Data-Driven Modelling and Monitoring of Industrial Processes with Applications in Nuclear Waste Vitrification

Jeremiah Corrigan

A thesis submitted for the degree of Doctor of Philosophy

School of Engineering
Newcastle University
Newcastle upon Tyne
United Kingdom

September 2021

This document has been prepared by Jeremiah Corrigan in the course of employment by Sellafield Ltd. Some of the information it contains is owned by Sellafield Ltd or NDA and is, or may be, proprietary or subject to restrictions relating to national security, commercial or personal obligations. It is submitted to Newcastle University for the purposes of professional assessment. It may be copied and distributed as required for this purpose, but no other use may be made of any component owned by Sellafield Ltd or NDA without the prior permission of Sellafield Ltd.

Abstract

Process models are critical for process monitoring, control, and optimisation. With the increasing amount of process data and advancements in computational hardware, data-driven models are a good alternative to mechanistic models, which often have inaccuracies or are too costly to develop. One problem with data-driven models is the difficulty in ensuring that the models perform well on new data and produce accurate predictions in complex situations, which are frequently encountered in the process industry.

Within this context, part of this thesis explores developing better data-driven models through using a latent variable technique, known as slow feature analysis, as a pre-processing step to regression. Slow feature analysis extracts slow varying features that contain underlying trends in the data, which can improve model performance through providing more meaningful information to regression, reducing noise, and reducing dimensionality. Firstly, the effectiveness of combining linear slow feature analysis with a neural network is demonstrated on two industrial case studies of soft sensor development and is compared with conventional techniques, such as neural networks and integration of principal component analysis with a neural network. It is shown that integration of slow feature analysis with neural networks can significantly improve model performance. However, linear slow feature analysis can fail to extract the driving forces behind data in nonlinear situations such as batch processes. Therefore, using kernel slow feature analysis with a neural network is proposed to further enhance process model performance. A numerical example was used to demonstrate the effective extraction of driving forces in a nonlinear case where linear slow feature analysis cannot. Model generalisation performance was improved using the proposed method on both this numerical example, and an industrial penicillin process case study.

Dealing with radioactive nuclear waste is an important obstacle in nuclear energy. Sellafield Ltd have a nuclear waste vitrification plant which converts high-level nuclear waste into a more stable, lower volume glass form, which is more appropriate for long term storage in sealed containers. This thesis presents three applications of data-driven modelling to this nuclear waste vitrification process. A predictive model of the pour rate of processed nuclear waste into containers, an early detection system for blockages in the dust scrubber, and a model of the long-term chemical durability of the stored glass waste. These applications use the previously developed slow feature analysis methods, as well as other data-driven techniques such as extreme learning machine and bootstrap aggregation, for enhancing the model performance.

Acknowledgments

Firstly, I would like to show my great appreciation for my academic supervisor, Dr Jie Zhang, for his guidance, support, and constructive criticism throughout my 4-year studies. I would also like to thank my industrial supervisor from Sellafield Ltd, Katy Spencer, who has also been very supportive and provided important knowledge and insight into the operations at Sellafield Ltd that have helped develop the work for them. I also greatly appreciate all the other staff in the chemical engineering department, who I have worked with through demonstrating and other activities, including all the admin staff that work hard behind the scenes for us research students. To all the research students and other people I have met and made friends with during this time, thank you for making the process much more enjoyable.

My biggest thanks go to my family, who are always there for me no matter what, and to my partner, who I would not have met if I had not done this PhD, for her endless support and love that I probably do not deserve. Doing the final year and a half of this PhD at home during a global pandemic is something no one would have expected, and so this support was needed more than ever.

Finally, thank you to the Engineering and Physical Sciences Research Council (EPSRC) and Sellafield Ltd for funding and supporting this work.

Table of Contents

List of Figures

List of Tables

Publications

Conference proceedings

Corrigan, J. and Zhang, J. (2019) *ICINCO 2019 - Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics*. Available at: https://www.scitepress.org/Link.aspx?doi=10.5220/0007958904390446.

Journal publications

Corrigan, J. and Zhang, J. (2020) 'Integrating dynamic slow feature analysis with neural networks for enhancing soft sensor performance', *Computers & Chemical Engineering*, 139, p. 106842.

Corrigan, J. and Zhang, J. (2021) ‘Developing accurate data-driven soft sensors through integrating dynamic kernel slow feature analysis with neural networks', *Journal of Process Control*, 106, p.208-220

Chapter 1. Introduction

1.1 Background

In industrial process plants, sensors provide measurements of key process variables that are used for process control and monitoring, and the ability to enhance these aspects can lead to improved product quality, reduced costs, and safer processes. For many companies, falling outside of certain process limits can not only lead to financial losses, but also breaching rules and regulations. In particular, the emphasis on the reduction of environmental pollution in more recent years has led to stricter laws that companies must adhere to. These goals highlight the need for frequent and reliable measurements of certain process variables, and optimised process monitoring and control.

In many processes, monitoring a wide range of process variables using hardware sensors can be of great expense and in some cases, on-line measurements may not be feasible. Furthermore, some hardware may not provide measurements at a high enough frequency or sufficient reliability, leading to sub-optimal control and monitoring. Off-line sampling is an alternative; however, this typically produces a lower rate of samples and significant delays. In addition, faults can arise in hardware sensors and planned maintenance leaves sensors out of order for a time.

The alternative arises in using a model in the form of a "soft sensor" (software-based sensor). A soft sensor (software based sensor) is a form of process modelling where a model is used to predict measurements of difficult to measure key product quality variables from other easily measured process variables (Tham *et al.*, 1991). A process model forms the basis of a soft sensor. Often measurements of certain product quality variables are infrequent or have long time delay, typically due to offline laboratory analysis. These infrequent and delayed measurements can affect process control performance, even with advanced control algorithms, because they lead to delayed disturbance detection which can result in large drifts from setpoints. One solution is to create soft sensors for these product quality variables so that predictions can be made with a more frequent sampling rate and minimal or no time delay. This will lead to improved process control and monitoring performance.

