) - z(q,u)); \; p,q \in M \quad u = 1,2,\dots,F \qquad (7)$$

F is the number of variables in the matrix, *z*. $C(p,q)$ is the covariance matrix of E [13], where:

$$C(p,q) = \frac{1}{M}(E'(p,q)E(p,q)) \qquad (8)$$

The results from MDKS studies show that this method outperforms both the original KS and SPXY methods.

3. The proposed M-SPXY algorithm

The proposed M-SPXY algorithm exhibits superior performance in splitting data via the MDKS algorithm, as well as the advantages of the SPXY algorithm. It takes into account the variability in both X and y-spaces. In the proposed method, we split the data as follows:

$$MED_{xy}(p,q) = \frac{MD_x(p,q)}{\max_{p,q\in[1,M]} MD_x(p,q)} + \frac{ED_y(p,q)}{\max_{p,q\in[1,M]} ED_y(p,q)} \qquad (9)$$

where equation (9) is an extension of the previous SPXY algorithm.

The MDKS algorithm is successful when used with the Mahalanobis distance, in which $ED(p,q)$ in equation (4) is replaced with $MD(p,q)$ from equation (5). However, the importance of the dependent variable (y) is also considered in this work, especially since $ED(p,q)$ is not modified.

In evaluating the performance of M-SPXY, an Extreme Learning Machine (ELM) was used to classify the dataset and compared the accuracy of the results. The MDKS and M-SPXY algorithms were compared.

4. Extreme learning machine

An Extreme Learning Machine (ELM) [25] tests the performance of artificial neural network models. This may have significant impact upon ANN models, which are widely used in many disciplines, such as chemistry and engineering [26], prediction data within distributed systems [27], classifications [28], as well as in recognition and businesses [29-30]. Moreover, many studies have proposed an improved version of ELM to optimize model structure [31-32]. The current study was done to show that data splitting may enhance ELM performance, in addition to the performance enhancement obtained through the model's design.

Huang et al. [25] proposed an ELM method for training algorithms for Single-Hidden Layer Feed-Forward Neural networks (SLFN). This ELM demonstrated several interesting advantages when combined with various neural networks. For example, it applied random computational nodes in a hidden layer independently within the training data. Additionally, it did the required parameter tuning of the hidden layer of the SLFN. Furthermore, it computed the output weights using a least-square solution. The output function of SLFN networks is given by:

$$f_L(X) = \sum_{i=1}^{L} \beta_i g_i(X) \qquad (10)$$

where $X \in R^d$, $\beta_i \in R^m$ and the output of the i^{th} hidden node, $G(a_i, b_i, X)$ is given by g_i.

Depending on the node type, the output is given as:

$$g_i = G(a_i, b_i, X) = \begin{cases} g(a_i.X + b_i) \\ af(b_i||-a_i||) \end{cases}$$

with $a_i \in R^d$, $b_i \in R$ for additive nodes
with $a_i \in R^d$, $b_i \in R^+$ for RBF nodes (11)

Table 1 Basic information about the ten standard benchmark datasets.

Datasets	Number of		
	Attributes	Classes	Instances
1. Banana	2	2	5300
2. Thyroid	5	2	215
3. Heart	13	2	270
4. Australian	14	2	690
5. German	20	2	1000
6. Dermatology	34	6	358
7. Led7digit	7	10	500
8. Ecoli	7	8	336
9. Wine	13	3	178
10. Yeast	8	10	1484

Using N arbitrary distinct samples, $(X_i, t_i) \in R^d \times R^m$, the solution of the output weights is given as:

$$\begin{bmatrix} G(a_1,b_1,X_1) & \cdots & G(a_L,b_L,X_1) \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ G(a_1,b_1,X_N) & \cdots & G(a_L,b_L,X_N) \end{bmatrix} \begin{bmatrix} \beta_1^T \\ \\ \vdots \\ \beta_L^T \end{bmatrix} = \begin{bmatrix} t_1^T \\ \\ \vdots \\ t_N^T \end{bmatrix} \quad (12)$$

