Input Variable Selection in Modelling of Desulphurization Efficiency

Riku-Pekka Nikula, Esko Juuso, Kauko Leiviskä

Control Engineering Laboratory, P.O.Box 4300, FI-90014 University of Oulu, Finland
(e-mail: riku-pekka.nikula@oulu.fi.)

Abstract: Several methods are applied to find the input variables with predictive power to the degree of desulphurization modelling. The methods are applied on the data from a desulphurization plant processing flue gases coming from a coal-fired power plant. In non-linear and complex industrial processes, the nature of the relationships between the variables may be vague and a functional model based on a physical interpretation of the process may be difficult to define. Data-driven statistical modelling approaches are, therefore, reasonable alternatives. However, such models may become corrupted due to the inclusion of uninformative, weakly informative or redundant variables. Linear correlation coefficients, principal component analysis and regression, partial least squares regression, mutual information based algorithms and the general regression neural network are tested in the selection of the informative variables. The results obtained are relevant to desulphurization plant monitoring development.

Keywords: artificial neural networks, desulphurization plant, input variable selection, machine learning, modelling, process monitoring

1. INTRODUCTION

In model development, the preliminary assumption is that one or several candidate variables are capable of describing some of the output behaviour. The input variable selection task is common to the development of all statistical models. It depends on the discovery of relationships within the available data. In the case of parametric, or semi-parametric empirical models, the difficulty of the input variable selection task is somewhat alleviated by the a priori assumption of the functional form of the model, which is based on some physical interpretation of the underlying system or process being modelled (May et al. 2011). Although such a model is theoretically the most accurate, it may be difficult to develop. Data-driven statistical modelling approaches do not have an assumption regarding the model structure. Instead, the model is developed after the variable selection or the variables are selected simultaneously during the model training.

Several aspects impact on the formation of an optimal input set. First of all, d potential inputs form 2^d-1 input subsets. The testing of all the subset combinations with a large d requires efficient algorithms. Including more inputs in a model increases the computational burden of the model. This is further exacerbated in time series studies, in which appropriate lags must be chosen. As the lag of input time series increases, so does the number of inputs to the model and consequently the memory requirement of the model increases. According to 'the curse of dimensionality' by Bellman (1961), the linear increase in the dimensionality of the model results in the total volume of the modelling problem domain increasing exponentially. Moreover, understanding complex models is more difficult than understanding simple models that give comparable results. Inclusion of redundant and irrelevant input variables worsens the training of the models – especially artificial neural networks (ANNs). Redundant variables increase the number of local minima in the error function that is projected over the parameter space of the model (Bowden et al. 2005; May et al. 2011). Irrelevant variables add noise into the model inducing misconvergence and poor model accuracy. The most important characteristic of the input set is the inclusion of predictive power. In conclusion, the optimal input variable set has the fewest input variables needed to describe the behaviour of the output, with minimum redundancy and without uninformative variables.

Using analytical methods to define an optimal input set evidently has advantages. However, a unifying theoretical framework is lacking (May et al. 2011). The approaches are diverse, but can be broadly classified into three main classes: wrappers, filters and embedded methods (Guyon and Elisseeff, 2003). Wrappers approach the task as part of the optimisation of model architecture. The optimisation searches through the input combinations and selects the set which yields the optimal generalisation performance of the trained learning machine. Embedded methods perform variable selection in the process of training and are usually specific to given learning machines. Filters distinctly separate the variable selection task from the specific learning machine. Filters use statistical analysis techniques to measure the relevance of individual, or combinations of, input variables. The approach provides a generic selection of variables, not tuned for the specific learning machine. The approach can be also used as a pre-processing step to reduce space dimensionality and overcome overfitting. Sophisticated wrappers and embedded methods improve predictor performance compared with simple variable ranking methods, but the improvements are not always significant (Guyon and Elisseeff, 2003). Wrappers and filters require a criterion or test to determine the influence of the selected input variable or variables and a strategy for searching among the combinations of candidate variables (May et al. 2011).

In this study, data from the desulphurization plant of a coal-fired power plant is analysed. Process systems generally contain varying degrees of non-linearity. Consequently, the presumption is that the process model should be non-linear although linear parts could be involved in the plant behaviour. Because of this, Artificial Neural Networks (ANNs) which are capable of modelling non-linear relationships give a good premise for modelling. ANN architectures can be built with arbitrary flexibility and can be successfully trained using any combination of the input variables which are good predictors. The model-free input variable selection approach – implying the filters – is considered here. The linear relationships of the candidate variables to the response variable – degree of desulphurization – are analysed with cross-correlations and partial correlation. Dimensionality reduction is performed by forming the linear combinations of the original variables by using Principal Component Analysis (PCA) and Partial Least Squares (PLS) regression. To get a grasp of the non-linear relationships among the variables, Mutual Information (MI) based criteria are used. In addition to all the original variables, the selections produced by correlation analyses are used as inputs to PCA and PLS regression; the selections from correlation analyses and the input scores produced by PCA and PLS regression are used as inputs to the evaluation of the mutual information based criteria. To obtain a generalized impression of the performance of all the formed input variable sets, General Regression Neural Networks (GRNNs) are trained. This type of neural network is chosen, because it can be designed very quickly. PLS regression models are tested for comparison. The next Section explains the used methods and the process being analysed. The main results are presented and discussed thereafter.

