Chemical Product and Process Modeling

Volume 7, *Issue* 1 2012 *Article* 13

A Modified Kennard-Stone Algorithm for Optimal Division of Data for Developing Artificial Neural Network Models

Agus Saptoro, *Curtin University, Australia*
Moses O. Tadé, *Curtin University, Australia*
Hari Vuthaluru, *Curtin University, Australia*

Recommended Citation:
Saptoro, Agus; Tadé, Moses O.; and Vuthaluru, Hari (2012) "A Modified Kennard-Stone Algorithm for Optimal Division of Data for Developing Artificial Neural Network Models," *Chemical Product and Process Modeling*: Vol. 7: Iss. 1, Article 13.
DOI: 10.1515/1934-2659.1645

A Modified Kennard-Stone Algorithm for Optimal Division of Data for Developing Artificial Neural Network Models

Agus Saptoro, Moses O. Tadé, and Hari Vuthaluru

Abstract

This paper proposes a method, namely MDKS (Kennard-Stone algorithm based on Mahalanobis distance), to divide the data into training and testing subsets for developing artificial neural network (ANN) models. This method is a modified version of the Kennard-Stone (KS) algorithm. With this method, better data splitting, in terms of data representation and enhanced performance of developed ANN models, can be achieved. Compared with standard KS algorithm and another improved KS algorithm (data division based on joint x - y distances (SPXY) method), the proposed method has also shown a better performance. Therefore, the proposed technique can be used as an advantageous alternative to other existing methods of data splitting for developing ANN models. Care should be taken when dealing with large amount of dataset since they may increase the computational load for MDKS due to its variance-covariance matrix calculations.

KEYWORDS: data division, kennard-stone algorithm, MDKS, ANN models

Author Notes: This work was part of the PhD project supported by TPSDP scholarship and Curtin International Research Tuition Scholarship. The authors also wish to thank the reviewers for their constructive comments.

Agus Saptoro, Moses O. Tadé, Hari Vuthaluru: Department of Chemical Engineering, Curtin University, Australia

1. Introduction

Optimal data division into a training dataset and an independent test subset is an important and critical step in artificial neural networks (ANN) modelling for complex data analysis. Recent studies have found that data splitting into subsets can have a significant influence on ANN's performance (Bowden *et al*., 2002; Minns *et al*., 1996; Maier *et al*., 2000; Maier *et al*., 2010). Typically, an ANN model is only capable of interpolating data and it is unable to extrapolate beyond the range of dataset used for training. As a consequence, poor predictive ability is expected when the trained model is tested using the dataset having values outside the range of training dataset. Flood et al. (1994) also noted that the performance of an ANN model strongly depends on the number of available training samples. The higher the number of training samples available for ANN modeling, the higher the potential level of accuracy can be obtained by the model. However, from practical points of view, having sufficient data for training ANN model is not always possible. Sometimes engineers and/or researchers can only obtain limited data for costly measurements and experimental procedures. In few cases, due to faulty sensors or maintenance activities, missing measurements result in small amount of collected data. Hence, the proportion of samples to include in each of the subsets becomes very important consideration.

There are several existing data division methods for systematically partitioning the available data into statistically representative subsets: random selection (RS) (Rajer-Kanduč *et al*., 2003; Wu *et al*., 1996; Kocjančič *et al*., 2000; Saptoro *et al*., 2008), kohonen self organising map (SOM) (Bowden *et al*., 2002; Rajer-Kanduč *et al*., 2003; Wu *et al*., 1996), genetic algorithm (GA) based approach (Bowden *et al*., 2002), D-optimal design (Wu *et al*., 1996; Atkinson *et al*., 1995; de Aguiar *et al*., 1995) and Kennard-Stone (KS) algorithm (Rajer-Kanduč *et al*., 2003; Wu *et al*., 1996; Kocjančič *et al*., 2000; Saptoro *et al*., 2008; Galvão *et al*., 2005; Kennard *et al*., 1969). The random selection approach is the way of selecting training set by applying random division where no clear selection criteria is applied. Due to its simplicity, this method is the most commonly used (Kocjančič *et al*., 2000).