The hidden layer output matrix of the SLFN is given as:

$$H = \begin{bmatrix} G(a_1,b_1,X_1) & \cdots & G(a_L,b_L,X_1) \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ G(a_1,b_1,X_N) & \cdots & G(a_L,b_L,X_N) \end{bmatrix} \quad (13)$$

The i^{th} column of the hidden matrix H would be the output of the hidden to the vector, $(x_1, x_2, \ldots, x_N)$. Hidden layer feature mapping is given by $G(a_1, b_1, x)$, ..., $G(a_L, b_L, x)$ and hidden layer feature mapping with respect to the i^{th} input, x_i, is defined as: $G(a_1, b_1, x_i)$, ..., $G(a_L, b_L, x_i)$. Huang et al. [25] showed that for an infinitely differentiable activation function, the hidden layer parameters may be randomly generated. The smallest norm least-squares solution of the linear system, is given as:

$$\hat{\beta} = H^{+}T \quad (14)$$

where H^+ is the Moore–Penrose generalized inverse of the hidden layer output matrix H.

For example, given a training set $N = \{(X_i, t_i) \mid X_i \in R^n, t_i \in R^m, i = 1, \ldots, N\}$, a hidden node output function, $G(a_1, b_1, x_i)$, and the number of hidden nodes, L, computation of an ELM algorithm consists of three steps, summarized as:

- Assign random hidden nodes by randomly generating parameters: (a_i, b_i), $i = 1, 2, \ldots L$.
- Calculate the hidden layer output matrix H.
- Calculate the output weight β: $\beta = H^+ T$.

According to Huang et al. [25], the universal approximation capability of an ELM is analyzed via an incremental method. It is widely known that single SLFN networks with randomly generated additive (or RBF) nodes having a wide range of activation functions can universally approximate any continuous target function within any compact subset of the Euclidean space R^n.

5. Experiments

5.1 Datasets

The current research is comprised of two experiments. In the first, the performance of the proposed M-SPXY algorithm is compared with that of the MDKS algorithm based on ten standard benchmark datasets from the repository of the University of California in Irvine (UCI) [33-34]. These datasets have been widely used in numerous model selections. In the second experiment, the proposed algorithm and the competitive algorithm are applied to five real world medical problems from the repository of the University of California in Irvine (UCI). Table 1 shows basic information about the ten standard benchmark datasets, which illustrates the various attributes or variables of the objects. Table 2 shows the details of the five medical problem datasets. Additionally, the attributes of each dataset have an effect upon the prediction accuracy and the generalization of ANN models. In both the first and second experiments, each dataset is randomly selected as a testing set (20%). Moreover, the rest of samples are divided into training (60%) and validation (20%) sets using the MDKS and M-SPXY methods respectively. These routines were implemented in a MATLAB environment. The procedure was repeated within each dataset before evaluation by the ELM.

5.2 Parameters and settings

These experiments were done to achieve an accurate comparison of the classification accuracy of the MDKS and M-SPXY algorithm based on two benchmark datasets. The experimental environment was evaluated on MATLAB 9.2.0 (R2017a), which was run on a personal computer, with a 3.2 GHz CPU and 8 GB RAM. The operating system was Microsoft Windows 7 (64-bit). An Extreme Learning Machine (ELM) was selected to test the performance of data splitting within the ANN model. ELMs have received attention from research communities because of the simplicity and fast training. The structure of the ELM, in this paper, consisted of 25 hidden neurons, classification types, and a sigmoidal activation function. There are three steps involved in measuring the performance of an ELM. The first is determining a training dataset. The second step is formation of a test dataset and the third is construction of a validation dataset. The classification performance of an ELM is measured as the classification accuracy of classifier k in sample S, which is the proportion of instances in S is correctly classified by k. The classification accuracy is defined as:

Table 2 Basic information about the five medical problem datasets.