Generally speaking, process models are created through two primary methods, mechanistic and data-driven modelling. Mechanistic models are commonly based on first principle models, which mathematically describe the physical and chemical nature of the process using fundamental knowledge, but other techniques such as Kalman Filtering (Kalman, 1960) have been used (de Assis and Maciel, 2000; Guo *et al.*, 2014). However, these models commonly focus on ideal steady states and so can make a poor soft sensor in real process conditions, which can be different from the ideal conditions due to process disturbances and degradation of process equipment, such as fouling in heat exchangers and reactors (Kadlec *et al.*, 2009). Mechanistic models do have the advantage of being able to exactly describe the process over a full range, however, many modern industrial processes are highly complex, and the production of a mechanistic model can be very time consuming and expensive, and any inaccuracies or assumptions of first principle knowledge can lead to a reduction in model accuracy.

The other type of modelling is data-driven, which, as the name suggests, use the process data to produce a model through an assortment of potential techniques. If sufficient process data is available, then data-driven modelling can overcome some, if not all, of these issues associated with some mechanistic models. Since data-driven models are based on the on-line measurements of the process, they can better describe the real conditions of the process, compared with some mechanistic models that have inaccuracies. Data-driven models do not require extensive process knowledge in their production and can create more accurate models where mechanistic models fail to adequately describe the process because the data can capture the real relationships between variables. These main advantages of data-driven over mechanistic modelling are summarised in Figure 1.1.

The main issue with data-driven models is in the quality and quantity of the data used for building the model. Data needs to cover the full range of operating conditions that the soft sensor is intended to be applied to because these models should only be able to interpolate and not extrapolate. Additionally, data quality issues, such as missing data and noise, are very common. Therefore, the correct selection and pre-processing of process data is critical to the success of a soft sensor.

Another key issue of data-driven models is in the deterioration of their performance due to process changes, often known as drifting data. If process conditions change, whether it be suddenly or gradually over time, then the model may not be within the conditions in was initially built upon. Gonzalez (1999) calls this the region of validity of the model and also discusses this issue of drifting process data in detail. In cases of process drift, the model will often need to be retrained so it is valid within the limits of the new process conditions. However, updating the data-driven model is not necessarily an easy task because the generation of new process data that covers an acceptable range (to give the sensor an acceptable region of validity**)** would require variation of process conditions. Having the inputs of the process varying in such a manner will disturb normal process operations, therefore being a costly procedure that should not be done very often. Adaptive models are one method of dealing with this, as discussed in Kadlec *et al.* (2011), where the authors review the types of adaptive soft sensors. Another way to overcome the costly manner of process data generation is what is known as smart data generation, which is to optimise the process changes necessary to generate sufficient and rich enough data for model training.

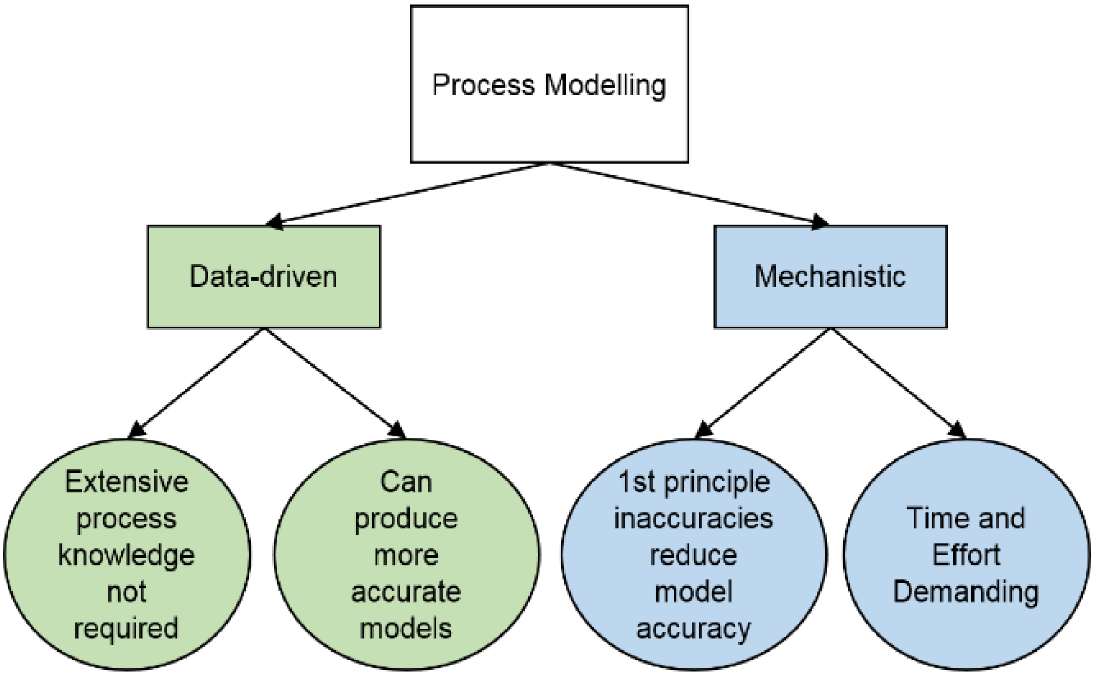
Process Modelling
Data-driven
Mechanistic
Extensive process knowledge not required
Can produce more accurate models
1st principle inaccuracies reduce model accuracy
Time and Effort Demanding

Figure 1.1 – Key advantages of data-driven models over mechanistic models

Aside from data selection and pre-processing, the next crucial stage in ensuring good data-driven model performance is through the techniques used.

