Nonlinear Data-driven Process Modelling using Slow Feature Analysis and Neural Networks

Jeremiah Corrigan and Jie Zhang
School of Engineering, Merz Court, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

Keywords: Slow Feature Analysis, Neural Network, Soft Sensor, Dynamic Process Modelling, Data-driven Modelling.

Abstract: Slow feature analysis is a technique that extracts slowly varying latent variables from a dataset. These latent variables, known as slow features, can capture underlying dynamics when applied to process data, leading to improved generalisation when a data-driven model is built with these slow features. A method utilising slow feature analysis with neural networks is proposed in this paper for improving generalisation in nonlinear dynamic process modelling. Additionally, a method for selecting the number of dominant slow features using changes in slowness is proposed. The proposed method is applied to creating a soft sensor for estimating polymer melt index in an industrial polymerisation process to validate the method's performance. The proposed method is compared with principal component analysis-neural network and a neural network without any latent variable method. The results from this industrial application demonstrate the effectiveness of the proposed method for improving model generalisation capability and reducing dimensionality.

1 INTRODUCTION

The advanced process monitoring and control of many industrial processes require robust and reliable models and measurements. The use of hardware sensors to provide measurements can be too costly or the samples may not be frequent enough for acceptable control or monitoring. Soft sensors (Tham et al., 1991) (soft*ware* sensors) provide the alternative to hardware sensors through the use of a process model. Kadlec et al. (2009) provided a comprehensive review of the soft sensor design process, detailing some of the commonly used techniques and associated issues with soft sensor applications. Mechanistic and data-driven modelling approaches make up the two main areas for the development of soft sensors. Figure 1 illustrates the key advantages of data-driven modelling over mechanistic modelling. Mechanistic models are developed by utilising first principle mathematical equations and fundamental scientific and engineering concepts to describe the process. However, many modern processes are highly complex and so the development of mechanistic models can be very expensive and time consuming. Data-driven modelling makes use of process data to create models through a variety of techniques. Data-driven models make use of easy to measure process variables to predict more difficult to measure variables, such as polymer quality variables.

When using data-driven techniques, the complexity of process can determine the necessary technique that is required. In its most basic form, a data-driven model can be produced using linear regression, though this can often lead to poor generalisation in many real world problems, particularly those that display nonlinearities that simply cannot be modelled adequately using linear techniques.

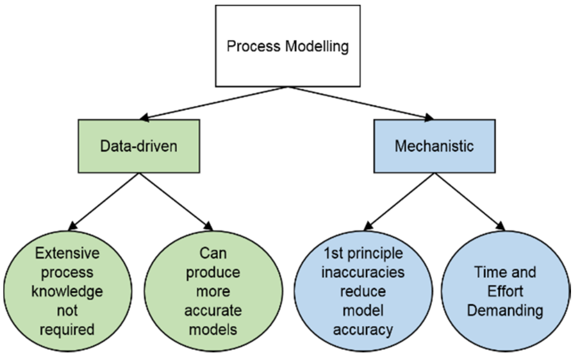
Process Modelling
Data-driven
Mechanistic
Extensive process knowledge not required
Can produce more accurate models
1st principle inaccuracies reduce model accuracy
Time and Effort Demanding

Figure 1: Key differences between data-driven and mechanistic process modelling.

Artificial neural networks are a nonlinear data-driven technique that have been applied to nonlinear

Corrigan, J. and Zhang, J.
Nonlinear Data-driven Process Modelling using Slow Feature Analysis and Neural Networks.
DOI: 10.5220/0007958904390446
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 439-446
ISBN: 978-989-758-380-3

process modelling and control (Bhat and McAvoy, 1990; Chen et al., 1990; Willis et al., 1991) and in particular in the polymerisation industry (Gonzaga et al., 2009; Zhang et al., 2006; Zhang et al., 1997). Bishop (1995) delivered a detailed overview of neural networks and their function. The most common form of neural network is the single hidden layer feed forward network. Although neural networks possess universal approximation capability (Cybenko, 1989), poor performance can still be observed in may applications due to a variety of issues, as discussed by Qin (1997). One of the main issues associated with neural network modelling, and modelling in general, is overfitting. This is where the model performs well on the data it was trained on, but performs poorly on unseen validation data. The ability to perform well on unseen data is known as the generalisation capability. Many techniques have been used to improve generalisation in neural networks, such as regularisation and early stopping (Bishop, 1995), ensemble methods (Breiman, 1996; Yang et al., 2013; Zhang, 1999), and combining neural networks with latent variable techniques.