2. METHODS AND MATERIALS

2.1 Linear and Non-linear Filters

Filter algorithms typically measure relevance and optimality criteria that are used to discover the important input variables. Incremental search strategies tend to dominate the filter approaches, because the relevance measure is typically bivariate statistic of the candidate-output relationship. Each of the relationships is evaluated. Currently, two broad classes of filters have been considered: those based on linear correlation; and those based on information theoretic measures, such as mutual information (May et al., 2011).

Input variable ranking based on the Pearson correlation is one of the most widely used methods. The candidate variables are sorted by the order of decreasing correlation and the selection is based on greedy selection of the first k variables, or upon all variables having correlation significantly different from zero. The method is classed as a maximum relevance filter and only the interactions between each candidate and the output is considered. The Pearson correlation, R_{xy}, is defined by

$$R_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

where x_i, y_i $\bar{x}$, $\bar{y}$ and n are the candidate variable, the target variable, the corresponding mean values and the total number of observations, respectively. In (1), the numerator is simply the sample covariance; and two terms in the denominator are the square root of the sample variances.

If the candidate variables are themselves correlated, redundancy is an important issue. In such a case, the correlation ranking approach is likely to select too many variables, since many candidates will each provide the same information regarding the target variable. Given three variables x, y and z, the partial correlation measures the correlation between x and y after the relationship between y and z has been discounted. The partial correlation $R_{xy\cdot z}$ can be determined from the Pearson correlation using

$$R_{xy\cdot z} = \frac{R_{xy} - R_{xz}R_{yz}}{\sqrt{\left(1 - R_{xz}^2\right)\left(1 - R_{yz}^2\right)}}. \tag{2}$$

The limitations of linear correlation analysis have created interest in alternative statistical measures of dependence, which are more adept at identifying and quantifying dependence that may be chaotic or non-linear; and which may therefore be more suitable for the development of ANN models (May et al. 2011). Mutual information is a measure of dependence that is based on information theory and the notion of Shannon's (1948) entropy, and is determined by the equation

$$I(x;y) = \iint p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy, \tag{3}$$

where $p(x)$ and $p(y)$ are the marginal probability density functions of x and y, respectively; and $p(x,y)$ is the joint (bivariate) density. $I(x;y)$ denotes the mutual information, which is a measure of dependence between the density of the variable x and the density of the target y. Mutual information measures the quantity of information about a variable y that is provided by a second variable x. The advantage of mutual information over linear correlation is that it is based solely on probability distributions within the data and is therefore an arbitrary measure, which makes no assumption regarding the structure of the dependence between variables (May et al., 2011). The difficulty is that the densities $p(x)$, $p(y)$ and $p(x,y)$ are all unknown and hard to estimate from data. The case of continuous variables is the hardest. One can consider quantizing the variables or approximating their densities with a non-parametric method such as Parzen windows (Guyon and Elisseeff, 2003).

In feature selection literature, there are several filters using a variety of heuristic criteria based on mutual information. Current best practice has been to hand-design the criteria, augmenting the individual feature relevance with various penalties to manage the feature redundancy (Brown, 2009). Brown (2009) offers a descriptive "top-down" framework, showing that several heuristic criteria in the literature can be expressed in a common functional form

$$J = I(x_n; y) - \beta \sum_{k=1}^{n-1} I(x_n; x_k) + \gamma \sum_{k=1}^{n-1} I(x_n; x_k \mid y), \tag{4}$$

where β and γ are configurable parameters varying in [0,1].Variable x_n is the n^{th} variable being evaluated; x_k represents the already selected variables; and y is the target variable. The first term $I(x_n;y)$ ensures feature relevance (mutual information); the second term with the parameter β penalises high correlations (redundancy) between variable itself and the existing variables; the third term with the parameter γ depends on the class conditional probabilities. Brown (2009) has identified 12 separate criteria that can be described within this framework; four of them are tested in this study. Mutual Information based Feature Selection (MIFS) criterion by Battiti (1994) includes the relevance and redundancy but omits the conditional term. Maximum-Relevance Minimum-Redundancy (MRMR) criterion by Peng et al. (2005) takes the mean of the redundancy term, but omits the conditional term. Joint Mutual Information (JMI) criterion by Yang and Moody (1999) has all the three terms and can be defined by (5). Conditional Mutual Information Maximization (CMIM) by Fleuret (2004) can be defined by (6):

$$J_{jmi} = I(x_n; y) - \frac{1}{n-1} \sum_{k=1}^{n-1} \left[I(x_n; x_k) - I(x_n; x_k \mid y) \right], \tag{5}$$

$$J_{cmim} = I(x_n; y) - \max_k \left[I(x_n; x_k) - I(x_n; x_k \mid y) \right]. \tag{6}$$

2.2 Principal Component Analysis and Regression

Principal component analysis (PCA) is a commonly adopted technique for reducing the dimensionality of a dataset X. PCA achieves dimensionality reduction by expressing the d variables ($x_1,\dots,x_d$) as k feature vectors named principal components (PCs). Mathematically, PCA relies on an eigenvector decomposition of the covariance or correlation matrix of the process variables. PCA is scale-dependent, and therefore, it is conventional to adjust the variables to zero mean and unit variance. PCA decomposes the data matrix X as the sum of the outer product vectors t_i and p_i plus a residual matrix E (Wise and Gallagher, 1996):

$$X = t_1 p_1^T + t_2 p_2^T + \dots + t_k p_k^T + E\ . \tag{7}$$

The t_i vectors are known as scores and contain information on how the samples relate to each other. The scores form an orthogonal set ($t_i^T t_j = 0$ for $i \neq j$). The p_i vectors are known as loadings and p_i are eigenvectors of the covariance matrix. The t_i, p_i pairs are arranged in descending order according to the associated eigenvalue λ_i. The first pair captures the largest amount of variation in the data that is possible to capture with a linear factor. Each subsequent pair captures the greatest possible amount of variance remaining at that step. Usually, it is not practical to compute all the k eigenvectors, since most of the variability in the data is typically captured in the first few PCs. A common selection method is to choose all the PCs whose eigenvalues exceed some threshold λ_0, or generate a plot of the cumulative eigenvalue as a function of the number of PCs so that the desired amount of variance is explained (May et al. 2011).