The second data division method, SOM, is used to cluster the data by presenting ANN input and output variables as the SOM's inputs. A SOM grid is then specified, where each cell in the grid represents a node in the Kohonen layer. By training SOM, similar data samples are clustered into each dot representing a sample of data. Using the method employed by Bowden et al. (2002), three data records from each cluster are sampled and allocated to each of the training, testing and validation subsets. However, if a cluster only contains one record, this record is allocated to the training set. If a cluster contains two records, one record is placed in the training set and the other one is in the testing set. This technique has

an advantage over other data division methods where it avoids the need to arbitrarily select which proportion of data to be included in each subset. It is also capable of constructing a representative training data set using the minimum number of samples.

The GA for data division is designed to allocate available data into pre-specified proportions of training and testing sets according to a set of pseudo-random numbers. Pseudo-random number seed is used to determine the optimal allocation of data into subsets. Therefore, this seed is optimized as a decision variable. The objective function of the optimization is to minimize sum of the absolute difference in mean and standard deviation values for each input and output variable between each pair of the two subsets as indicated by the equation below (Bowden *et al*., 2002).

$$Objective\ function = \sum_{i=1}^{K+1} \{[\mu(i)_{train} - \mu(i)_{test}] + [\sigma(i)_{train} - \sigma(i)_{test}\} \quad (1)$$

K is the number of inputs and μ and σ are mean and standard deviation of the input or output variable, respectively. To ensure that the maximum and minimum values of each variable are included in the training set, penalty constraints are added to the objective function. Penalty constraints are preferable than manually removing extreme values from the data and placing them in the training set. This is because of a possible trade-off between keeping the statistical properties of the training and testing sets and at the same time ensuring the extreme values in the training set.

The KS algorithm technique was originally applied to generate a training set when no standard experimental design can be implemented. With this technique, all objects are considered as candidates for the training set. The selected candidates are chosen sequentially. KS algorithm can be summarized as follows: First, the KS algorithm takes the pair of samples with the largest Eucledian distance of *x*-vectors (predictors) and then it sequentially selects a sample to maximize the Eucledian distance between *x*-vectors of already selected samples and the remaining samples. This process is repeated until the required number of samples is achieved. For each pair of samples i and j, the Eucledian distance in *x* space is defined as (Wu *et al*., 1996; Saptoro *et al*., 2008; Galvão *et al*., 2005; Kennard *et al*., 1969)

$$d_x(i,j) = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{M} (x_{ik} - x_{jk})^2} \quad (2)$$

It is well-known that each proposed approach above suffers from some drawbacks. Firstly, when using RS method, since there is no clear selection criteria applied, there is a risk that some 'rich information' sets of data are not selected in the training set. The random nature of this method may result in uncertain characteristics into the selected training and testing sets. Consequently, no guarantee that representative training data set can be obtained.

Partitioning data with a GA can be very time consuming as finding the best allocation requires comparisons of many different combinations of data arrangement. In a simple example given by Bowden et al. (2002), if there are 60 sets of data which must be divided into 45 data for training set and 15 data for testing set, there will be

$$\frac{60!}{(45!).(15!)} = 5.32 \; . \; 10^{13} \tag{3}$$

ways of arranging samples of data. In reality, it is unlikely that an optimal data division will be found within a reasonable time frame, although good results can still be obtained. Moreover, as noted by Bowden et al. (2002), the cross-over operator is usually unable to function as it should. Crossing over two random number seeds does not result in a set of random numbers that inherits the properties of the parents. Thus this method only relies on the selection and mutation to find an appropriate data set which further slowdown the process.

Using data division method based on SOM, the number of data allocated to each data subset depends on the specified Kohonen grid size. However, as indicated by Bowden et al. (2002), there is no theoretical principle for determining the optimum size of the Kohonen layer. According to Shahin et al. (2004), the specified grid size can have a significant impact on the results obtained from SOM data division method. This is because of an underlying assumption that the data in one cluster should provide the same information in high-dimensional space. Bowden et al. (2002) also stated that the selected grid size should be large enough to ensure that the maximum number of clusters are formed from the available data. However, even though the grid size may be large enough, sometimes each grid only contains one sample of data. This creates difficulties in choosing representative subsets. Furthermore, by only selecting one sample from each cluster, the amount of data used for developing ANN model is significantly reduced. In one of the case studies presented by Bowden et al. (2002), 2005 available data were reduced to 147 data with only 49 data allocated to each data subset. Such a reduction in data analysis may result in a significant loss of information where the selected training data set will not adequately represent the population of data. This lost normally depends on the intra cluster

variation. If this variation is large, important information may be omitted from the training set by only selecting one sample from each cluster.