Using latent variables (LVs) with regression models has been applied before, most commonly with methods such as principal component analysis (PCA) (Jolliffe 2002) and partial least squares (PLS) (Geladi and Kowalski 1986). Principal component regression (PCR) has been applied to soft sensors (Ge et al., 2011; Ge et al., 2014; Hartnett et al., 1998). PCA has also been combined with neural networks (Dong and Mcavoy, 1994). The reason for applying PCA to the data first is to remove collinearity because a neural network model trained on collinear data is only valid when new data follows the same collinearity (Qin, 1997). Thus using PCA on the data before the neural network training can improve generalisation.

Slow feature analysis (SFA) is a technique that extracts slow varying trends from data in the form of LVs known as slow features (SFs) (Wiskott and Sejnowski, 2002). The slowest SFs capture the most important trends, while the fastest mostly represent noise. Applying SFA to process modelling means that underlying dynamics of the process can be captured, as well as offering a de-noising affect when looking at the slower SFs. Additionally, by only selecting a certain number of SFs to be considered for modelling, the dimensionality can be reduced leading to a decrease in model complexity. PCA also produces dimensionality reduction in a similar way by retaining only a certain number of principal components (PCs). By using a reduced number of inputs that express the key trends, combining SFA with data-driven modelling can improve generalisation. This has been demonstrated for soft sensing applications utilising SFA with linear regression (Shang et al., 2015a; Shang et al., 2015b).

However, since many process are nonlinear, using linear regression with SFA will often lead to poor model performance and so combining SFA with neural networks is a promising approach for improving generalisation capability.

A method combining SFA with neural networks for nonlinear dynamic process modelling is proposed in this paper. Dynamic SFA is first applied to the process data and the number of retained (dominant) SFs is selected via inspection of the changes in slowness of the SFs. The dominant SFs are then used as inputs for a single hidden layer feed forward neural network.

The paper is organised as follows: Section 2 describes SFA, Section 3 defines the proposed method and Section 4 presents the application of the proposed method to a soft sensor for polymer melt index in an industrial polymerisation process. Finally, the conclusions of this work are presented.

2 OVERVIEW OF SLOW FEATURE ANALYSIS

Slow feature analysis aims to transform a set of inputs into outputs that are as slowly varying as possible. This allows for the extraction of information that is not overwhelmed by noise and it can reveal underlying dynamics of the inputs.

Wiskott and Sejnowski (2002) defined the optimisation problem, as described below, that enables the extraction of the slow features from the input signals. Given an input vector $x(t)$, the objective is to determine a function $g(x)$ such that the output $y(t)$ varies as slowly as possible, without being constant, so that relevant information can still be extracted, i.e. $y(t) = g(x(t))$.

$$minimise\ \langle \dot{y}_j^2 \rangle \tag{1}$$

subject to three constraints that $\mathbf{y}$ has a zero mean, unit variance and is decorrelated (identity covariance matrix), i.e.

$$\langle y_j \rangle = 0 \tag{2}$$

$$\langle y_j^2 \rangle = 1 \tag{3}$$

$$\langle y_j, y_j \rangle = 0 \tag{4}$$

where $\langle f \rangle$ represents temporal averaging

$$\langle f \rangle = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} f(t)\, dt \tag{5}$$

The first two constraints are included so that the solution of a constant **y** is avoided. Constraint (4) ensures that the extract slow features are decorrelated and so are not duplicates of one another.