Datasets	Number of		
	Attributes	Class	Instances
1. Pima Indian Diabetes (PID)	8	2	768
2. Wisconsin Breast Cancer (WDBC)	31	2	569
3. Liver Disorder (LD)	7	2	345
4. Haberman Surgery Survival (HSS)	3	2	306
5. Parkinson's (PD)	23	2	195

Table 3 Comparison classification accuracy of MDKS and M-SPXY algorithms using ten benchmark datasets.

Datasets	Methods	Testing Accuracy				Wilcoxon Signed-Rank Test p-value, α = 0.05
		Min	Max	Mean	Std	
1. Banana	MDKS	0.6386	0.8409	0.7478	0.0510	7.56E-10
	M-SPXY	0.8626	0.9110	**0.8911**	0.0118	
2. Thyroid	MDKS	0.5702	0.8555	0.7242	0.0689	3.28E-03
	M-SPXY	0.9611	0.9708	**0.9669**	0.0050	
3. Heart	MDKS	0.6700	0.8318	0.7596	0.0385	7.18E-03
	M-SPXY	0.8211	0.8741	**0.8523**	0.0149	
4. Australian	MDKS	0.8599	0.8919	0.8819	0.0141	7.56E-10
	M-SPXY	0.8640	0.8942	**0.8824**	0.0059	
5. German	MDKS	0.7000	0.7400	0.7207	0.0135	1.40E-02
	M-SPXY	0.7464	0.775	**0.7587**	0.0101	
6. Dermatology	MDKS	0.5298	0.7450	0.6557	0.0581	7.18E-03
	M-SPXY	0.5373	0.7943	**0.7289**	0.0951	
7. Led7digit	MDKS	0.7171	0.7285	0.7234	0.0044	7.39E-10
	M-SPXY	0.9658	0.9958	**0.9878**	0.0075	
8. Ecoli	MDKS	0.6382	0.7574	0.6970	0.0394	1.08E-09
	M-SPXY	0.7515	0.8074	**0.7788**	0.0079	
9. Wine	MDKS	0.4320	0.6560	0.5560	0.1916	2.27E-02
	M-SPXY	0.3888	0.6666	**0.5666**	0.1143	
10. Yeast	MDKS	0.5047	0.5726	0.5377	0.0165	9.79E-05
	M-SPXY	0.5574	0.6022	**0.5798**	0.0110	

$$Accuracy_s(k) = \frac{1}{n}\sum_{i=1}^{n} I(y_i = k(x_i)) \quad (15)$$

where $S = \{(x_i, y_i) \mid i = 1, \ldots, n\}$ is the given test set of observations. I (.) is the indicator function. The quality of algorithms is determined by the maximum value (max), minimum value (min), average value (mean), and standard deviation (std) of the testing accuracy of values that are computed from 50 replicates.

6. Results and discussion

6.1 Results of the first experiment

In the first experiment, the results demonstrate that the M-SPXY algorithm is better in dividing a large pool of data, into training and testing sets. Table 3 shows a comparison of the ELM classification accuracy using MDKS and M-SPXY methods employing ten standard benchmark datasets. A higher performing ELM is clearly visible in the classification accuracy of the M-SPXY method. For example, considering the results on the banana dataset, the average accuracy of M-SPXY is 0.8911 in testing. Whereas, the average accuracy using MDKS is 0.7478. Table 3 also indicates the accuracy of M-SPXY, where the results from all ten datasets were found to be higher than that of MDKS, highlighted with boldface type. Moreover, the Wilcoxon Signed-Rank test also confirms the statistically significant higher performance of M-SPXY over MDKS.

The first experiment also shows the different outcomes of the MDKS and M-SPXY methods. Scatter plots of 2-D dermatology data are shown with various dot colors in Figures 1 and 2, representing different classes. The red, yellow, green, light blue, dark blue, and purple dots depict classes 1-6, respectively. The special red cycles represent the selection symbol, i.e., the training data of the MDKS (Figure 1) and the M-SPXY algorithms (Figure 2).