In their work, Bowden et al. (2002) indicated that SOM based method performs better than GA and the random approaches. However, in later work done by Galvão et al. (2005), KS method is preferred than SOM due to its ability to select more representative training data set. Moreover, KS algorithms are far simpler than the SOM and GA algorithm.

Despite the comparative advantage of KS over the other methods, KS algorithm has also a possible shortcoming. In multivariate data analysis, dependent variable (y) and independent variable (x) are statistically related. Therefore, data division criteria should take into account the statistical contributions from both x and y. Whereas KS algorithm only incorporates the variability of the independent variable. It could be postulated that the inclusion of y information in the selection process might result in a more effective distribution of training set in the multidimensional space, thus it may improve the predictive ability and robustness of the developed model.

A KS algorithm which considers variabilities in both x and y dimensions, was first proposed by Galvão et al. (2005). The method, namely sample set partitioning based on joint x-y distances (SPXY), extends the original KS algorithm by encompassing both x- and y- differences in the calculation of inter-sample distances. In this method, Galvão et al. (2005) still utilised the Eucledian distance to measure variabilities in both x and y spaces. They found that SPXY technique outperforms both RS and KS based training set design algorithms.

In statistics, it has been agreed that Mahalonobis distance gives better distance analysis compared with Eucledian approach especially in their applications for detecting outliers (Maesschalck *et al*., 2000). To date, however, no studies have been dedicated to incorporate Mahalanobis distance as selection criteria for data division in developing empirical models including ANN models.

In this work, we propose data splitting method for ANN modelling based on Mahalanobis distance (MD) framework. Although the concept of MD is not new, its application is mainly limited on the outlier detection method and no one has utilised MD as criteria for data partition. Therefore, the new method, namely MDKS or KS based on MD method, was used for partitioning the dataset for developing ANN models and its performance was compared with standard KS and SPXY method.

2. Theory and Algorithms

2.1. KS algorithm

The classical KS algorithm aims to select a representative subset from a pool of N samples. To ensure a uniform distribution of such a subset along the *x* data space, the algorithm follows a stepwise procedure in which new selections are taken in the regions of space located far from the already selected samples. For this purpose, the algorithm employs Eucledian distance $ED_x(p,q)$ between the *x*-vectors of each pair (p,q) of samples as shown by the equation below (Wu *et al*., 1996; Saptoro *et al*., 2008; Galvão *et al*., 2005; Kennard *et al*., 1969).

$$ED_x(p,q) = \sqrt{\sum_{j=1}^{N}\left[x_p(j) - x_q(j)\right]^2} \qquad p,q \in [1,M] \qquad (4)$$

In Eq. (4), N is the number variables in x and M is the number of samples. $x_p(j)$ and $x_q(j)$ are the j th variable for samples p and q, respectively.

The selection starts by taking a pair of samples for which the distance is the largest. At each subsequent iteration, the algorithm selects the sample that exhibits the least distance with respect to any sample already selected. Such a procedure is repeated until the number of samples required is achieved.

2.2. SPXY algorithm

The basic principle of SPXY algorithm is to combine the contributions from the distance defined in Eq. (4) with the distance in the dependent variable y space for parameter under consideration. Such a distance $ED_y(p,q)$ can be calculated for each pair of samples p and q as (Galvão *et al*., 2005)

$$ED_y(p,q) = \sqrt{\sum_{k=1}^{K}\left(y_p(k) - y_q(k)\right)^2} \qquad ; p,q \in M \qquad (5)$$

where K is the number of variables in y and $y_p(k)$ and $y_q(k)$ are the k th variable for samples p and q, respectively.

In order to assign an equal importance to the distributions of samples in the x and y spaces, distances $ED_x(p,q)$ and $ED_y(p,q)$ are divided by their maximum values in the data set. In this manner, a normalised *xy* distance is calculated as (Galvão *et al*., 2005)

$$ED_{xy} = \frac{ED_x(p,q)}{\max_{p,q\in[1,M]} ED_x(p,q)} + \frac{ED_y(p,q)}{\max_{p,q\in[1,M]} ED_y(p,q)} \qquad (6)$$

A stepwise selection procedure similar to the KS algorithm can then be applied using selection criteria of $ED_{xy}(p,q)$ instead of $ED_x(p,q)$.