2.1 Linear SFA

For the linear case of SFA, the function $g(x)$ is simply a vector of weights W, such that the output $y(t)$ is a linear combination of all of the input variables:

$$y(t) = Wx(t) \tag{6}$$

The optimisation problem can therefore be reduced to the following generalised eigenvalue problem (Shang et al., 2016):

$$\langle \dot{x}\dot{x}^T \rangle W = \langle xx^T \rangle W\Omega \tag{7}$$

where $\langle \dot{x}\dot{x}^T \rangle$ is the covariance matrix of the first order derivative of X, $\langle xx^T \rangle$ is the covariance matrix of X, and Ω is a diagonal matrix of the generalised eigenvalues, which are the optimal objectives of the objective function.

Solving this problem first requires normalising the input signal to zero mean and unit variance. The normalised input, $\boldsymbol{x(t)}$, is then sphered (or whitened) to remove underlying correlations, giving the sphered matrix $\boldsymbol{z(t)}$. The next stage is performing singular value decomposition on the matrix $\langle \boldsymbol{\dot{z}\dot{z}^T} \rangle$:

$$\langle \dot{z}\dot{z}^T \rangle = P\Omega P^T \tag{8}$$

From this, the slow features can be calculated:

$$y(t) = Pz(t) \tag{9}$$

In reality, process data typically has discrete intervals and so the first order derivative can be approximated by a first order difference approximation.

2.2 Dynamic SFA

In many processes, the relationship between the inputs and outputs may involve significant time delay. Dynamic modelling includes inputs from previous sampling times so that prediction performance can be improved.

Dynamic SFA is simply the inclusion of time lagged process inputs into the input signals for SFA. For a given time lag d, the input matrix is as follows:

$$\mathrm{X(t)} = \begin{bmatrix} \mathrm{x(t)} & \cdots & \mathrm{x(t-d)} \\ \vdots & \ddots & \vdots \\ \mathrm{x(t+N-1)} & \cdots & \mathrm{x(t+N-d-1)} \end{bmatrix} \tag{10}$$

where $x(t)$ is a vector of the process inputs at time t and N is the number of samples.

2.3 Selection of Dominant Slow Features

Given a set of m derived slow features, it is necessary to select the number of dominant slow features, M, that best capture the dynamics of the process since many of the faster features represent mostly noise. Including these faster features in model building could decrease generalisation performance because the model is fitting the noise as opposed to the underlying trends. Additionally, decreasing the number of slow features reduces the model complexity, as with other latent variable methods, such as PCA.

There are very few standard procedures for the selection of the dominant slow features but there are some ways this can be accomplished.

2.3.1 Cross Validation Slow Feature Selection

A common method for selecting LVs in general is through cross validation, which has been applied to slow feature regression previously (Shang et al., 2015b). This can work well enough when this is the only parameter to be determined, however, in the case of neural networks, cross validation is often already used for the selection of the number of hidden neurons. Therefore, also using cross validation for slow features selection can lead to poor generalisation, especially for complex processes where it can be difficult to obtain the optimal values for such hyper parameters.

2.3.2 Slowness Criterion based on Reconstruction

Shang et al. (2015c) derived a slowness criterion based on a de-noised reconstruction, suggesting to discard slow features that are faster than all of the input variables. This method gives good model performance when the number of inputs is not too large, however, when using dynamic SFA for a system with a large number of process variables (such as the case study in this work), the number of M slow features calculated is too great and produces poor generalisation.

2.3.3 Slowness Gradient Slow Feature Selection

This work proposes using the changes in slowness of the slow features to determine *M*. This is done by observing the first relatively significant change in the gradient of the sorted eigenvalues, which relate to the Ω matrix from equation 7, and then selecting all of the slow features up until this point. The values of the eigenvalues directly relate to the slowness of the features. Slow feature selection in this way ensures that the selected features carry the most significant slow varying trends without too much noise. Figure 2 provides an example of the trend in eigenvalues. In this case, the significant change in slowness occurs after 11 slow features and so this is the number that would be selected for *M*.