Alternatively, Principal Component Regression (PCR) can be used to select the PCs. Then, multiple linear regression models are built based on response variables Y and varying number of PCs explaining X. Selection of the components can be done by choosing the PCs that optimize the predictive ability of the model (Wise and Gallagher, 1996). Typically, the available data is divided into training and validation sets. The residual error of prediction on the validation samples is determined as a function of the number of PCs. In k-fold cross-validation, the original set is randomly partitioned in k subsamples. Thus, in *10*-fold cross-validation 90 % of the data is used to train the model and 10 % is used in validation. Each subsample is used exactly once as the validation data.

2.3 Partial Least Squares Regression

Partial Least Squares (PLS) regression extracts latent variables that explain the variation in the predictor variables X and the variation in X which is the most predictive of the response variables Y. In other words, PLS attempts to find factors that are correlated with Y while describing a large amount of the variation in X. As a point of comparison, in PCR the components solely explain the variance in X. As in PCA, the latent vectors or scores (t_1, t_2...) are orthogonal. The selection of components can be done like in PCA and PCR. In addition to choosing the components that explain the most variance in X, the components that explain the most variance in Y can be chosen. See Geladi and Kowalski (1986) for more detailed information on PLS regression.

2.4 General Regression Neural Networks

Developed by Specht (1991), the general regression neural network (GRNN) is a supervised feedforward artificial neural network. It uses a nonparametric estimate for the probability density function of the data. Non-linear relationships between inputs and output can be modelled with a GRNN. The network architecture is fixed which means that multiple models do not need to be trained to optimise the network architecture. It has only a single parameter, the kernel bandwidth, which needs to be learned during training. The parameter is named 'spread' hence. Training is much faster than with other artificial neural networks, such as Multi-Layer Perceptrons (MLPs) trained using the backpropagation algorithm. The GRNN uses memory based (lazy) learning, and therefore it has an increased memory requirement to store the training data and a greater computational requirement when querying the network than an MLP. Further information about the method can be found in Specht (1991).

2.5 Performance Criteria

Three criteria are adopted for assessing the models developed. The popular measure of predictive performance is the mean squared error (*MSE*). Another statistical error measure is the mean absolute error (*MAE*). Goodness-of-fit can be evaluated with the coefficient of determination (r^2). The drawback of these criteria is that the best result does not necessarily mean an optimal model. Models with large number of input variables tend to be biased as a result of overfitting. In (8), (9), and (10), y_j, $\bar{y}$, $\hat{y}_j$, $\tilde{y}$, and n are the observed value, the mean of the observed value, the corresponding predicted value, the mean predicted value and the total number of observations, respectively. The criteria are expressed as:

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2 \;, \tag{8}$$

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \;, \tag{9}$$

$$r^2 = \frac{\sum_{j=1}^{n}\left(y_j - \bar{y}\right)\left(\hat{y}_j - \tilde{y}\right)}{\sqrt{\sum_{j=1}^{n}\left(y_j - \bar{y}\right)^2 \sum_{j=1}^{n}\left(\hat{y}_j - \tilde{y}\right)^2}} \;. \tag{10}$$

2.6 Desulphurization Process and the Data

The coal-fired Salmisaari power plant consists of two main units for energy production and a desulphurization plant, which processes the flue gases from both main units. Boiler 1 is a combined heat and power plant with the capacity of 160 MW_{el} and 300 MW_{th} and boiler 7 is a heat unit with the capacity of 180 MW_{th}. The efficiency of nearly 90 % is reached in combined heat and power generation.

The desulphurization plant consists of two parallel reactors which process the flue gases. Fig. 1 demonstrates the principle of operation. Flue gases from the furnace of the steam boiler of the power plant arrive at the electrostatic precipitator ('preliminary separator') which separates fly ash. Flue gases without any solid particles come to the reactor from above and are mixed with lime sludge using compressed air. The particles of sludge and sulphur dioxide molecules are partly mixed in the reactor and reaction products fall at the bottom of the silo. Reaction continues in bag filters, in which gases flow through textile tube and 99.7 % of solid particles remain on its walls. Purified gases go via the fans into the chimney and out in the air. Middle product accumulating at the bottom of the reactor is used to produce sludge, and necessary amount of end product from below the filter is added to it. Together with water, these form the base of sludge. Lime is added in the form of lime milk to achieve the desired level of desulphurization. The amount of sludge pumped into the reactor is controlled so that all water in sludge evaporates and flue gas going to the filter is dry. The method is called half-dry, because the chemical reaction occurs partly in the wet, partly in the dry state. The outgoing end product is used for earth works such as filling ditches, strengthening man-made hills or under the dumping areas.