2.3. Proposed MDKS algorithm

The proposal of the present work involves calculating Mahalanobis distance of the matrix *z* where *z* is the augmentation matrix of the matrices *x* and *y*. Then $MD(p,q)$ is calculated using Eq. (7) (Maesschalck *et al*., 2000)

$$MD(p,q) = \sqrt{E'(p,q)C^{-1}(p,q)E(p,q)} \qquad (7)$$

where

$$E'(p,q) = \left[e_{pq}(1)\, e_{pq}(2) \ldots e_{pq}(u) \ldots e_{pq}(F)\right] \qquad (8)$$

and

$$e_{pq}(u) = \left(z(p,u) - z(q,u)\right) \qquad ; p,q \in M \qquad ; u = 1,2,\ldots,F \qquad (9)$$

F is number of variables in matrix *z*. Meanwhile, $C(p,q)$ is the covariance matrix of matrix E, where (Maesschalck *et al*., 2000)

$$C(p,q) = \frac{1}{M}\left(E'(p,q)E(p,q)\right) \qquad (10)$$

Step-by-step procedure of the MDKS is analogous to the SPXY algorithm where $MD(p,q)$ is used instead of $ED(p,q)$. This proposed method is expected to be better than the SPXY method since besides it accounts for the integrated relations between *x* and *y*, it also uses Mahalanobis distance as selection criteria instead of Eucledian distance. Theoretically it has been proven that Mahalanobis distance has a better similarity measure than of Eucledian distance (Maesschalck *et al*., 2000).

3. Simulation studies

In this research, three approaches for data partition, two of them are established methods (KS and SPXY) and the other is the proposed one (MDKS), were tested through case studies of predicting carbon content (C) in coal using its proximate analysis (case study 1) and estimating calorific value (CV) of coal using its proximate and ultimate analysis (case study 2). The original datasets were extracted from coal database compiled by Hatch et al. (2006). The independent variables for predicting C are fixed carbon (FC), volatile matter (VM) and moisture content (M). Meanwhile, for predicting CV, C, H no H_2O, FC, N, VM, H, O, O no H_2O and organic S (O-S) were selected for input variables. All variables were measured in weight percentage - as-received basis except CV is in MJ/kg – as-received basis. The sizes of available datasets are respectively 90 x 4 and 90 x 10 where the size of matrix y is 90 x 1 for both cases. Tables 1 and 2 summarise the statistical properties of original datasets.

Table 1: Statistical Properties of Case Study 1 Data.

Statistical Properties	FC	VM	M	C
Minimum	8.59	11.21	0.57	12.46
Maximum	71.01	38.09	30.28	82.02
Mean	45.18	29.31	12.06	59.96
Standard Deviation	11.97	7.22	9.86	13.60
Skewness	-0.32	-1.06	0.32	-1.07
Kurtosis	0.35	-0.19	-1.46	2.29

Table 2: Statistical Properties of Case Study 2 Data.

Statistical properties	C	H no H_2O	FC	N	VM	H	O	O no H_2O	O-S	CV
Minimum	12.46	1.12	8.59	0.02	11.21	1.18	1.85	0.16	0.02	5.07
Maximum	82.02	5.28	71.01	1.97	38.09	7.10	38.45	13.61	1.35	32.88
Mean	59.96	3.83	45.18	1.14	29.31	5.18	18.91	8.46	0.49	24.22
Standard Deviation	13.60	0.79	11.97	0.40	7.22	1.31	13.03	4.63	0.28	5.55
Skewness	-1.07	-1.01	-0.32	-0.32	-1.06	-1.06	0.04	-0.61	0.69	-0.98
Kurtosis	2.29	2.62	0.35	-0.23	-0.19	0.97	-1.69	-1.09	0.07	1.97

To test each method, ANN models were developed and simulated using ANN Toolbox of MATLAB 7.1 software. Since the main objective of this study is to propose MDKS algorithm based data division method and to compare its performance with other algorithms, other network and training parameters which may influence the model performance such as activation functions, training algorithm, number of iterations, weight initialization algorithm etc were kept constant. The performances were evaluated in terms of training performance and

generalisation ability (testing performance) where mean squared errors (MSE) were calculated. Figures 1 and 2 show the MSE versus number of hidden neurons for case studies 1 and 2.