1.2 Aims and Objectives

The first main aim, described in the first half of this work, focuses on enhancing the performance of nonlinear, data-driven modelling techniques. This is done by developing novel, nonlinear techniques and then demonstrating their improved performance over other similar techniques on some benchmark process case studies. Ensuring that data-driven models have strong generalisation performance (perform well on new data) is critical if they are to be successfully implemented on real processes.

The second main aim is concerned with using these developed techniques for modelling and monitoring applications for multiple problems that have arisen from the nuclear waste vitrification process at Sellafield Ltd. Dealing with nuclear waste is critical to ensuring that radioactive elements do not leach into the environment. Therefore, optimising this process is critical for public safety, as well as improving efficiency and reducing costs.

1.3 Scope of the Research

The scope of this research is to overcome disadvantages associated with hardware sensors through utilising soft sensors. The disadvantages of mechanistic soft sensors are overcome through data-driven models. Additionally, data-driven modelling is used to tackle issues associated with Sellafield Ltd's nuclear waste vitrification process.

This work was done by utilising latent variable techniques, in combination with neural networks, to optimise data-driven modelling performance. The primary focus of the latent variable techniques is slow feature analysis, but other similar techniques (principal component analysis) were used for comparison. Furthermore, only neural networks were used as a machine learning technique. This was to highlight the effectiveness of incorporating a nonlinear technique over a linear technique, as well as having a constant nonlinear technique to properly highlight the differences in incorporating slow feature analysis (and other latent variable techniques) into the method. There are many other data-driven modelling techniques that could be used for comparison; however, this is beyond the scope of this work, which was to highlight the effectiveness of slow feature analysis over principal component analysis, and neural networks over linear regression.

Furthermore, the focus of comparison between techniques was the goodness of fit, using metrics (R^2/root mean squared error) and visual inspection. Computational time was not considered for comparison in the scope of this work. However, for no techniques were the differences in computational times detrimental to soft sensor application. This is because training is performed offline, where slightly longer computational times are not very important. When processing new samples online, all techniques have fast computational times, and any differences are insignificant.

1.4 Thesis Layout

The following summarises the main content and contributions of each chapter in this thesis. Chapter 2 provides background to process modelling and monitoring, including literature review of data pre-processing, data-driven modelling applications and data-driven modelling techniques.

Chapter 3 presents the development of a novel, data-driven process modelling technique using slow feature analysis (SFA) and neural networks. This work proposes integrating SFA with neural networks (SFA-NN) for soft sensor development. Dynamic linear SFA is applied to the easy-to-measure process variable data. Then the dominant slow features are selected as the inputs of a neural network to predict the difficult-to-measure product quality variables. SFA can capture the underlying dynamics of industrial processes through the extraction of slowly varying latent variables, known as slow features. The extracted underlying trends provide more meaningful information to the neural network and reduce noise in the data through retaining only the slower features, which also reduces model complexity through dimensionality reduction. These aspects all lead to improved model generalisation performance. Selection of dominant slow features using a scree plot is proposed. Neural networks are utilised to cope with nonlinearities present in many real industrial processes. The effectiveness of the proposed method is evaluated on two real industrial processes and is compared with slow feature regression, partial least square regression, traditional feedforward neural networks, and using principal component analysis (PCA) prior to a neural network (NN), PCA-NN. The proposed SFA-NN method gives the best generalisation performance among these techniques in both case studies.

Chapter 4 further develops the work in Chapter 3 by creating a nonlinear version of slow feature analysis using kernels. The work proposes another novel technique and application, developing a data-driven soft sensor modelling approach based on dynamic kernel slow feature analysis (KSFA) and a neural network. Slow feature analysis is a feature extraction method that aims to extract slowly varying features that can capture the driving forces behind data. However, there are situations where linear SFA (LSFA) cannot capture the driving forces due to nonlinear relationships between the driving forces and input signals. KSFA is a nonlinear extension of LSFA that utilises the kernel trick to map the inputs into a higher-dimensional feature space. Extracting the nonlinear driving forces can improve soft sensor performance by utilising the nonlinear slow features as inputs to a neural network, which provides information on the key underlying trends, with the added benefit of noise reduction. Combining KSFA with a neural network further improves soft sensor performance for cases where nonlinear relationships between the driving forces and soft sensor outputs are present. The effectiveness of the proposed method is first demonstrated on a numerical example, where the theoretical advantages of KSFA can be easily observed. It is then applied to a benchmark simulated industrial fed-batch penicillin process, where the improvement over other similar techniques is demonstrated further.

Chapter 5 presents a data-driven predictive model for the pour rate of high-level waste from the melter to container in the nuclear waste vitrification process at Sellafield Ltd. While the previous two chapters focused on optimising data-driven modelling techniques to improve model performance, this chapter focuses on producing the best model for a real industrial problem. Predicting the pour rate prior to a pour has not been done previously. The main challenges presented in this chapter involve concept development, data selection and pre-processing, and selecting the right technique to optimise model generalisation performance. Bootstrap aggregation (bagging) was used to create more reliable and robust models. Additionally, slow feature analysis was integrated into some of the tested models, and it further demonstrated its strength over similar techniques such as principal component analysis (PCA).

Chapter 6 deals with another process challenge for Sellafield Ltd's nuclear waste vitrification process. The dust scrubber is a key part of the process because it helps recycle particles in off-gas back to the calciner to improve efficiency and prevent radioactive particles going into the environment. Narrow pipe work means that particles can accumulate and form a blockage. When the pipework becomes fully blocked, it requires down time and maintenance that is very costly. The work in this chapter looks at using a multivariate statistical process control approach to aid in early blockage detection to save on these costs. Several techniques are used to optimise the early blockage detection, including PCA and SFA. Additionally, residual contribution plots were utilised to assess the impact of each process variable on the blockage. As with Chapter 6, data pre-processing and selection present the main challenges when dealing with real process data such as this.