Comparison of the results in Figures 1 and 2 show the training and testing subsets obtained through the consideration of information found in X and y. The MDKS algorithm proved unable to represent the entire distribution of data, failing to completely cover class 2 (Figure 1). The M-SPXY algorithm (Figure 2) did not exhibit this deficiency. The MDKS and M-SPXY algorithms share the same Mahalanobis measurement, yet have a different variance in each dataset due to the information provided by y. Moreover, the M-SPXY selected data by comparing all data, unlike the MDKS which selected data based only on the data's ranking. Owing to this observation, the ELM combined with M-SPXY should have outperformed the ELM combined with MDKS because it had a more reasonable result in the selection of training datasets of M-SPXY than did MDKS.

Table 4 Comparison classification accuracy of MDKS and M-SPXY algorithms using five medical datasets.

Datasets	Methods	Testing Accuracy				Wilcoxon Signed-Rank Test p-value, α = 0.05
		Min	Max	Mean	Std	
1. Pima Indian Diabetes (PID)	MDKS	0.6561	0.7137	0.6869	0.0171	7.45E-10
	M-SPXY	0.6970	0.7035	**0.7029**	**0.0020**	
2. Wisconsin Breast Cancer (WDBC)	MDKS	0.5906	0.8919	0.7465	0.1211	7.56E-10
	M-SPXY	0.6447	0.8933	**0.7723**	**0.1013**	
3. Liver Disorder (LD)	MDKS	0.5096	0.6923	0.6271	0.0541	6.75E-09
	M-SPXY	0.6788	0.7080	**0.6985**	**0.0097**	
4. Haberman Surgery Survival (HSS)	MDKS	0.7523	0.7717	0.7644	0.0076	7.18E-03
	M-SPXY	0.8934	0.9180	**0.9073**	**0.0067**	
5. Parkinson's (PD)	MDKS	0.7457	0.8305	0.7909	0.0317	7.56E-10
	M-SPXY	0.7435	0.8717	**0.8205**	**0.0314**	

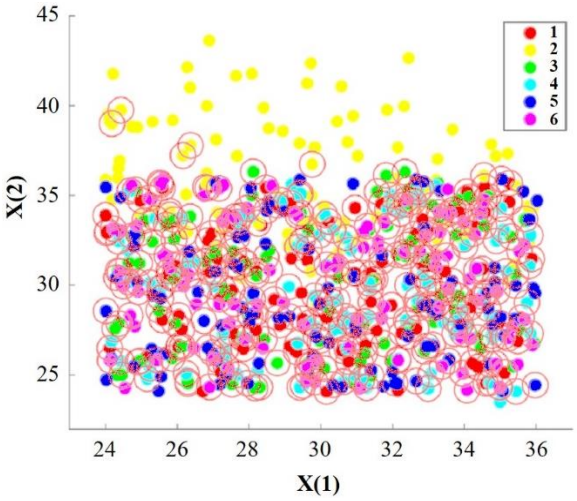


Figure 1 Data selected by MDKS.

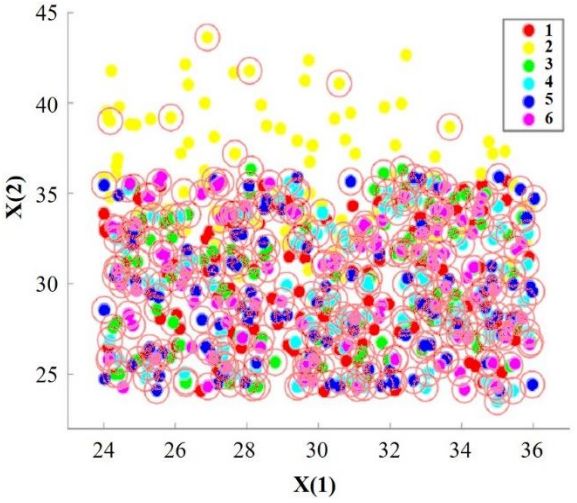


Figure 2 Data selected by M-SPXY.