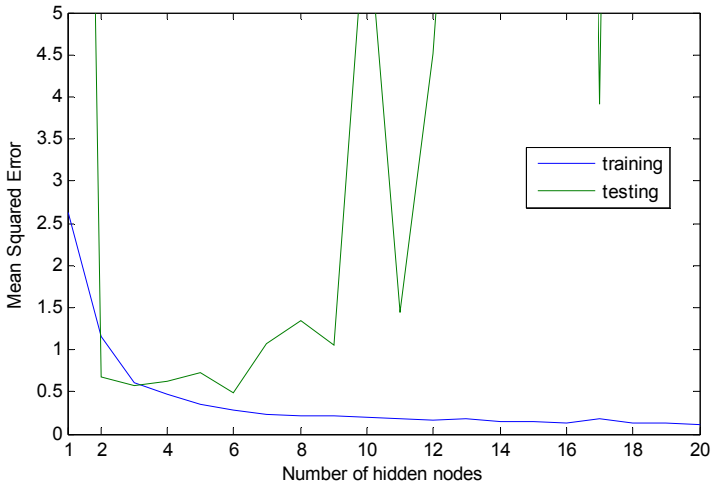

Figure 1: Simulation results representing MSE vs number of hidden nodes for Case Study 1.

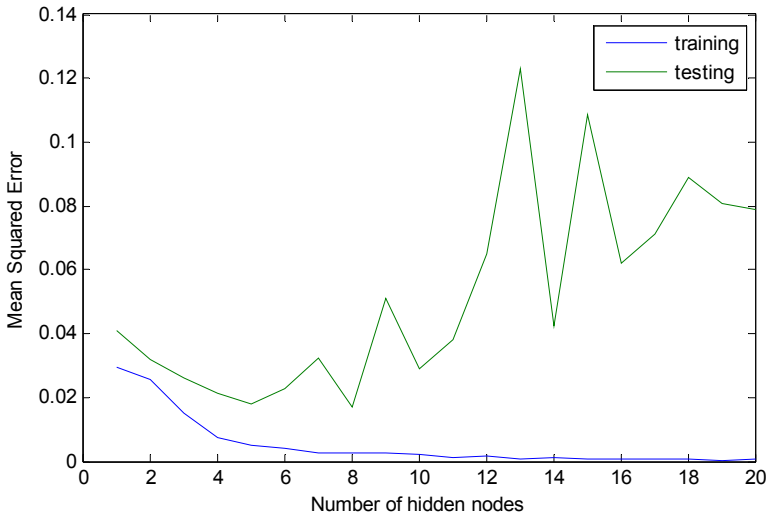

Figure 2: Simulation results representing MSE vs number of hidden nodes for Case Study 2.

From these two figures, it is evident that the optimum network configurations are 3 – 6 – 1 and 9 – 8 - 1 for case studies 1 and 2, respectively.

4. Results and discussion

4.1. Dataset characteristics

After all three data division techniques were applied into the original datasets, the training and testing subsets were chosen. The proportion of data used in the case studies is 80% and 20% for training and testing data, respectively. Tables 3 and 4 show statistical properties of both training and testing data sets.

It is apparent that for both case studies, the statistical properties of training datasets obtained from the three methods are similar in terms of data range, except for case 1 where training set chosen by the standard KS has the narrowest region. It is also found MDKS method is able to choose the narrowest region of the testing subset shown by its minimum and maximum values of each variable. This indicates that the MDKS method is not only able to select wide area but also dense area of training set. This characteristic is beneficial since the trained model is expected to have better generalisation ability.

Table 3: Statistical Properties of Data after Data Division for Case Study 1.

Statistical Properties	FC	VM	M	C
KS-algorithm				
Training set				
Minimum	17.69	12.59	0.64	20.08
Maximum	71.01	38.09	30.28	82.02
Mean	44.41	29.99	14.54	59.24
Standard Deviation	10.33	6.85	9.50	10.86
Skewness	0.37	-1.25	-0.06	-0.33
Kurtosis	-0.19	0.22	-1.45	1.42
Test set				
Minimum	8.59	11.21	0.57	12.46
Maximum	68.45	37.31	5.35	81.85
Mean	48.25	26.60	2.17	62.82
Standard Deviation	17.08	8.17	1.55	21.57
Skewness	-1.39	-0.52	0.74	-1.68
Kurtosis	1.09	-0.93	-0.66	1.67
SPXY-algorithm				
Training set				
Minimum	8.59	11.21	0.57	12.46
Maximum	71.01	38.09	30.28	82.02
Mean	43.59	28.62	14.59	57.72
Standard Deviation	12.00	7.63	9.45	13.03
Skewness	-0.09	-0.90	-0.06	-1.16
Kurtosis	0.67	-0.67	-1.43	3.16
Testing set				
Minimum	23.42	19.31	0.69	30.39
Maximum	61.91	37.31	4.61	79.89
Mean	51.52	32.06	1.97	68.89
Standard Deviation	9.75	4.43	1.11	12.43