The final application of data-driven modelling for the nuclear waste vitrification process is presented in Chapter 7. The vitrified nuclear waste stored in stainless steel containers is likely to remain radioactive for a very long time and when stored in deep geological repositories, they may encounter groundwater when the containers eventually erode. Understanding the leach rate of radioactive elements into the groundwater for different types of glass could lead to optimising the glass compositions to minimise leaching into the environment. Data-driven models of the leach rate of 8 elements were developed based on accelerated static leach rate experimental data. Bootstrap aggregated extreme learning machine with principal component analysis as a pre-processing step was used to improve model generalisation performance over single extreme learning machines models and neural network-based models. A sensitivity analysis was carried out to optimise the composition of main glass forming additives to minimise the total leach rate.

Chapter 2. Data-Driven Modelling and Monitoring of Industrial Processes

2.1 Process Data

In its most basic form, an industrial process creates a desired product from raw materials, typically on a large scale. These industries cover a wide range of sectors, not limited to but including chemical, oil and gas, pharmaceutical, consumer goods and food production. These processes typically fall into two main categories: continuous processes and batch processes.

Continuous processes, as the name suggests, run in a constant way so that product is always produced between plant start up and shut down.

Batch, or semi-batch, processes produce product in a distinct time period. Batch processes are commonly used in food, biochemical and speciality chemicals industries. For example, a fermentation process operates at non steady state until the desired product quality is attained.

From temperatures in reactors, to stirring speed of an impeller, process data has been collected more frequently in the last few decades as both hardware and software technology rapidly develops.

As the quality and quantity of this process data increases, ways to utilise this data to optimise processes is of great value from many aspects. Garcia-Munoz and Macgregor (2016) commented:

> *"Not only do we want to find and interpret patterns in the data and use them for predictive purposes, but we also want to extract meaningful relationships that can be used to improve and optimize a process".*

This statement highlights the true potential of utilising process data for not just process optimisation but also for understanding processes on a deeper, fundamental level.

Although one of the advantages of data-driven modelling seems to be that process knowledge is not required for model development, the reality is slightly different, with process knowledge often needed throughout the different stages of model development. It is often said that an empirical model is only as effective as the data it is built on. The importance of correctly selecting model training data and the necessary pre-processing steps is typically a largely manual task, often requiring expert process knowledge. Data pre-processing involves dealing with missing data, outlier detection and replacement, feature selection, data scaling and noise reduction; all with the aim of improving data quality to improve model performance.

2.1.1 Outliers

A data outlier is typically a measurement from a sensor that deviates from an expected or meaningful range. Pearson (2002) informally defined outliers as data points within a dataset that are inconsistent with expectations, based on the majority of available data. There is a classification of outliers that can split them into two categories: obvious and non-obvious outliers (Kadlec *et al.*, 2009). Obvious outliers violate some clear bounds, often fundamental or technical, such as negative concentrations. Non-obvious outliers are within these fundamental limits but are still outside of meaningful or expected ranges. These non-obvious outliers are the most difficult to detect and deal with. Figure 2.1 shows a non-obvious outlier in a process variable that is easily detected by visual inspection. This variable operates within a defined pattern and so this single point so far from the trend must be an error.

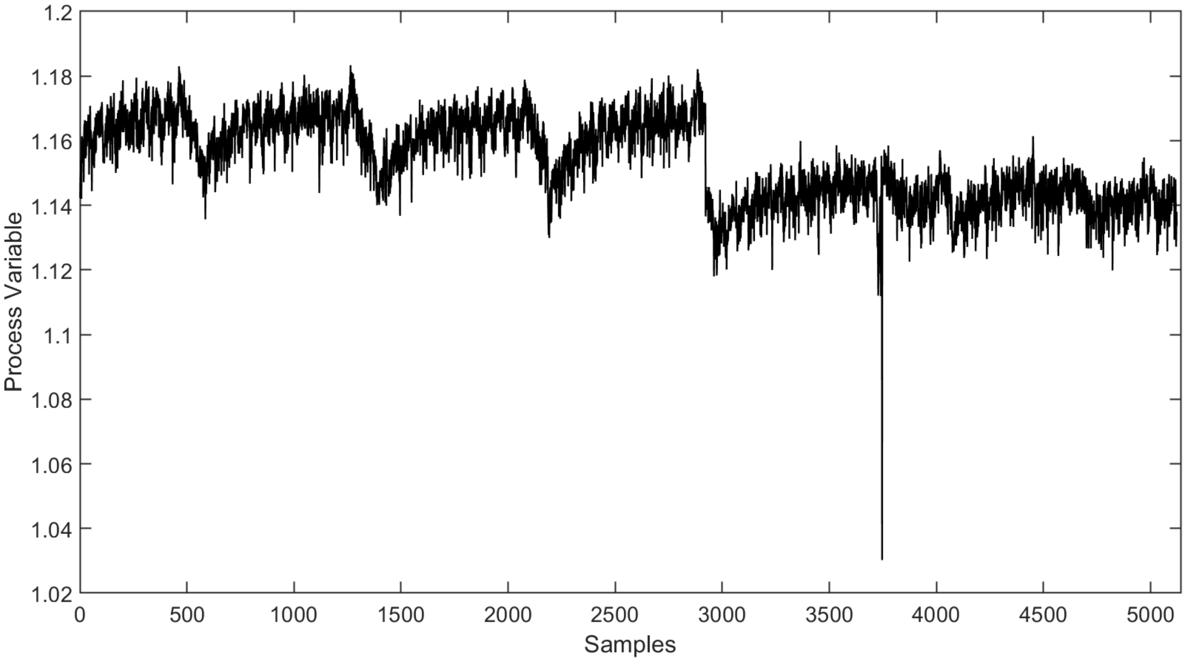

Figure 2.1 – Example of a non-obvious outlier in process data

Outliers can have serious consequences, especially when the data has an important role on the plant, such as for modelling and monitoring. Therefore, it is critical to pre-process data to detect these outliers. Often visual inspection with some process knowledge is sufficient to detect outliers, however, when the required level of process knowledge is not available, automatic methods of outlier detection can be employed. However, the results of automatic outlier detection can typically require manual validation to prevent any false positive and false negative detections (Kadlec *et al.*, 2009). One of the most commonly used outlier detection methods is the “3σ edit rule”, which detects an outlier as a sample that is three or more standard deviations from the mean (Barnett, 1994). This rule fails in many circumstances because it detects too few outliers, since outliers typically increase the estimated variance, as discussed in detail by Pearson (2002). Pearson (2002) also discussed many other issues in dealing with outliers in process data, especially when considering process modelling applications. Primarily, that outliers cannot simply be ignored, and must either be detected and replaced, or alternative methods be used that are less sensitive to outliers.