The variable y for the degree of desulphurization is formed from the similarly standardized SO_2 concentration measurements from the flue gas before (SO_2^{in}) and after (SO_2^{out}) the desulphurization plant. Therefore, the equation $y = 1 - SO_2^{out} / SO_2^{in}$ was considered proper for this study. Table 1 shows some characteristics of the used data. One hour average data is used. Data sets A and B represent the typical operation of the plant; data sets C and D represent a campaign during which higher than the typical degree of desulphurization was used. The set E is a combination of A, B and D. 1686 hours were removed from the set due to memory overflows during the training of GRNNs with the full length data. Thus, the final length of E was 3000 hours. The term 'std' is standard deviation. The candidates for a model input set include 66 variables. The variables that are presented in Section 3 are described in Table 2. The set included some computational variables. With y and other computational variables, it has to be noticed that the uncertainty of measurement is cumulative. The values produced by the computations are less accurate than the single original measurement values. However, the measurement uncertainty is not considered in this study any further. Equations for x_{57}, x_{58}, x_{64}, and x_{65} are presented in (11), (12), (13), and (14):

$$x_{57} = \frac{x_1 \cdot x_3}{\frac{\sqrt{x_6}}{\sqrt{x_6} + \sqrt{x_7}} \cdot x_{60} \cdot x_8} \;, \tag{11}$$

$$x_{58} = \frac{x_2 \cdot x_3}{\frac{\sqrt{x_7}}{\sqrt{x_6} + \sqrt{x_7}} \cdot x_{60} \cdot x_8} \;, \tag{12}$$

$$x_{64} = \frac{(x_1 + x_2) \cdot x_3}{x_{52} + x_{53}} \;, \tag{13}$$

$$x_{65} = x_8 / x_{14} \;. \tag{14}$$

Table 1. Description of the used data

set	year/month	length	degree of desulph. (%)			
		(hours)	mean	std	max	min
A	2009/12 - 2010/5	2801	69.2	6.1	91.9	32.0
B	2009/10-12	1401	71.5	7.6	91.6	51.0
C	2012/1-3	969	81.4	4.0	92.1	67.4
D	2012/3	484	87.2	2.1	96.0	80.0
E	(part of A)+B+D	3000	74.9	8.2	96.0	51.0

Table 2. Candidate variables

variable	*explanation*
x_1	sludge volume flow to reactor 1
x_2	sludge volume flow to reactor 2
x_3	sludge density in feeding tank
x_6	pressure over reactor 1 inlet duct
x_7	pressure over reactor 2 inlet duct
x_8	SO_2 emission measurement before desulphurization plant
x_9	Power plant power output
x_{11}	H_2O in B chimney
x_{14}	feeding tank level
x_{16}	mixing water volume flow
x_{17}	mixing tank density
x_{20}	O_2 after electrostatic precipitator 1
x_{22}	flue gas temperature before desulphurization plant
x_{23}	flue gas pressure before desulphurization plant fan
x_{24}	flue gas pressure after desulphurization plant fan
x_{27}	circulating dust silo level
x_{28}	raw water volume flow to feeding tank
x_{32}	NO_x in B chimney
x_{33}	NO_x into A chimney
x_{36}	pressure difference over reactor 1
x_{37}	pressure difference over reactor 2
x_{38}	pressure difference over bag filters 1
x_{42}	O_2 in boiler 1 flue gas
x_{44}	CO_2 in boiler 1 flue gas
x_{45}	pressure after electrostatic precipitator from unit 7
x_{52}	SO_2 mass flow from unit 1
x_{53}	SO_2 mass flow from unit 7
x_{57}	'ratio of sludge and SO_2 into reactor 1'
x_{58}	'ratio of sludge and SO_2 into reactor 2'
x_{59}	'sum of sludge mass flows to reactors 1 and 2'
x_{60}	'sum of flue gas flows from units 1 and 7'
x_{62}	flue gas temperature out of reactor 2
x_{63}	SO_2 mass flow to desulphurization plant
x_{64}	'ratio of sludge flow and SO_2 mass flow'
x_{65}	'ratio of SO_2 before desulph. plant and feeding tank level'

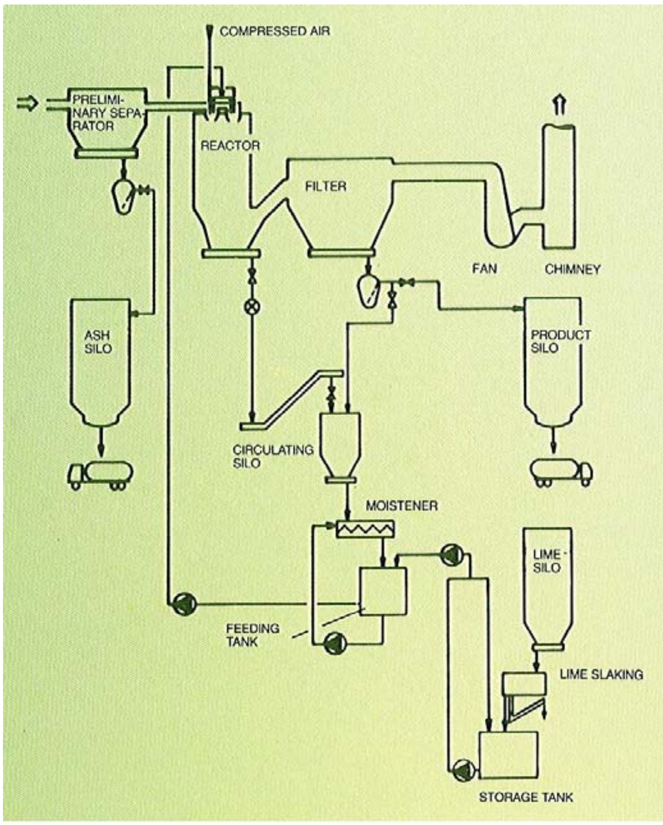

Fig. 1. The desulphurization plant with half-dry method (Helsinki Energy Board, 1980s).