Skewness	-1.67	-1.39	0.69	-2.03
Kurtosis	2.87	2.80	-0.03	4.59
MDKS-algorithm				
Training set				
Minimum	8.59	11.21	0.57	12.46
Maximum	71.01	38.09	30.28	82.02
Mean	47.44	28.84	10.39	62.07
Standard Deviation	12.30	7.72	8.97	14.42
Skewness	-0.81	-0.88	0.56	-1.52
Kurtosis	1.10	-0.68	-1.10	3.07
Testing set				
Minimum	32.23	16.46	0.74	47.18
Maximum	42.61	36.12	28.55	55.90
Mean	36.11	31.19	18.77	51.51
Standard Deviation	3.00	4.39	10.66	2.50
Skewness	0.83	-2.36	-0.90	0.19
Kurtosis	-0.28	7.26	-0.98	-0.36

Table 4: Statistical Properties of Data after Data Division for Case Study 2.

Statistical properties	**C**	**H no H_2O**	**FC**	**N**	**VM**	**H**	**O**	**O no H_2O**	**O-S**	**CV**
KS-algorithm										
Training set										
Minimum	12.4	1.12	8.59	0.02	11.2	1.18	1.85	0.16	0.02	5.07
Maximum	82	5.28	71.0	1.97	38.0	7.10	38.4	13.6	1.35	32.8
Mean	61.9	3.91	47.3	1.21	28.8	5.09	16.8	7.86	0.54	25.0
Standard Deviation	14.5	0.86	12.4	0.42	7.70	1.33	12.4	4.66	0.29	5.90
Skewness	-1.47	-1.22	-0.78	-0.75	-0.87	-1.08	0.26	-0.45	0.53	-1.39
Kurtosis	2.85	2.46	0.90	0.26	-0.68	1.14	-1.56	-1.30	-0.03	2.56
Testing set										
Minimum	47.1	2.70	32.2	0.72	16.4	3.15	4.02	0.43	0.16	18.7
Maximum	55.9	3.98	42.6	1.18	36.1	6.72	37.0	13.5	0.63	23.6
Mean	52.0	3.53	36.7	0.87	31.2	5.56	26.9	10.8	0.31	20.9
Standard Deviation	2.31	0.26	2.84	0.15	4.42	1.18	12.3	3.74	0.18	1.26
Skewness	-0.37	-1.67	0.36	1.07	-2.38	-0.91	-1.09	-1.72	1.05	0.81
Kurtosis	0.59	5.50	-0.40	-0.45	7.28	-0.85	-0.60	2.39	-0.77	0.91
SPXY-algorithm										
Training set										
Minimum	13.8	1.28	12.6	0.02	11.2	1.35	1.85	0.16	0.02	5.70
Maximum	82.0	4.90	71.	1.97	38.0	7.10	38.4	13.6	1.35	32.4
Mean	58.4	3.69	44.1	1.08	28.8	5.32	21.4	8.80	0.44	23.4
Standard Deviation	11.8	0.63	11.2	0.37	7.38	1.31	13.4	5.09	0.27	4.74
Skewness	-0.91	-1.56	0.18	-0.18	-0.95	-1.08	-0.41	-0.77	1.09	-0.91
Kurtosis	2.95	5.15	0.30	0.04	-0.55	0.77	-1.53	-1.19	1.36	2.68
Testing set										
Minimum	12.4	1.12	8.59	0.19	12.3	1.18	5.72	5.12	0.23	5.07
Maximum	79.8	5.28	61.9	1.80	37.3	5.58	13.3	9.26	1.07	32.8
Mean	65.9	4.39	49.3	1.37	30.9	4.61	8.84	7.10	0.70	27.2
Standard Deviation	18.2	1.09	14.0	0.44	6.43	1.16	2.23	1.31	0.24	7.43
Skewness	-2.05	-2.13	-1.91	-1.55	-1.80	-2.02	0.37	0.06	-0.36	-2.12
Kurtosis	3.95	4.41	3.45	2.31	3.52	3.91	-0.98	-1.36	-0.40	4.32
MDKS-algorithm										