In the study by Wang *et al.* (2019), the authors presented a more up to date review of current outlier detection techniques, including, but not limited to, statistical-based methods (such as principal component analysis), ensemble-based methods, and learning-based methods.

When an outlier is detected, what to do with that sample is another issue. Outliers can have negative effects on data-driven model performance, however, removing too much information can also have a detrimental impact on generalisation performance, that is, the performance on the model on new, unseen data. An important variation in the data that needs to be included in the model training could be mistaken for an outlier, especially if manual validation with process knowledge is not performed. One method for dealing with an outlier is removing the sample or the variable entirely. If there is no clear and sensible way to replace the sample, then it can be removed if it does not remove too much important information for the model creation. If a large amount of outliers is present in a single variable, then removing the variable may be the best option. Replacing an outlier can be done using the mean, however, process data often comes in a range of conditions and so using the total mean of all process conditions may be a poor choice. More often than not, using a local mean/median (from a window of data of similar operating conditions) or interpolation is more appropriate.

Without some level of process knowledge, deciding on what method to use, for both outlier detection and replacement/removal, can be difficult.

2.1.2 Missing data

Missing data can be defined as measurement samples which have values that do "*not reflect the real state of the physical measured quantity*" (as quoted by Kadlec *et al.* (2009)). Typically, these samples come in the form of a 0 value, NaN (not a number), or another constant value that does not make sense from a operational standpoint. Missing process data can be caused by hardware (sensor) malfunctions or technical issues, such as the measurement not being recorded on the database (Kadlec *et al.*, 2009). Figure 2.2 provides an example of missing process data, illustrating both NaN values and 0 values to indicate two periods missing data.

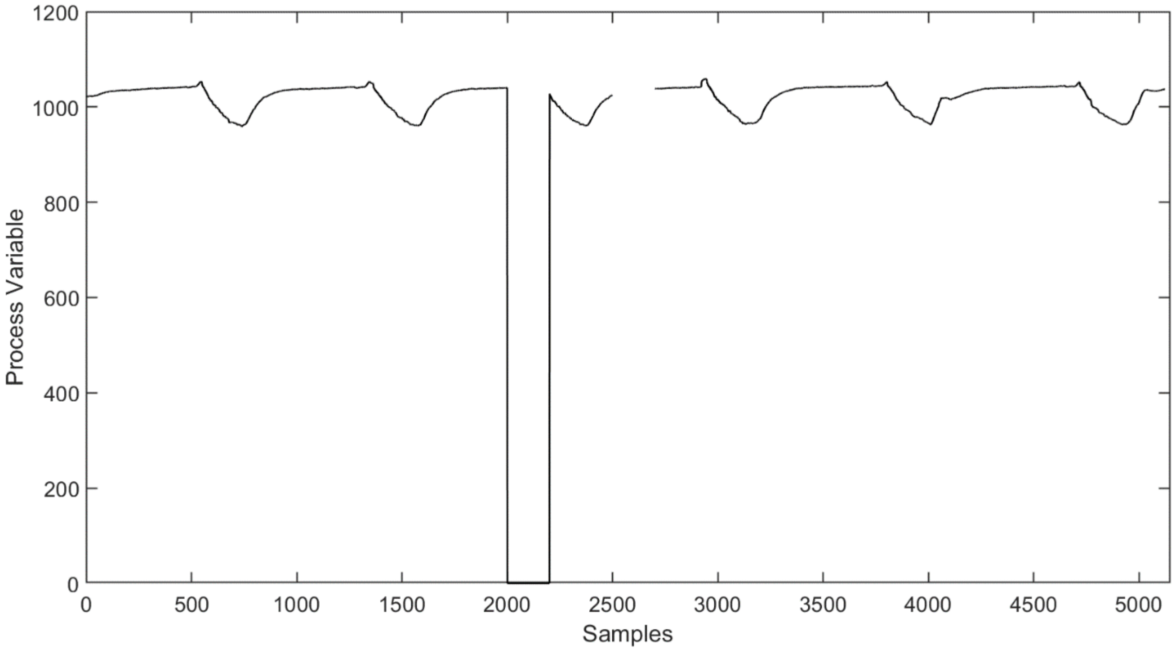

Figure 2.2 – Example of missing data in a process variable

Most data-driven techniques cannot deal with missing data and so proper handling of missing data is important. Replacement methods are usually similar to the outlier replacement methods discussed in Section 2.1.1. Baraldi and Enders (2010) discussed both traditional and modern missing data techniques, particularly focusing on maximum likelihood and multiple imputation. Multiple imputation is a relatively new method and involves replacing the missing data multiple times to create several different data sets and then using statistical analysis. Although multiple imputation often achieves better results than single imputation, it requires creating multiple data sets and so requires more computational effort (Zhu *et al.*, 2018).

A review of missing data strategies, as a part of a review of data pre-processing methods for process modelling, is discussed by Zhu *et al.* (2018). The authors considered three main issues of missing data (proportion, patterns and mechanisms), as well as some replacement methods such as hot-deck substitution, conditional distribution-based substitution, and multiple imputation.