6.2 Results of the second experiment

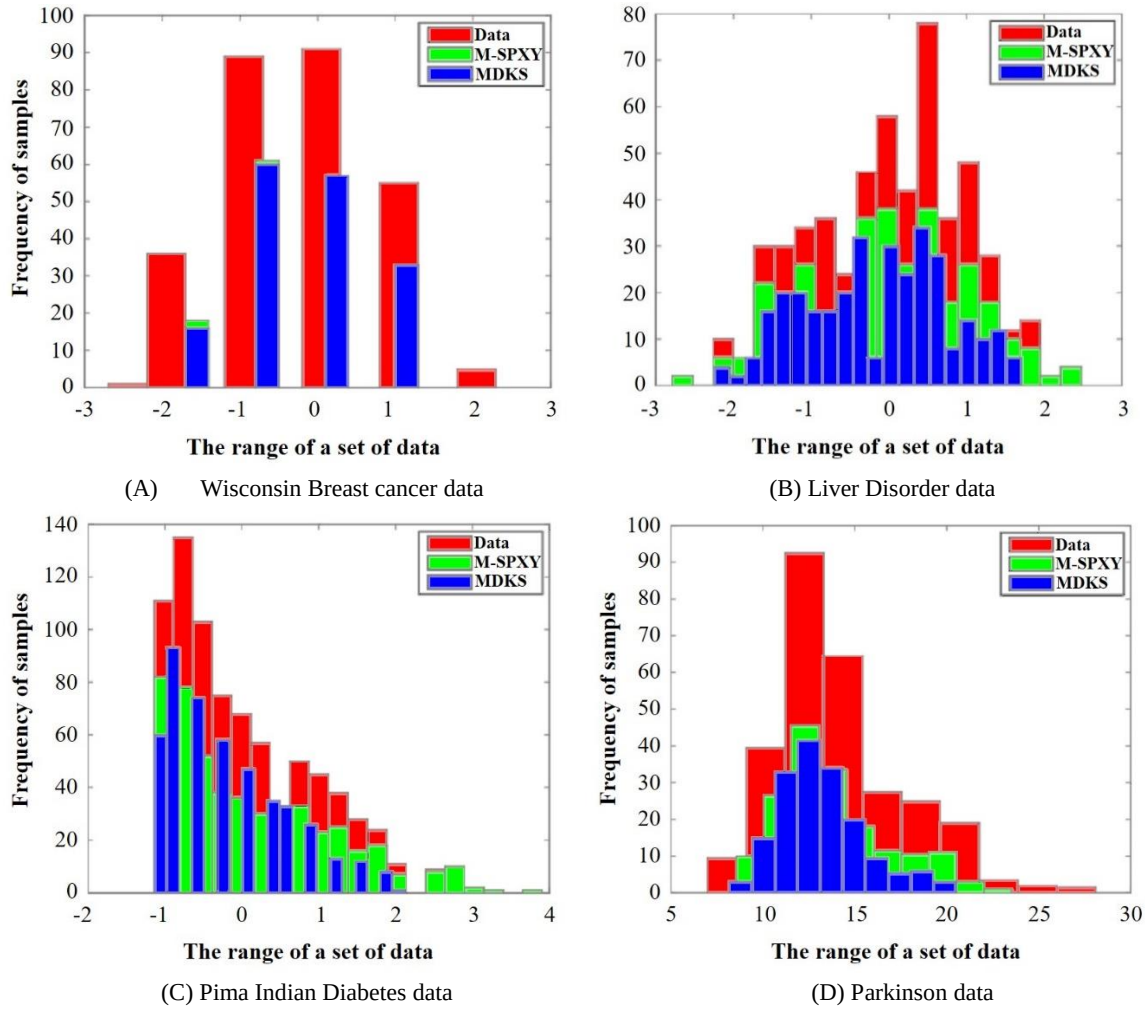
In the second experiment, the performance accuracy of MDKS and M-SPXY are compared using five medical data problems. Medical data is used to address the problems of diagnosis and treatment. Moreover, the details of data can assist physicians in making decisions about serious diseases by collecting symptoms and medical analyses. The parameter settings employed were as described in the experiments section. In Table 4, the results show that M-SPXY maintains a high level of testing accuracy and outperforms the MDKS using these five datasets, highlighted in boldface type. The Wilcoxon Signed-Rank test also confirms the statistically significant higher accuracy of M-SPXY over that of MDKS.