3. RESULTS

In this Section, the mentioned methods are applied to the Salmisaari data. All the studies were performed using Matlab (version 7.12) software (Mathworks Inc., Natick, MA, USA, 2011). All the methods were applied on data normalized to zero mean and unit variance. The main results are presented and discussed.

3.1 Correlations

Correlation coefficients were studied with data sets A and C. Only the coefficients to the response variable y were studied. Cross-correlations were used to find the 24 hour lags. Data was normalized so that auto-correlations at zero-lag were identically 1.0. The results with zero-lag correspond to the output of (1). Data was modified with logarithm of the absolute value, square, inverse, and square root of the absolute value. To define strong correlations two limits were made: Limit I was $|R_{xy}|>2/\sqrt{d}$, and limit II was $|R_{xy}| > 0.5$. The parameter d is the number of candidates. Table 3 shows the variables, of which correlations were larger than the limit II. The column with the title 'zero-lag' indicates the correlation without any modifications or lag. The column with the title 'the largest' indicates the largest correlations using the modified data. The column 'modification' shows the lag in hours and the method used to modify the data.

Forty-nine variables measured up to the limit I with the set A, whereas the corresponding number was 43 with the set C. The variables that measured up to the limit I were tested with partial correlation in such a way that z included all the other variables that measured up to the limit I except the variable being tested. Using partial correlation, only a single variable x_{63} measured up to the limit I. The partial correlation $R_{x_{63}y \cdot z}$ was -0.333 with the set A, and -0.602 with the set C.

Table 3. The largest absolute correlations

set A	zero-lag	the largest	modification
x_8	0.617	0.617	0
x_{11}	-0.640	-0.643	-1, *log(|x|)*
x_{17}	-0.572	-0.572	0
x_{27}	0.543	0.584	-14, x^2
x_{33}	0.611	0.614	-3, x^2
x_{44}	-0.711	0.720	0, x^{-1}
x_{45}	-0.280	-0.690	-13, *log(|x|)*
x_{63}	-0.598	-0.619	-1, x^2
x_{65}	0.610	0.610	0
set C	**zero-lag**	**the largest**	**modification**
x_{16}	-0.376	-0.511	-1, x^2
x_{20}	-0.444	0.505	-23, x^{-1}
x_{23}	0.479	-0.502	-2, x^2
x_{44}	-0.571	-0.724	-11, *log(|x|)*
x_{52}	-0.819	-0.825	0, x^2
x_{53}	-0.637	-0.637	0
x_{62}	-0.505	0.506	-21, x^{-1}
x_{63}	-0.816	-0.816	0
x_{64}	0.639	-0.783	0, x^{-1}

3.2 Principal Component Analysis and Partial Least Squares Regression

All the variables and only the variables that measured up to the correlation limits I and II were fed to PCA and PLS regression. PCA produced the same number of principal component scores which was the number of input variables. To reduce the dimensionality, the components were chosen based on the variance that they explain from the total variance of the input space. The variance of the components that explain most were summed up until the set criterion was measured up. In this study, the first limit was 80 %, the second was 95 %, and the third was 99 %. Table 4 shows the number of components explaining the defined variance using different data sets. On the sets A and B correlation selections from the set A were used, and on the sets C and D the selections from C were used. The percentage of variance explained in y was considered with PLS regression. The same limits were used and results are shown in Table 4.

An alternative way to determine the proper amount of components was to develop regression models to the prediction of the response y. The models were built with data sets A, B, C, and D so that the amount of components varied from one to thirty. All the variables and only the variables that measured up to the correlation limit I were used. The models were validated with *10*-fold cross-validation. The mean squared error performance was evaluated based on the difference between estimated $\hat{y}$ and actual y. The results from principal component regression validation are shown in Fig. 2. The y axis values correspond to the mean observed.

Table 4. The number of components explaining the defined variance in *X* with PCA (left) and *y* with PLS regression (right)

variance limit	A	B	C	D	variables
80	8|3	8|4	11|4	8|5	all
95	19|9	18|8	24|8	20|7	all
99	29|28	28|22	35|13	29|15	all
80	4|2	6|2	6|3	5|3	cor. limit I
95	13|5	14|4	15|6	13|6	cor. limit I
99	23|14	22|12	25|11	20|18	cor. limit I
80	4|1	4|2	3|1	3|2	cor. limit II
95	6|2	7|2	6|2	6|5	cor. limit II
99	8|3	8|3	8|4	7|6	cor. limit II

The linear regression models performed relatively well taking into consideration the weak linear relationships of the variables to the response variable. *MSE* decreases as the number of components increases with some exceptions. *MSE* was below 0.43 using five components with all the data sets. Apart from the set D, *MSE* was below 0.28 with eight components. With sets A and B the standard deviations of *MSE* were low and decreasing as the number of components increased. The set C had the lowest standard deviation of MSEs with four to five components. Inclusion of components seemed to increase the standard deviation. The set D had lower than 0.1 standard deviations the first time with 16 components. The results show that using a large number of components will generally do a good job in fitting the current observed data. On the other hand, it is a strategy that can lead to overfitting and can give an overly optimistic estimate for the expected error. Fig. 2 indicates that achieving a quite constant *MSE* level needs three to seven components, which can be interpreted as a proper amount of input components for a principal component regression model.