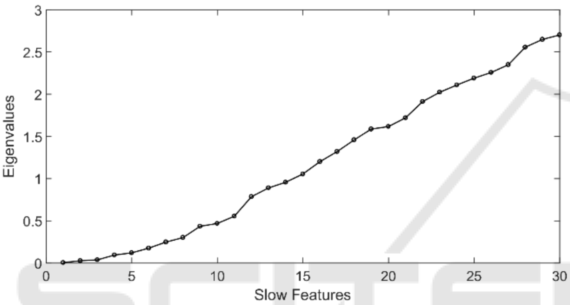

Figure 2: Eigenvalue trend for slow features used for dominant slow features selection.

3 SLOW FEATURE ANALYSIS WITH NEURAL NETWORK METHODOLOGY

A simplified block diagram of the proposed method is illustrated in Figure 3.

The neural networks used are single hidden layer feed forward networks (SLFNN) that were trained using the Levenberg-Marquardt algorithm with regularisation and early stopping. The proposed method utilising SFA and neural networks (SFA-NN) is described in the following steps:

Step 1: Add *d* time lagged inputs to the input data matrix for dynamic modelling, as detailed in Section 2.2.

Step 2: Partition data into a training and testing (TT) set and an unseen validation set.

Step 3: Normalise and sphere TT data and apply SFA as described in Section 2.1 to derive the $\boldsymbol{P}$ matrix and obtain *m* slow features via $\boldsymbol{y_{TT}(t)} = \boldsymbol{Pz_{TT}(t)}$.

Step 4: Select the *M* dominant slow features by the slowness gradient based method as described in Section 2.3.3.

Step 5: Randomly partition this dynamic slow feature TT data into training and testing data sets.

Step 6: The training data is used to train a neural network model for each number of hidden neurons in a given range, e.g. from 1 to 30.

Step 7: Optimal number of hidden neurons is selected by cross validation using the testing data set.

Step 8: The unseen validation data set is normalised and sphered to give $\boldsymbol{z_v(t)}$. Slow features for this data set are calculated based on the previously derived $\boldsymbol{P}$ matrix: $\boldsymbol{y_v(t)} = \boldsymbol{Pz_v(t)}$. These validation slow features are applied to the trained neural network model to assess the model's performance on unseen data.

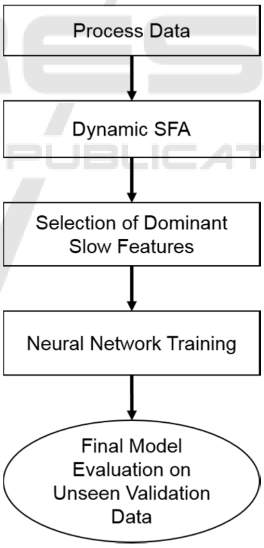
Process Data
Dynamic SFA
Selection of Dominant Slow Features
Neural Network Training
Final Model Evaluation on Unseen Validation Data

Figure 3: Simplified block diagram of the SFA-NN method.

4 CASE STUDY: AN INDUSTRIAL POLYMERISATION PROCESS

4.1 Process Description

The process used for application of the proposed method is a propylene polymerisation process based in China. The data from this process has been used in previous work on data-driven modelling using bootstrap aggregated neural networks (Zhang et al., 2006). Two continuously stirred tank reactors (CSTR) and two fluidised bed reactors (FBR) in series make up the main section of this process, as illustrated in Figure 4. The feed to the first CSTR consists of a catalyst, hydrogen and propylene. The melt index (MI) of polypropylene in the reactor is a key variable of interest in assessing product quality, however, it is difficult to measure and so creating a model to estimate MI from easy to measure process variables could lead to improvements in process monitoring and product quality by providing MI measurements at shorter intervals.

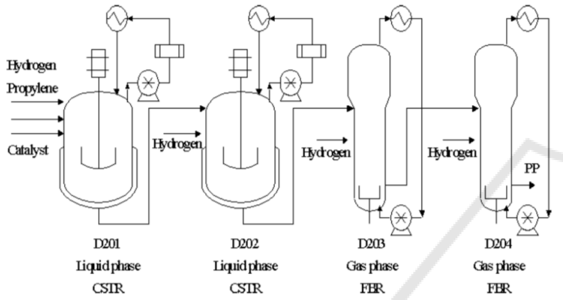

Figure 4: Simplified diagram of a polymerisation process (Zhang et al., 2006).