7. Discussion

7.1 The advantages of M-SPXY

Generally, with data splitting, the MDKS method (a modified form the classic KS method) is expected to outperform SPXY. The KS differs from SPXY in that KS focuses only on the independent variable vector (X), to represent the subsets of data. Euclidean distance is employed to determine the training subsets. SPXY, however, selects samples in a way similar to KS, but in addition to focusing on the independent variable vector (X), this algorithm also combines the dependent variable (y), enabling SPXY to outperform KS.

Recently developed data splitting methods reveal that MDKS proves superior in optimal data splitting by changing the Euclidean distance of KS to a Mahalanobis distance. The proposed method, however, also considers the perfect Mahalanobis distance, and the dependent variable (y) in the selection of subsets that influence heightened performance. Mahalanobis distance provides a higher similarity measure than the Euclidian distance. Figure 3 shows distributed samples, presented in three different color bars. The red, blue and green bars indicate distribution of the original value of the dataset, the distribution of samples split from the MDKS and M-SPXY algorithm, respectively. The X-axis depicts the range of minimum to maximum values of each of the datasets. Additionally, the Y-axis represents the distribution of each sample corresponding to the range of values on the X-axis. For example, in Figure 3 (A), the minimum value of the Wisconsin Breast cancer dataset was -3 while the maximum value was 3. The first red bar on the left side of Figure 3 (A) represents the two sample data which contained value in the range of -3 to -2. Whereas, the second red bar in Figure 3 (A) shows 38 data samples with values ranging from -2 to -1. MDKS successfully covers all selected samples (as the blue bar appears in every interval of data) within each dataset, Figure 3 (A) Wisconsin Breast cancer, Figure 3 (B) Liver Disorder, Figure 3 (C) Pima Indian Diabetes, and Figure 3 (D) Parkinson. However, the dependent variable (y) and Mahalanobis distance in our proposed makes the M-SPXY superior to the MDKS. This is evident from the



(A) Wisconsin Breast cancer data

(B) Liver Disorder data

(C) Pima Indian Diabetes data

(D) Parkinson data

Figure 3 Comparison of MDKS and M-SPXY in selection data. The X-axis is the distribution of the dataset and Y-axis is the frequency of each sample.

presence of the green bar in every dataset. The green bar indicates that selected samples overlap the blue bar. The distribution of the green bars indicates that the proposed M-SPXY could select data over a wider range than the MDKS algorithm.

7.2 M-SPXY abilities in avoiding over-training/over-fitting

The experimental results confirm that the M-SPXY method enhances the capability of ANN models. Additionally, M-SPXY also avoids the over-training/over-fitting problem, through the selection of qualified data for network models. Avoidance of overtraining/over-fitting helps to trim the results, which enhances its generalization capability, especially when the ANN models are required to handle training sets that are too large or too small for their resources. The M-SPXY method is capable of handling numerous problems in relation to the proportion of data it can manipulate. In Tables 3 and 4, the M-SPXY method reveals higher accuracy in classification within the sample datasets obtained from the ELM predictions. Thus, the quality of the data divided into training subsets will enhance the performance of the ANN model resulting in good generalization that helps prevent over-training and over-fitting.