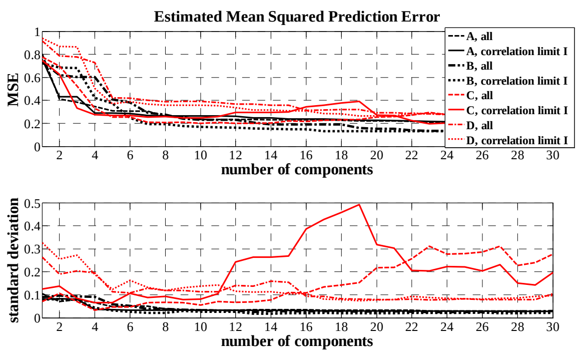

Fig. 2. Principal component regression validation results.

Fig. 3 shows corresponding results for PLS regression. The *MSE* reaches a relatively low level when the model has three or four components, and the level stays quite constant when the amount of components is increased. Also the standard deviation of MSE was quite constant after the fourth component apart from the C set. In general, the performance of PLS regression was somewhat better than PC regression using a small number of components. The results indicate that a proper amount of components for a PLS regression model is slightly smaller than for a PC regression model.

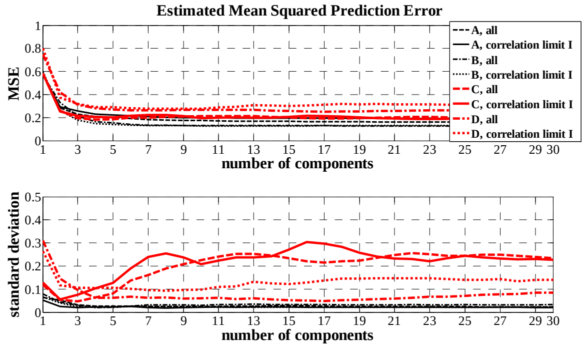

Fig. 3. Partial least squares regression validation results.

3.3 Mutual Information Based Criteria

All the variables, PCA scores, and PLS regression input scores together with the response variable were analysed with the mutual information based criteria using data sets A, B, C, and D. PCA and PLS regression scores were done using all the candidates. The procedure greedily chose the variable with the largest incremental gain until the chosen number of variables was reached. The number of the input variables to be selected was chosen to be 20 variables. The continuous variables were quantized for the mutual information calculations. The data of each variable were quantized in the first case to 10 bins and in another case to 100 bins so that the bin limits were quantiles with a regular interval. In MIFS, the heuristic weighting β was chosen to be 0, 0.5, and 1 in different tests. When $\beta = 0$, the method is reduced into mutual information ranking. In conclusion, $3\cdot4\cdot2\cdot6=144$ input sets were created. Because of limited space, Table 5 shows only the largest mutual information of ten first variables chosen from sets A and C together with the set B (10 bins), which eventually led to the most promising modelling result.

Table 5 shows that the quantization has a large effect on the result. With all the tested criteria, the first selected variable was always the same on the same data set, but could differ based on the quantization as Table 5 indicates with C. The subsequent variables varied based on the criteria used. This arises from the varying emphasis the different criteria have on the redundancy and conditional terms of (4). Obviously, the varying operational states of the different sets have an influence in the varying selections of input variables. The same is seen in correlation rankings, see Table 3.

Table 5. The largest mutual information using sets A, C, and B

set A		set C		set B
10 bins	100 bins	10 bins	100 bins	10 bins
x_{44},0.6773	x_{44}, 2.4090	x_{52},0.9652	x_{65},3.5723	x_{63},0.6883
x_{45},0.6194	x_{8}, 2.3962	x_{64}, 0.9198	x_{8},3.5400	x_{45},0.5371
x_{11},0.5062	x_{65}, 2.3961	x_{63}, 0.8950	x_{59},3.5145	x_{32},0.5012
x_{65},0.4758	x_{45}, 2.3339	x_{53}, 0.7426	x_{64},3.5134	x_{64},0.4798
x_{8},0.4703	x_{11}, 2.3314	x_{44},0.6950	x_{57},3.5018	x_{22},0.4500
x_{33},0.4481	x_{7}, 2.3111	x_{65},0.4806	x_{58},3.5013	x_{44},0.4317
x_{17},0.4285	x_{36}, 2.3033	x_{8},0.4520	x_{9},3.4994	x_{53},0.3935
x_{42},0.4210	x_{37}, 2.2976	x_{24},0.4336	x_{62},3.4990	x_{52}, 0.3879
x_{63},0.4038	x_{52}, 2.2929	x_{42},0.4206	x_{24},3.4916	x_{23},0.3716
x_{27},0.3916	x_{62},2.2799	x_{28},0.4193	x_{38},3.4909	x_{65},0.3690