4.2 Modelling of MI

The data provided covered 31 days and consisted of MI measurements in reactors 1 and 4 that were made every 2 hours, and measurements of 30 process variables that were made every half an hour. All of the process variables are shown in Figure 5. The melt index in reactor 1 is shown in Figure 6. For confidentiality reasons, the units of the variables are omitted.

Table 1: Performance of SFA-NN for training, testing and unseen validation data.

d	Validation MSE / r^2	Testing r^2	Training r^2	*M*
0	**188.8 / 0.9496**	0.9373	0.9483	11
1	**224.3 / 0.9401**	0.9310	0.9461	11
2	**243.7 / 0.9349**	0.9404	0.9474	10
3	1271.5 / 0.6605	0.9433	0.9480	10
4	622.4 / 0.8338	0.9482	0.9623	17
5	2326.7 / 0.3788	0.9425	0.9540	16

Table 2: Performance of PCA-NN for training, testing and unseen validation data.

d	Validation MSE / r^2	Training r^2	Testing r^2	#PCs
0	305.1 / 0.9186	0.9474	0.9229	23
1	636.9 / 0.8300	0.9783	0.9422	37
2	402.6 / 0.8925	0.9749	0.9459	50
3	785.5 / 0.7903	0.9813	0.9476	61
4	1059.5 / 0.7171	0.9943	0.9460	70
5	876.5 / 0.7660	0.9897	0.9363	81

Table 3: Performance of NN for training, testing and unseen validation data.

d	Validation MSE / r^2	Testing r^2	Training r^2
0	287.0 / 0.9234	0.9423	0.9593
1	708.9 / 0.8107	0.9385	0.9756
2	1088.5 / 0.7094	0.9443	0.9856
3	**627.0 / 0.8326**	0.9469	0.9921
4	**471.2 / 0.8742**	0.9400	0.9956
5	**384.0 / 0.8975**	0.9385	0.9815

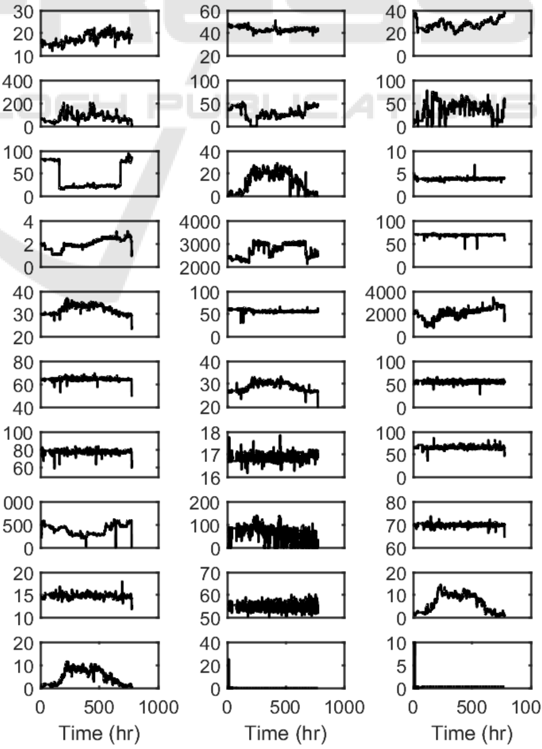

Figure 5: Time series plots of all thirty of the process variables.

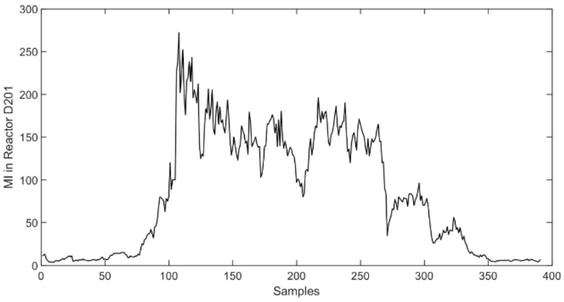

Figure 6: Melt index data for reactor D201.