8. Conclusions

Data splitting into subsets can enhance the performance and generalization capabilities of ANN models. This paper presents application of an M-SPXY data splitting method to select appropriate training sets for developing ANN models. The proposed M-SPXY method was modified from the original SPXY algorithm, in that the M-SPXY considers the variability of each dataset in a way that considers both independent X and dependent y spaces using the Mahalanobis distance, rather than merely the Euclidean distance of the X space only. The accuracy of the proposed method's performance was evaluated using an ELM neural network. In our experiments, the ELM's performance was compared using the data sets obtained from two different splitting methods, the M-SPXY and MDKS methods. In the comparison of two benchmark datasets, the results showed that the accuracy of ELM employing the M-SPXY method to split the data outperformed the MDKS method.

Further research is suggested for investigation of the distance function computational time with the goal of speeding-up the MD process. The MD method generally tends to consume more computational resources with large datasets.

9. Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

10. References

[1] Haykin S. Neural networks: a comprehensive foundation. USA: Prentice Hall PTR; 1998.

[2] Maier HR, Jain A, Dandy GC, Sudheer KP. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. Environ Model Software. 2010;25(8):891-909.

[3] Baskin I, Palyulin V, Zefirov N. Neural networks in building QSAR models. In: Livingstone D, editor. Artificial neural networks. USA: Humana Press; 2009. p. 133-54.

[4] Stojanović MB, Božić MM, Stanković MM, Stajić ZP. A methodology for training set instance selection using mutual information in time series prediction. Neurocomputing. 2014;141:236-45.

[5] Picard RR, Cook RD. Cross-validation of regression models. J Am Stat Assoc. 1984;79:575-83.

[6] May RJ, Maier HR, Dandy GC. Data splitting for artificial neural networks using SOM-based stratified sampling. Neural Networks. 2010;23(2):283-94.

[7] Wu W, Walczak B, Massart DL, Heuerding S, Erni F, Last IR, et al. Artificial neural networks in classification of NIR spectral data: Design of the training set. Chemometr Intell Lab Syst. 1996;33(1): 35-46.

[8] Refaeilzadeh P, Tang L, Liu H. Cross-Validation. In: Liu L, ÖZsu MT, editors. Encyclopedia of database systems. USA: Springer; 2009. p. 532-8.

[9] Pahikkala T, Suominen H, Boberg J, Salakoski T. Efficient hold-out for subset of regressors, In: Kolehmainen M, Toivanen P, Beliczynski B, editors. Adaptive and natural computing algorithms. Berlin: Springer Berlin Heidelberg; 2009. p. 350-9.

[10] Kennard RW, Stone LA. Computer Aided Design of Experiments. Technometrics. 1969;11(1):137-48.

[11] Galvão RKH, Araujo MCU, José GM, Pontes MJC, Silva EC, Saldanha TCB. A method for calibration and validation subset partitioning. Talanta. 2005;67(4): 736-40.

[12] Saptoro A, Tadé MO, Vuthaluru H. A modified kennard-stone algorithm for optimal division of data for developing artificial neural network models. Chem Prod Process Model. 2012;7(1):1-14.

[13] Maesschalck RD, Jouan-Rimbaud D, Massart DL. The Mahalanobis distance. Chemometr Intell Lab Syst. 2000;50(1):1-18.

[14] Yang KC, Huang CH, Yang C, Chao PY, Shih PH. Using artificial neural back-propagation network model to detect the outliers in semiconductor manufacturing machines. In: Ali M, Pan JS, Chen SM, Horng MF, editors. Modern Advances in Applied Intelligence. Berlin: Springer International Publishing; 2014. p. 240-9.

[15] Sun Z, Wang J, Nie L, Li L, Cao D, Fan J, Zang H. Calibration transfer of near infrared spectrometers for the assessment of plasma ethanol precipitation process. Chemometr Intell Lab Syst. 2018;181:64-71.