Walczak and Massart (2001a; 2001b) produced a two-part publication of methods to deal with missing data. The first part focuses on an iterative algorithm applied to classic chemometrical methods such as PCA and partial least squares (PLS). The second part presents an expectation-maximisation algorithm based on a maximum likelihood approach.

2.1.3 Noise

Sensor and measurement noise is common in process data and can lead to model performance issues if not properly dealt with. As demonstrated by the example in Figure 2.3, high levels of noise are often found in process measurements, and it can make extracting meaningful information from the raw data very difficult.

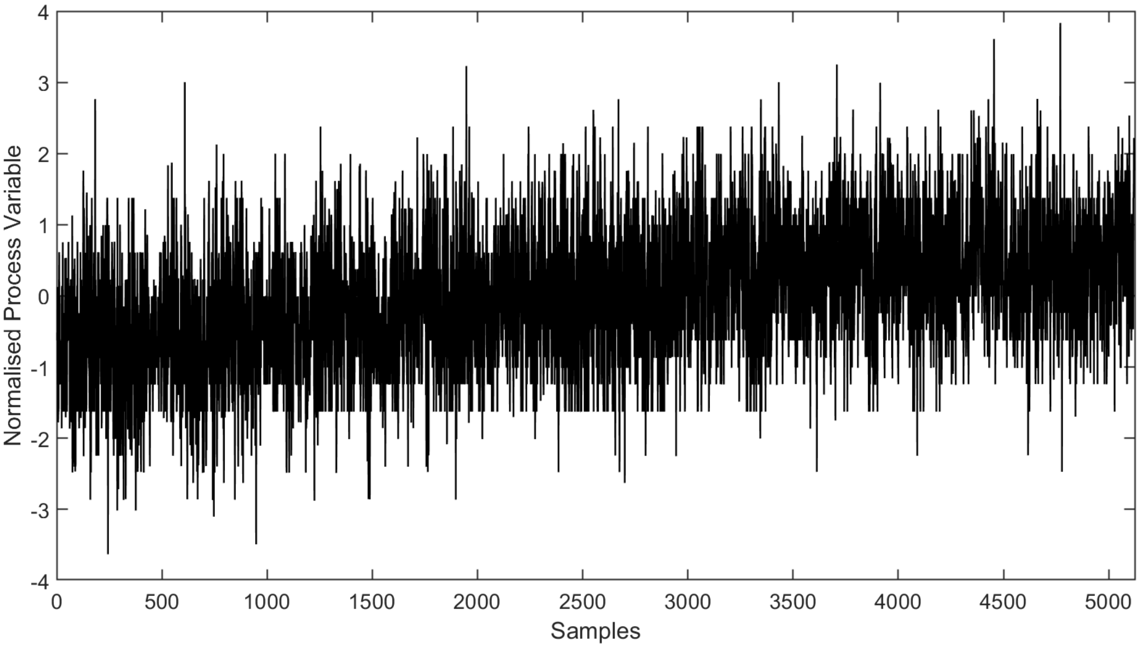

Figure 2.3 – Real example of noise in process data

Pre-processing noisy data can be done using statistical latent variable methods such as principal component analysis (Jolliffe, 2002), partial least squares (Wold *et al.*, 1984) and slow feature analysis (Wiskott and Sejnowski, 2002). These methods map the original data into a new feature space in the form of latent variables, which are ordered in terms of variation (PCA, PLS) or slowness (SFA). By only retaining a certain amount of these latent variables, that is fewer than the total number of variables, the effect of noise can be reduced as the noise is typically represented in the latter, higher-order latent variables because they contain the least variance or slowness, which is typical of noise (Zamprogna *et al.*, 2004).

Kaneko and Funatsu (2015) proposed different smoothing methods, such as moving average-based methods and Savitzky-Golay filtering, that could be superior to the statistical projection methods (PCA, PLS) for dealing with noise for process modelling. This is due to applying the smoothing methods of each process variable, not each sample, such that the temporal variations in time series data are better captured. Methods such as Savitzky-Golay filtering can reduce the effect of noise while also retaining the time-varying nature of time series data.

Slow feature analysis (SFA) can capture underlying time-varying trends in time series data, whilst also reducing the effect of noise. SFA reduces the effect of noise by retaining only the slowest features as the fastest features typically only represent noise. The ability of SFA to extract underlying trends provides a significant advantage to process modelling over the smoothing methods. These properties of SFA have been shown to improve process modelling generalisation performance when compared with the statistical projection methods, PCA/PLS (Shang *et al.*, 2015a; Fan *et al.*, 2018; Qin *et al.*, 2019; Corrigan and Zhang, 2020; Jia *et al.*, 2020; Yuan *et al.*, 2020b).

2.1.4 Data co-linearity

Data co-linearity presents another issue for process modelling because often process data can be very co-linear, that is, different process variables can be highly correlated with each other. For example, two pressure sensors on an industrial plant may be in similar locations and therefore may provide very similar present measurements. This is very typical on plants where detailed information is required for process control and in case of sensor malfunction, backups sensors are in place. However, this excess of similar information can be detrimental to process modelling as it increases the complexity without additional meaningful information. Therefore, dealing with co-linearity in the data leads to improvements in model performance. An example of real co-linear process data is shown in Figure 2.4, which demonstrates 3 process variables providing very similar information (data rich but information poor).