Often in complex chemical processes such as this, there are time delays between some of the inputs and outputs and so it is necessary to consider time lagged inputs, hence a dynamic model is considered. The form of the input matrix is as follow:

$$X(t) = [\, u_1(t) \dots u_{30}(t), \dots, u_1(t-d) \dots u_{30}(t-d)\,] \tag{11}$$

where X is the input matrix and u_n is the n^{th} process input.

The number of inputs is already large and including time lagged inputs would only increase this significantly. This is where the merit of latent variable methods, which lead to dimensionality reduction, comes in. To better understand and assess the effect of adding in different time lagged inputs, models were created for time lags d from 0 to 5 (representing 0, 0.5, 1, 1.5, 2, and 2.5 hours respectively).

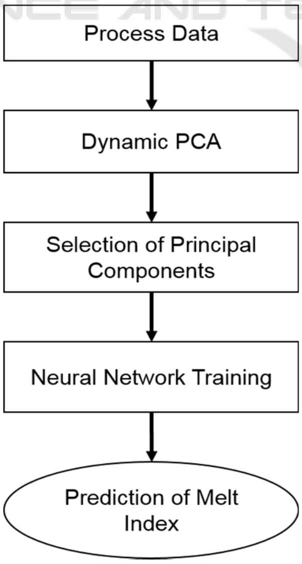

Figure 7: Simplified block diagram of the PCA-NN method.

The SFA-NN method, as described in Section 3, was applied and compared to single neural network (NN) and principal component analysis with neural networks (PCA-NN). The number of dominant slow features was selected by the gradient based method that was described in Section 2.3.3. For PCA-NN, the number of retained principal components (PC) was selected as the PCs that captured up to 99% of the variance. Retained PCs up to 90% variance was also tested although using 99% produced much better generalisation. Figure 7 shows a basic block diagram of the main steps involved in the PCA-NN method.

The data was partitioned into training, testing and unseen validation data sets as described in Section 3. The first 55% of the data was used for training and testing, with the remaining 45% used as unseen validation data. The single hidden layer feedforward neural networks were trained using the Levenberg-Marquardt algorithm with early stopping and regularisation. Regularisation was necessary for this problem because the complexity of the system meant that producing good generalisation performance on the validation data was difficult without it. A sigmoidal activation function was used for the hidden layer neurons and the output layer used a linear function. The optimal number of hidden neurons was selected by cross validation on the testing data's mean squared error (MSE).

Tables 1-3 show the MSE on the validation data along with the R^2 for all three data sets, for each time lag for SFA-NN, PCA-NN and NN. The MSE values in bold show the best model for that time lag across the three techniques. It can be seen that SFA-NN produces the best generalisation for the first three time lags, with these models being the top three for any model across all of the time lags and techniques. The performance of SFA-NN decreases significantly for time lags greater than 1. This is likely because the number of inputs becomes extremely large (e.g. 120 model inputs for $d = 3$) and SFA struggles to capture the relevant slow varying information with such a number of inputs. PCA-NN and NN offer quite similar performance, however, PCA-NN suffers a similar drop in performance as SFA-NN for $d > 1$, while NN seems to handle the increasing number of inputs much better than the other two methods since it displays consistent performance across the different time lags. Another apparent advantage of SFA-NN is that it reduces the dimensionality greater than PCA-NN for all time lags, thus reducing model complexity. The fact that $d = 0$ is the best model for SFA-NN shows that it does not require the additional information from added time lagged inputs since it captures the key trends via the slow feature extraction.

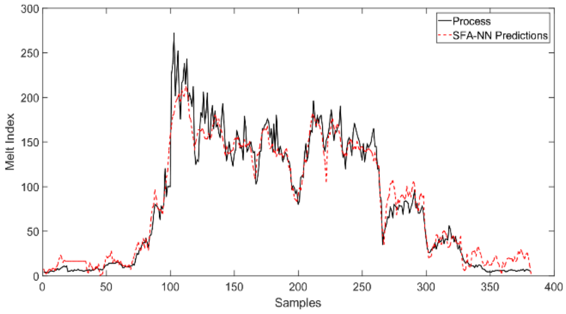

Figure 8: Predictions of MI for full data set using SFA-NN with $d = 0$.