[16] Tian H, Zhang L, Li M, Wang Y, Sheng D, Liu J, et al. Weighted SPXY method for calibration set selection for composition analysis based on near-infrared spectroscopy. Infrared Phys Tech. 2018;95:88-92.

[17] Gueorguieva N, Valova I, Georgiev G. M&MFCM: fuzzy c-means clustering with mahalanobis and minkowski distance metrics. Procedia Comput Sci. 2017;114:224-33.

[18] Wu W, Mallet Y, Walczak B, Penninckx W, Massart DL, Heurding S, et al. Comparison of regularised discriminant analysis, linear discriminant analysis and quadratic discriminant analysis, applied to NIR data. Anal Chim Acta. 1996;329(3):257-65.

[19] Jouan-Rimbaud D, Massart DL, Saby CA, Puel C. Characterisation of the representativity of selected sets of samples in multivariate calibration and pattern recognition. Anal Chim Acta. 1997;350(1-2):149-61.

[20] Jouan-Rimbaud D, Massart DL, Saby CA, Puel C. Determination of the representativity between two multidimensional data sets by a comparison of their structure. Chemometr Intell Lab. 1998;40(2):129-44.

[21] Shenk JS, Westerhaus MO. Population definition, sample selection, and calibration procedures for near infrared reflectance spectroscopy. Crop Sci. 1991; 31(2):469-74.

[22] Dean T, Isaksson T. Standardisation: What is it and how is it done? Part 1. NIR news. 1993;4:8- 9.

[23] Bouveresse E, Massart DL. Standardization of near-infrared spectrometric instruments: a review. Vibr Spectrosc. 1996;11:3-15.

[24] Dean T, Kowalski BR. Multivariate instrument standardization: review of the state of the art. In: Brown SD, editor. Computer Assisted Analytical Spectroscopy. Chichester, UK: Wiley; 1996. p. 175-87.

[25] Huang GB, Zhu QY, Siew CK. Extreme learning machine: Theory and applications. Neurocomputing. 2006;70(1-3):489-501.

[26] Zhang Y, Zhang P. Optimization of nonlinear process based on sequential extreme learning machine. Chem Eng Sci. 2011;66(20):4702-10.

[27] Sun Y, Yuan Y, Wang G. An OS-ELM based distributed ensemble classification framework in P2P networks. Neurocomputing. 2011;74(16):2438-43.

[28] Zhao LJ, Chai TY, Diao XK, Yuan DC. Multi-class Classification with One-Against-One Using Probabilistic Extreme Learning Machine. In: Wang J, Yen G, Polycarpou M, editors. Advances in Neural Networks. Berlin: Springer Berlin Heidelberg; 2012. p. 10-19.

[29] Qiu S, Gao L, Wang J. Classification and regression of ELM, LVQ and SVM for E-nose data of strawberry juice. J Food Eng. 2015;144:77-85.

[30] Chacko B, Vimal Krishnan VR, Raju G, Babu Anto P. Handwritten character recognition using wavelet energy and extreme learning machine. Int J Mach Learn Cybern. 2012;3(2):149-61.

[31] Li G, Liu M, Dong M. A new online learning algorithm for structure-adjustable extreme learning machine. Comput Math Appl. 2010;60(3):377-89.

[32] Junhai Z, Jinggeng W, Xizhao W. Ensemble online sequential extreme learning machine for large data set classification. 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC); 2014 Oct 5-8; San Diego, USA. USA: IEEE; 2014. p. 2250-5.
[33] Soliman OS, Aboelhamd E. Classification of breast cancer using differential evolution and least squares support vector machine. Int J Emerg Trends Technol Comput Sci. 2014;3:155-61.
[34] Dutta RK, Karmakar NK, Si T. Artificial neural network training using fireworks algorithm in medical data mining. Int J Comput Appl. 2016;137:1-5.