Training set										
Minimum	12.4	1.12	8.59	0.02	11.2	1.18	1.85	0.16	0.02	5.07
Maximum	82.0	5.28	71.0	1.97	38.0	7.10	38.4	13.6	1.35	32.8
Mean	62.0	3.90	47.5	1.23	28.4	4.90	15.0	7.42	0.54	25.1
Standard Deviation	14.4	0.87	12.3	0.40	7.78	1.29	11.5	4.60	0.26	5.85
Skewness	-1.52	-1.18	-0.84	-0.90	-0.75	-0.93	0.45	-0.29	0.41	-1.47
Kurtosis	3.06	2.20	1.14	0.87	-0.87	0.85	-1.37	-1.35	-0.02	2.86
continued										
Testing set										
Minimum	47.8	3.28	32.0	0.69	29.1	4.34	20.1	11.0	0.16	19.1
Maximum	53.0	3.93	38.5	0.98	36.1	6.85	37.4	13.5	1.28	21.4
Mean	51.4	3.56	35.8	0.78	32.9	6.29	34.3	12.6	0.30	20.5
Standard Deviation	1.65	0.13	1.95	0.08	1.55	0.54	3.69	0.81	0.27	0.59
Skewness	-0.98	0.89	-0.04	1.64	-0.20	-2.97	-3.70	-0.66	3.05	-0.92
Kurtosis	-0.38	4.05	-1.52	2.81	1.48	11.0	15.0	-1.05	10.0	0.42

4.2. ANN model performances and accuracy

Performances of the developed models using training set obtained from KS, SPXY and MDKS data splitting techniques are shown in Table 5. The model performances for all cases are similar. This is logical since the models were developed using very similar range of data. However, from this table also, it is evident that from the mean squared error of the testing data, there are differences in terms of generalisation abilities. Models built using training data from data partition based SPXY algorithm performs better than of the standard KS and this is consistent with the results from Galvao et al [9]. Further analysis on the generalisation ability of the models, it can be clearly seen that performances of the models developed using training data selected by the proposed method are superior to the SPXY. The results from experimental studies show that this improved method is not only capable of improving the performance of the ANN model but also it outperforms both standard KS and SPXY methods. It is evident that the use of MDKS is able to improve the performance of training and testing by up to 33 % and 75 %, respectively, compared with KS method and by up to 16 % and 57 %, respectively, compared with SPXY approach. Consequently, it appears the MDKS approach is a more suitable approach for dividing data into training and testing datasets for more accurate ANN modelling.

Table 5: ANN model performances trained with training data developed from different data division method.

Data Division Method	Prediction of C		Prediction of CV	
	MSE of Training	MSE of Testing	MSE of Training	MSE of Testing
KS	0.25	15.50	2.37×10^{-4}	8.86
SPXY	0.25	8.98	1.88×10^{-4}	5.29
MDKS	0.25	3.85	1.57×10^{-4}	2.28

Table 6: Comparisons of computational loads for KS, SPXY and MDKS data division methods.

Dataset	Data division technique	Computational Time (s)
1 (90 x 4)	KS	0.1192
	SPXY	0.1323
	MDKS	0.1737
2 (90 x 10)	KS	0.2335
	SPXY	0.2784
	MDKS	0.392

4.3. Computational load

Despite its superiority to standard KS and SPXY methods, MDKS approach requires longer computational time than of KS and SPXY. Theoretically, this is correct since MDKS approach includes calculations of variance-covariance matrices. Consequently, the larger the dataset the heavier the computational load for this calculation would be. Table 6 presents the comparisons of computational loads for the three data division methods assessed when the algorithms were run in a CPU with a processor of Intel (R) Core (TM) 2 Duo E7300 having 2.66 GHz in speed and 1 GB in memory. From this table, it can be seen that for both datasets, computational loads for MDKS data division technique are up to 68 % higher than of standard KS and 41 % higher than of SPXY. Despite of this longer computational time, all the numerical values of the computational times for the three techniques are actually still in the order of less than 1 second. Therefore, for the two datasets taken as case studies the computational loads of the proposed method are still acceptable. However, care should be given to the treatment of large datasets where further research should be directed to speed-up the calculations of variance-covariance matrices. Thus, the promising use of the MDKS method can also be implemented faster in dealing with large amount datasets.