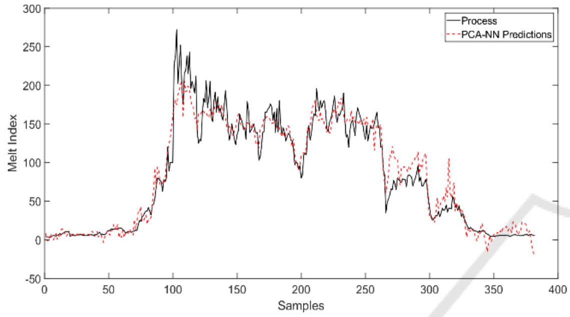

Figure 9: Predictions of MI for full data set using PCA-NN with $d = 0$.

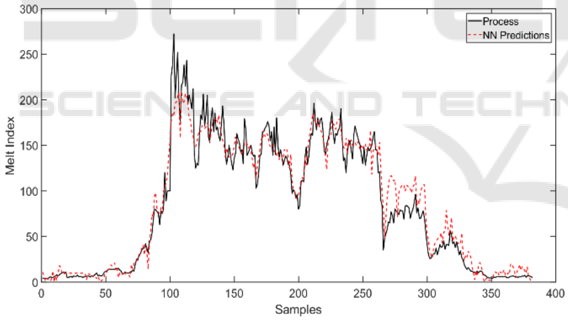

Figure 10: Predictions of MI for full data set using NN with $d = 0$.

Focusing on the models with $d = 0$, since these are the best models across the three techniques, SFA-NN provided a 34.2% and 38.1% improvement in validation MSE when compared to NN and PCA-NN respectively. Figures 8 to 10 show the predictions of MI for the full data set using SFA-NN, PCA-NN and NN respectively, all using $d = 0$. These figures confirm that SFA-NN fits the data the best, particularly on the unseen validation data, represented by the last 45% of the samples. PCA-NN has some predictions that produce a negative MI, highlighting inadequacies in this model.

5 CONCLUSIONS

In this paper, combining slow feature analysis with neural networks for nonlinear process modelling has been presented. Slow feature analysis is used on process data to extract underlying trends in the form of slow features. By retaining a lower number of slow features, model complexity can be reduced through having fewer inputs. Selection of the dominant slow features was performed by observing the slowness for each slow features, represented by the eigenvalues derived from the SFA. The slow features up until a relatively large change in slowness were selected as the dominant features. These dominant features were used as the inputs for building a neural network model. Many industrial processes are complex and nonlinear, and so using neural networks as opposed to linear techniques is often necessary. The proposed SFA-NN method was applied to an industrial polymerisation process for predicting polymer melt index, which is a difficult to measure quality variable with a relatively low sampling rate.

SFA-NN was compared to PCA-NN and NN. Additionally, different time lags for the dynamic inputs were assessed to see the effect on generalisation capability for each technique. The prediction error on unseen validation data was the lowest for SFA-NN for the first three time lags. These were also the best performing models across all of the 18 models that were created for the different time lags and techniques. The $d = 0$ models had the best generalisation performance for each technique and when comparing these models, SFA-NN showed a 34.2% and 38.1% improvement in generalisation over NN and PCA-NN respectively. SFA-NN also used a lower number of latent variables than PCA-NN, reducing the model complexity. Application of the proposed SFA-NN method for the nonlinear modelling of an industrial polymerisation process shows its effectiveness in improving generalisation capability and reducing dimensionality.

REFERENCES

Bhat, N. and McAvoy, T. J. (1990) Use of neural nets for dynamic modeling and control of chemical process systems. *Computers & Chemical Engineering, 14*(4-5), pp. 573-582.

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition,* Oxford; New York: Oxford University Press; Clarendon Press ; .

Breiman, L. (1996) Bagging predictors. *Machine Learning, 24*(2), pp. 123-140.

Chen, S., Billing, S. A. and Grant, P. M. (1990) Non-linear system identifcation using neural networks. *International Journal of Control, 51*(6), pp. 1191-1214.