3.4 Evaluation of the Selected Inputs in Modelling

Evaluation of the chosen sets was done with a heuristic trial-and-error approach by building GRNNs with different input sets. In GRNNs, spread values 0.1, 0.5, 1, 2, and 10 were tested, but an optimal value for the spread value was not searched for. To test GRNNs, there were three cases. In the first case, the network was trained with the set A and the performance was tested with the set B. In the second case, the network was trained with the set C and tested with the set D. In the third case, the sets A, B, and D formed the training set E and the set C was the test set. The number of components from PCA and PLS regression were chosen based on the variance limits discussed earlier. Only the number of components defined with the sets A and C were used. Considering correlation, PCA, and PLS regression sets, the selected sets formed from the analysis of A were used to train GRNNs on A set and E set. Similarly, sets formed from the analysis of C were used to train GRNNs on C set and E set. Mutual information criteria based selections were tested with all the 20 selected variables and with only ten variables which were chosen first by each criterion. All of these selections were used on each training set A, C, and E. Sets formed based on linear correlations included zero lagged variables without any modifications. GRNNs were trained with every input variable set, spread value, and training set (A, C, and E) separately. Therefore, the number of trained GRNNs was $2\cdot5\cdot2 + 4\cdot5 + 9\cdot5\cdot2 + 18\cdot5 + 9\cdot5\cdot2 + 18\cdot5 + 144\cdot5\cdot3 + 144\cdot5\cdot3 = 4720$.

The model prediction performance on the test sets was monitored with mean squared error, mean absolute error, and the coefficient of determination. Table 6 shows the best results of the GRNN performance on the test sets with the input sets formed with the different methods. From the test sets B and D only the better result is shown. The best result at the last row of Table 6 was achieved with MIFS ($\beta = 0$) from the analysis of the set B. All the presented mutual information criteria based results in Table 6 were reached with data quantized into ten bins.

The best performing neural networks were obtained with input sets that were formed by mutual information based criteria using data quantized into ten bins. Obviously, the quantization into 100 bins was not a proper method. Mutual information criteria based sets performed better with ten than twenty variables indicating that the amount of input variables is highly important part of the selection. A neural network with PLS regression input scores as inputs performed poorly. Variables chosen by the linear correlation produced slightly better performing models than the models with PCA scores as inputs.

Table 6. General regression neural network prediction performance without validation

method	*MSE*	*MAE*	*r²*	*spread*	*test set*
correlation	0.54	0.58	0.46	1	B
PCA	0.54	0.59	0.46	1	B
PLS regression	0.72	0.63	0.28	0.1	D
MI-based (20 var.)	0.42	0.49	0.58	1	B
MI-based (10 var.)	0.36	0.46	0.64	1	B
correlation	0.44	0.51	0.56	1	C
PCA	0.52	0.54	0.48	1	C
PLS regression	0.92	0.76	0.07	0.1	C
MI-based (20 var.)	0.36	0.48	0.64	1	C
MI-based (10 var.)	0.30	0.41	0.70	1	C

The performance of the best performing GRNN was compared with PLS regression models, which were built with the same input set. Also the linear correlation sets were tested in PLS regression models for the sake of comparison. In PLS regression models, one to nine latent variables (components) were tested. Training and test sets were composed from the set E, which was randomly partitioned into ten subsamples of the same size; validation set was the set C. *10*-fold cross-validation was performed to validate the results. Table 7 summarises the performance of the models. The values correspond to the mean observed. The corresponding variability is indicated by the standard deviations, which are the values in parentheses. The presented results are chosen so that the validation *MSE* is as small as possible. The PLS regression model with the correlation limit I set in Table 7 has two latent variables; the model with MI-based set has three latent variables. Fig. 4 shows the normalized degree of desulphurization and the predictions by the GRNN and PLS regression models on the validation period C. The predictions do not follow all the peaks or slower changes in the degree of desulphurization; the predictions have some harsh errors as well. Some of these parts are marked by red ellipses in Fig. 4. The general level of the degree of desulphurization can be found promisingly well.

Table 7. *10*-fold cross-validation *MSE*

model	*training*	*test*	*validation*
PLSR (correlation set)	0.2518(0.0033)	0.2574(0.0413)	0.2652(0.0022)
PLSR (MI-based set)	0.2862(0.0027)	0.2888(0.0279)	0.2707(0.0023)
GRNN (MI-based set)	0.2217(0.0023)	0.2468(0.0276)	0.3009(0.0051)

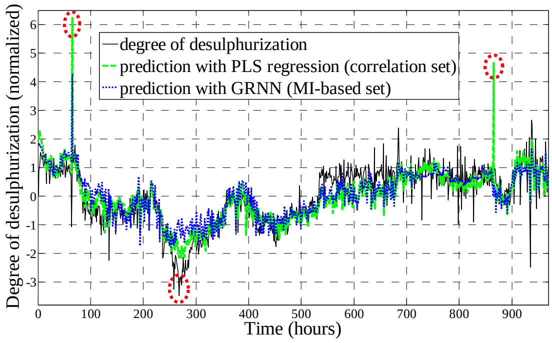

Fig. 4. The degree of desulphurization predicted by PLS regression and GRNN models during the validation period.

4. DISCUSSION

The results indicate that variable selection based on mutual information criteria is the most valid approach from the tested approaches in the considered context. The optimal way to quantize the process variables for the approach needs further research. The fact that industrial processes often involve non-linear behaviour came up in the study. Variables or components chosen by linear methods gave worse non-linear predictions. On the other hand, the linear PLS regression models performed slightly better than the non-linear GRNN model considering the prediction error in validation. However, the GRNN was not optimised by searching for the optimal number of input variables and the optimal value of the kernel bandwidth of the network. The development of the model for the desulphurization benefits from the lags discovered by cross-correlation, because the dynamics of the process need to be taken into consideration.