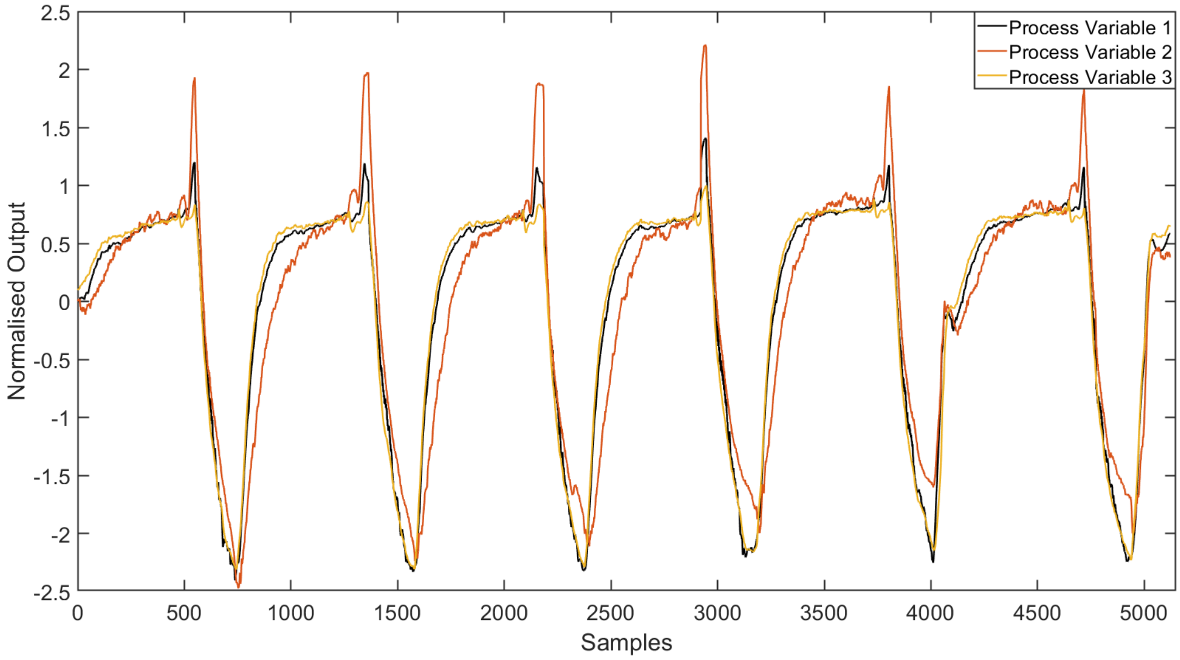

Figure 2.4 – Example of co-linear process data

One of the most common ways to reduce co-linearity is to select a reduced set of input variables that is less co-linear. This can be simply done by removing the variables that provide similar measurements, which often requires process knowledge. Without the process knowledge, this can be done by cross-correlation methods to spot the co-linearity amongst input variables. However, removing variables is prone to sensor faults and noise. The other common methods are to use statistical projection methods, such as PCA and PLS. This works by converting the inputs to a reduced feature space that is less co-linear (Kadlec *et al.*, 2009). PCA has been widely used in soft-sensing applications to deal with data co-linearity (Ge *et al.*, 2011; Xibilia *et al.*, 2020; Zhang *et al.*, 2020b; Li *et al.*, 2021a).

2.2 Introduction to Data-Driven Modelling in the Process Industry

With an abundance of process data available, which is pre-processed to tackle the issues mentioned in Section 2.1, the question becomes: how can we use this data to optimise a process? The answer comes in the form of a predictive model, which estimates a measurement (output) from other correlated measurements (input). As stated by Hangos (2001):

> *"A model is an imitation of reality and a mathematical model is a particular form of representation. We should never forget this and get so distracted by the model that we forget the real application which is driving the modelling. In the process of model building we are translating our real world problem into an equivalent mathematical*

> *problem which we solve and then attempt to interpret. We do this to gain insight into the original real world situation or to use the model for control, optimization or possibly safety studies."*

2.3 Data-driven Modelling Applications

2.3.1 On-line prediction – soft sensors

One main use of data-driven models if for soft sensors used for on-line prediction, often as a backup for hardware sensors or where no sensor is available – either because the technology is not available, or it is too expensive to do so. Typically, these predictions are required for product quality measurements.

One of the first published uses of the term "soft sensor" dates back to 1991 where Tham *et al.* (1991) used state-space estimation and applied it to an adaptive inferential control scheme for several industrial case studies. It was found that earlier detection of the effects of load disturbances, due to increase output measurements from the soft sensor, can lead to faster disturbance rejection in the closed loop control scheme. Since that time, the research into soft sensors has grown and more industries, particularly the chemical, oil and gas industries, have made use of soft sensors on plant. Kadlec *et al.* (2009) provided a significant list of research applications for soft sensing in the process industry, particular for data-driven soft sensors.

The application of soft sensors has been vast, ranging from wastewater treatment (Chang and Li, 2021; Fernandez de Canete *et al.*, 2021; Foschi *et al.*, 2021; Wang *et al.*, 2021a) to bioprocesses (de Assis and Maciel, 2000; Gopakumar *et al.*, 2018; Wang *et al.*, 2020a; Zhang *et al.*, 2020a; Li *et al.*, 2021b), for example.

2.3.2 Process monitoring and fault detection

The application of data-driven models to process monitoring is often known as statistical process monitoring (SPM) or statistical process control (SPC). Traditionally, control charts such as Shewhart, CUSUM and EWMA charts were used to monitor key process variables to detect any abnormal variations that could point to a process fault or a decline in product quality. However, this method of SPC assumes that process variables are independent and so does not consider the correlation of variables.

Some of the first work in applying multivariate statistical methods to process monitoring, named multivariate statistical process control (MSPC), was carried out by Kresta *et al.* (1991) and Macgregor and Kourti (1995). Typically, methods such as PCA and PLS are the choice for

MSPC due to their ability to capture the correlations between variables. Non-linear extensions of these methods have also been investigated (Dong and Mcavoy, 1994). Hotelling's T^2 statistic and the Q statistic, also known as the squared prediction error (SPE), are classically used to detect an abnormal situation in MSPC. Not only can these methods be used for fault detection, but also for fault diagnosis and reconstruction. For fault diagnosis, contribution plots can easily be used with a PCA or PLS based method and no prior knowledge is required in their creation, however, this knowledge is often used for their interpretation (Qin, 2003).