5. Conclusions

This paper presents a modified Kennard-Stone algorithm to perform data splitting for developing ANN models. The method, namely MDKS, employs data division algorithm that considers variability of dataset in both *x*- and *y*- spaces and uses Mahalanobis distance as selection criteria. The results from experimental studies show that this improved method is not only capable of improving the performance of the ANN model but also it outperforms both standard KS and SPXY methods. In terms of computational load, however, there might be issues of using this improved method for dealing with large amount datasets. Nevertheless, the

improved method can be utilised to perform better data partition for better development of ANN models. Further research on how to speed-up the calculations of variance-covariance matrices within the Mahalanobis distance would surely be the next step to further refine this method for its faster implementation in dealing with huge amount of dataset.

References

Atkinson, A.C. 1995. Beyond response surfaces: recent developments in optimum experimental design. Chemometrics and Intelligent Laboratory Systems **28**, 35-47.

Bowden, G.J., Maier, H.R., & Dandy, G.C. 2002. Optimal data division for neural network models in water resources applications. Water Resources Research **38** (2), 1-11.

de Aguiar, P.F., Bourguignon, B., Khots, M.S., Massart, D. L., & Phan-Than-Luu, R. 1995. D-optimal designs. Chemometrics and Intelligent Laboratory Systems **30**, 199-210.

Flood, I., & Kartam, N. 1994. Neural networks in civil engineering. I: Principles and understanding. Journal of Computing in Civil Engineering **8** (2), 131-148.

Galvão, R.K.H., Araujo, M.C.U., José, G.E., Pontes, M.J.C., Silva, E.C. & Saldanha, T.C.B. 2005. A method for calibration and validation subset partitioning. Talanta **67**, 736-740.

Hatch, J.R., Bullock Jr., J.H., & Finkelman, R.B. 2006. Chemical Analyses of Coal, Coal-Associated Rocks and Coal Combustion Products Collected for the National Coal Quality Inventory. http://pubs.usgs.gov/of/2006/1162/ (retrieved on 3 April 2008).

Kennard, R.W. & Stone, L.A. 1969. Computer Aided Design of Experiments. Technometrics **11** (1), 137-148.

Kocjančič, R., & Zupan, J. 2000. Modelling of the river flowrate: the influence of the training set selection. Chemometrics and Intelligent Laboratory Systems **54**, 21-34.

Maesschalck, R.D., Jouan-Rimbaud, D., & Massart, D.L. 2000. The Mahalanobis Distance. Chemometrics and Intelligent Laboratory Systems **50** (1), 1-18.

Maier, H.R., & Dandy, G.C. 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. Environmental Modelling & Software **15**, 101-124.

Maier, H.R., Jain, A., Dandy, G.C., & Sudheer, K.P. 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. Environmental Modelling & Software **25**, 891-909.

Minns, A.W., & Hall, M.J. 1996. Artificial neural networks as rainfall-runoff models. Hydrological Sciences Journal **41** (3), 399-417.

Rajer-Kanduč, K., Zupan, J., & Majcen, N. 2003. Separation of data on the training and test set for modelling: a case study for modelling of five colour properties of a white pigment. Chemometrics and Intelligent Laboratory Systems **65**, 221-229.

Saptoro, A., Yao, H.M., Tadé, M.O., & Vuthaluru, H.B. 2008. Prediction of coal hydrogen content for combustion control in power utility using neural network approach. Chemometrics and Intelligent Laboratory Systems **94**, 149-159.

Shahin, M.A., Maier, H.R., & Jaksa, M.B. 2004. Data division for developing neural networks applied to geotechnical engineering. Journal of Computing in Civil Engineering **18** (2), 105-114.

Wu, W., Walczak, B., Massart, D.L., Heuerding, S., Erni, F., Last, I.R., & Prebble, K.A. 1996. Artificial neural networks in classification of NIR spectral data: Design of the training set. Chemometrics and Intelligent Laboratory Systems **33**, 35-46.