The use of score vectors from PCA or PLS regression in a neural network was the least promising approach in this study. However, Mohamad-Saleh and Hoyle (2008) use PCA successfully for the elimination of correlated information in the input data of a Multi-Layer Perceptron neural network. Linker (2005) use PCA scores as inputs to a sigmoid feedforward neural network. Lennox et al. (2001) also report the use of PCA and PCR in addition to cross-correlation in analysis of input and output variables for dynamic process models. Li et al. (2007) use PLS regression input and output scores combined with a Radial Basis Function neural network. To compare, only input scores were used as inputs to GRNN in this study.

There are several potential methods not studied here. The use of another mutual information based algorithm, Partial Mutual Information (PMI), for input selection of GRNNs is reported in May et al. (2008) and Bowden et al. (2005). The use of Self-Organizing Maps solely or combined with other methods is reported by (Similä and Laine 2005; Bowden et al. 2005). Laurinen and Röning (2005) report the use of a Bayesian network and expert information in the selection of inputs to a feedforward neural network. The assistance from a process expert is generally reported being valuable in input variable selection (Simula and Alhoniemi, 1999; Lennox et al. 2001; Laurinen and Röning, 2005).

Guyon and Elisseeff (2003) recommend selecting variables in two ways. Firstly, variables should be ranked using a correlation coefficient or mutual information, and secondly, a nested subset selection method performing forward or backward selection or multiplicative updates should be used. This study is in agreement with that. Considering the GRNN, the sets selected with mutual information based criteria performed the best. However, a good model is achieved by fine tuning, where the use of a wrapper or an embedded method seems inviting.

ACKNOWLEDGEMENTS

This research was done as a part of the Measurement, Monitoring and Environmental Assessment (MMEA) programme managed by Cleen Ltd. The authors would like to thank the Finnish Funding Agency for Technology and Innovation (Tekes) for financial support. Helsinki Energy and the personnel are gratefully acknowledged for their contribution.

REFERENCES

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Networks,* volume 5 (4), 537-550.

Bellman, R. (1961). *Adaptive control processes: a guided tour,* Princeton university press, New Jersey.

Bowden, G. J., Dandy, G. C., and Maier, H. R. (2005). Input determination for neural network models in water resources applications. Part 1 – background and methodology. *Journal of Hydrology,* volume 301 (1-4), 75-92.

Brown, G. (2009). A new perspective for information theoretic feature selection. *Proceedings of the twelfth international conference on artificial intelligence and statistic,* volume 5, 49-56.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research,* volume 5, 1531-1555.

Geladi, P., and Kowalski, B. R. (1986). Partial least squares regression: a tutorial. *Analytica Chimica Acta,* volume 185, 1-17.

Guyon, I., and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research,* volume 3, 1157-1182.

Helsinki Energy Board (1980s). An official brochure of the Salmisaari desulphurization plant.

Laurinen, P., and Röning, J. (2005). An adaptive neural network model for predicting the post roughing mill temperature of steel slabs in the reheating furnace. *Journal of Materials Processing Technology,* volume 168 (3), 423-430.

Lennox, B., Montague, G. A., Frith, A. M., Gent, C., and Bevan, V. (2001). Industrial application of neural networks – an investigation. *Journal of Process Control,* volume 11, 497-507.

Li, R., Meng, G., Gao, N., and Xie, H. (2007). Combined use of partial least-squares regression and neural network for residual life estimation of large generator stator insulation. *Measurement Science and Technology*, volume 18 (7). 2074-2082.

Linker, R. (2005). Spectrum analysis by recursively pruned extended auto-associative neural network. *Journal of Chemometrics*, volume 19, 492-499.

May, R. J., Dandy, G. C., Maier, H. R., and Nixon, J.B. (2008). Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. *Environmental Modeling and Software,* volume 23 (10-11), 1289-1299.

May, R., Dandy, G., and Maier, H. (2011). Review of input variable selection methods for artificial neural networks, In Kenji Suzuki (Ed.), *Artificial neural networks - methodological advances and biomedical applications,* 19-44. InTech.

Mohamad-Saleh, J., and Hoyle, B. S. (2008). Improved neural network performance using principal component analysis. *Int. Journal of the Computer, the Internet and Management*, volume 16 (2), 1-8.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* volume 27 (8), 1226-1238.

Shannon, C. E. (1948). A Mathematical theory of communication. *Bell System Technical Journal,* volume 27, 379-423, 623-656.

Similä, T., and Laine, S. (2005). Visual approach to supervised variable selection by self-organizing map. *International Journal of Neural Systems,* volume 15 (1&2), 101-110.

Simula, O., and Alhoniemi, E. (1999). SOM based analysis of pulping process data. *Proceedings of international work-conference on artificial and natural neural networks (IWANN '99),* volume II, 567-577.

Specht, D., F., (1991). A General regression neural network. *IEEE Transactions on Neural Networks*, volume 2 (6), 568-576.

Wise, B.M., and Gallagher, N.B. (1996). The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control,* volume 6 (6), 329-348.

Yang, H., and Moody, J. (1999). Data visualization and feature selection: new algorithms for nongaussian data. *Advances in Neural Information Processing Systems,* volume 12.