Cybenko, G. (1989) Approximation by superposition of a sigmoidal function. *Mathematics of Control Signals and Systems* 2, pp. 303–314..

Dong, D. and Mcavoy, T. J. (1994) Nonlinear principal component analysis - based on principal curves and neural networks. *Proceedings of the 1994 American Control Conference, Vols 1-3*, pp. 1284-1288.

Ge, Z. Q., Gao, F. R. and Song, Z. H. (2011) Mixture probabilistic PCR model for soft sensing of multimode processes. *Chemometrics and Intelligent Laboratory Systems, 105*(1), pp. 91-105.

Ge, Z. Q., Huang, B. and Song, Z. H. (2014) Nonlinear semisupervised principal component regression for soft sensor modeling and its mixture form. *Journal of Chemometrics, 28*(11), pp. 793-804.

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

Gonzaga, J. C. B., Meleiro, L. A. C., Kiang, C. and Maciel, R. (2009) ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process. *Computers & Chemical Engineering, 33*(1), pp. 43-49.

Hartnett, M. K., Lightbody, G. and Irwin, G. W. (1998) Dynamic inferential estimation using principal components regression (PCR). *Chemometrics and Intelligent Laboratory Systems, 40*(2), pp. 215-224.

Jolliffe, I. T. (2002) *Principal Component Analysis, Springer series in statistics,* 2nd ed., New York: Springer.

Kadlec, P., Gabrys, B. and Strandt, S. (2009) Data-driven Soft Sensors in the process industry. *Computers & Chemical Engineering, 33*(4),pp. 795-814.

Qin, S. J. (1997) Neural Networks for Intelligent Sensors and Control — Practical Issues and Some Solutions. in *Neural Systems for Control*, San Diego: Academic Press,. pp. 213-234.

Shang, C., Huang, B., Yang, F. and Huang, D. X. (2015a) Probabilistic slow feature analysis-based representation learning from massive process data for soft sensor modeling. *AIChE Journal, 61*(12), pp. 4126-4139.

Shang, C., Huang, B., Yang, F. and Huang, D. X. (2016) Slow feature analysis for monitoring and diagnosis of control performance. *Journal of Process Control, 39*, pp. 21-34.

Shang, C., Yang, F., Gao, X. Q. and Huang, D. X. (2015b) Extracting latent dynamics from process data for quality prediction and performance assessment via slow feature regression. *2015 American Control Conference*, pp. 912-917.

Shang, C., Yang, F., Gao, X. Q., Huang, X. L., Suykens, J. A. K. and Huang, D. X. (2015c) Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis. *AIChE Journal, 61*(11), pp. 3666-3682.

Tham, M. T., Montague, G. A., Morris, A. J. and Lant, P. A. (1991) Soft-sensors for process estimation and inferential control. *Journal of Process Control, 1*(1), pp. 3-14.

Willis, M. J., Di Massimo, C., Montague, G. A., Tham, M. T. and Morris, A. J. (1991) Inferential measurement via artificial neural networks. *Intelligent Tuning and Adaptive Control, 15-17*, pp. 85-90.

Wiskott, L. and Sejnowski, T. J. (2002) Slow feature analysis: Unsupervised learning of invariances. *Neural Computation, 14*(4), pp. 715-770.

Yang, J., Zeng, X. Q., Zhong, S. M. and Wu, S. L. (2013) Effective neural network ensemble approach for improving generalization performance. *IEEE Transactions on Neural Networks and Learning Systems, 24*(6), pp. 878-887.

Zhang, J. (1999) Developing robust non-linear models through bootstrap aggregated neural networks. *Neurocomputing, 25*, pp. 93-113.

Zhang, J., Jin, Q. B. and Xu, Y. M. (2006) Inferential estimation of polymer melt index using sequentially trained bootstrap aggregated neural networks. *Chemical Engineering & Technology, 29*(4), pp. 442-448.

Zhang, J., Martin, E. B., Morris, A. J. and Kiparissides, C. (1997) Inferential estimation of polymer quality using stacked neural networks. *Computers & Chemical Engineering, 21*, pp. S1025-S1030.