Qin (2012) provided a detailed survey on data-driven process monitoring and diagnosis, highlighting some of the issues associated with some of these methods, particularly in the area of fault detection and diagnosis. In fault diagnosis, the contribution plots can be used to assess which variables have the largest contributions to a fault, however, they are prone to misdiagnosis and so should only really be used for initial testing in fault diagnosis. This misdiagnosis occurs due to an effect known as "smearing", which is where the contribution of one variable spreads to other variables (Qin, 2012). An improved method for analysing variable contributions is using reconstruction-based contributions of a variable (Alcala and Qin, 2009), with extensions of this method to kernel PCA having been used (Alcala and Qin, 2010). These reconstruction-based contributions provided an increased rate of correct fault diagnosis when compared to traditional contribution plot methods. Venkatasubramanian *et al.* (2003) provided a comprehensive review of the methods used in process detection and diagnosis based on process history methods, focusing on data-driven methods.

The need to detect faulty sensors is important because most data-driven modelling methods are not able to distinguish between data from a working sensor and a faulty sensor. Typically, sensor fault detection and reconstruction methods are based on PCA, similar to process fault detection. Dunia *et al.* (1996) presented a method of sensor fault detection via reconstruction using PCA. A sensor validity index was proposed for detecting faulty sensors in that work. Extensions of this method to dynamic process has also been investigated (Lee *et al.*, 2004a). More advanced methods for sensor fault detection have been applied, such as the use of multi-stage Bayesian belief networks (Mehranbod *et al.*, 2005).

Not only can soft sensors be used for sensor fault detection, but a soft sensor can be used as a back-up for the hardware, taking over as the primary sensor if the hardware is removed for maintenance (Fortuna, 2007).

2.4 Data-Driven Modelling Techniques

The area of data-driven models is constantly evolving due to the development of new techniques and advances in computational hardware that enable more complex models to be built without excessive computational effort. The need to create more robust (i.e. accurate predictions when slightly different conditions occur) and reliable (i.e. consistently accurate predictions under normal conditions) models is of great importance for process modelling applications. Techniques that extract more information, and more meaningful information, are crucial to improving generalisation performance of a process model.

2.4.1 Principal component regression

Principal component regression (PCR) uses PCA as a pre-processing step to a linear regression (LR) model. PCA aims to reduce the number of variables into a set of new variables, known as principal components (PCs), which are orthogonal to each other and retain the most variance possible. Jolliffe (2002) provided a detailed description of PCA, along with applications and adaptations of the method. A brief description of PCR is given below.

The general form of linear regression is

$$\hat{y} = X\beta, \tag{2.1}$$

where X is the input matrix, $\hat{y}$ is the prediction of the output and β is the weighting matrix, which is calculated as

$$\beta = (X^T X)^{-1} X^T Y. \tag{2.2}$$

Here Y is the output vector.

To combine PCA with LR, the PCA scores are used as the new input matrix to LR. Given that the scores are calculated as

$$T = XP_{PCA}, \tag{2.3}$$

then the LR equation is transformed to the following:

$$\hat{y} = T\beta_{PCA} \tag{2.4}$$

$$\beta_{PCA} = (T^T T)^{-1} T^T Y, \tag{2.5}$$

where T is the scores matrix, and P_{PCA} is the eigenvector matrix that is derived from carrying out singular value decomposition (SVD) on the covariance matrix of the input data X.

PCA is used for process modelling because it reduces the co-linearity in the data, helps deal with noise by retaining features with the most variance, and reduces the model complexity by retaining fewer PCs. One of the common uses for PCA is for process monitoring using multivariate statistical process control (MSPC) (Macgregor and Kourti, 1995; Bersimis *et al.*, 2007). PCR has been used for inferential estimation (soft sensing) by many researchers (Hartnett *et al.*, 1998; Zhang, 2006; Ge *et al.*, 2011).

One issue with PCA is that it cannot deal with nonlinearity, as only linear combinations of the original variables are used to create the PCs. However, non-linear adaptations of PCA meant that it became more useful for building process models of complex processes. Dong and Mcavoy (1994) proposed a non-linear PCA method that integrated the principal curve algorithm with neural networks. Other extensions of PCA include fast moving window PCA (Wang *et al.*, 2005), dynamic PCA (Lee *et al.*, 2004a), nonlinear semi supervised PCR (Ge *et al.*, 2014) and locally weighted kernel PCR (Yuan *et al.*, 2014).

In another example, Xibilia *et al.* (2020) used PCA prior to a deep neural network due to the high linear correlations amongst the inputs in a hazardous gas detection system. In this way, PCA decorrelated the input data to improve the soft sensor performance. Another application of PCA within soft sensor design is for outlier detection (Bella *et al.*, 2007).

PCA does not just have to be combined with LR, it has also been combined with other regression techniques for process modelling applications, such as with support vector machines (Li *et al.*, 2021a).

One of the main advantages of PCA is that it can compress high dimensional correlated variables into a much smaller number of uncorrelated PCs with minimum information loss. Therefore, a key question for utilising PCA effectively is: how to select the optimal number of retained PCs? Since the aim of PCA is retain the most amount of variance possible in the fewest components, the number of PCs that cumulatively explain a set variance target can be retained. For example, Figure 2.5 shows the cumulative explained variance for the PCs from PCA when it was applied to a case study used later in this thesis. We can choose some minimum limit of the variance we want to ensure we retain, let's say 90%. In this case, over 90% of the variance is explained within the first 5 PCs. For 99% variance, this is achieved in the first 10 PCs. These numbers of PCs are independent to his example. In both cases, this is a significant reduction in the dimensionality from the 27 original variables. The issue with this method is in selecting the limit of variance explained and this is often problem dependant and can be subjective. One reason to not retain all of the variance possible, or even a very high amount such as 99%, is to avoid overfitting when incorporating PCA into a process model. Overfitting involves good performance on training data but poor performance on testing or validation data.

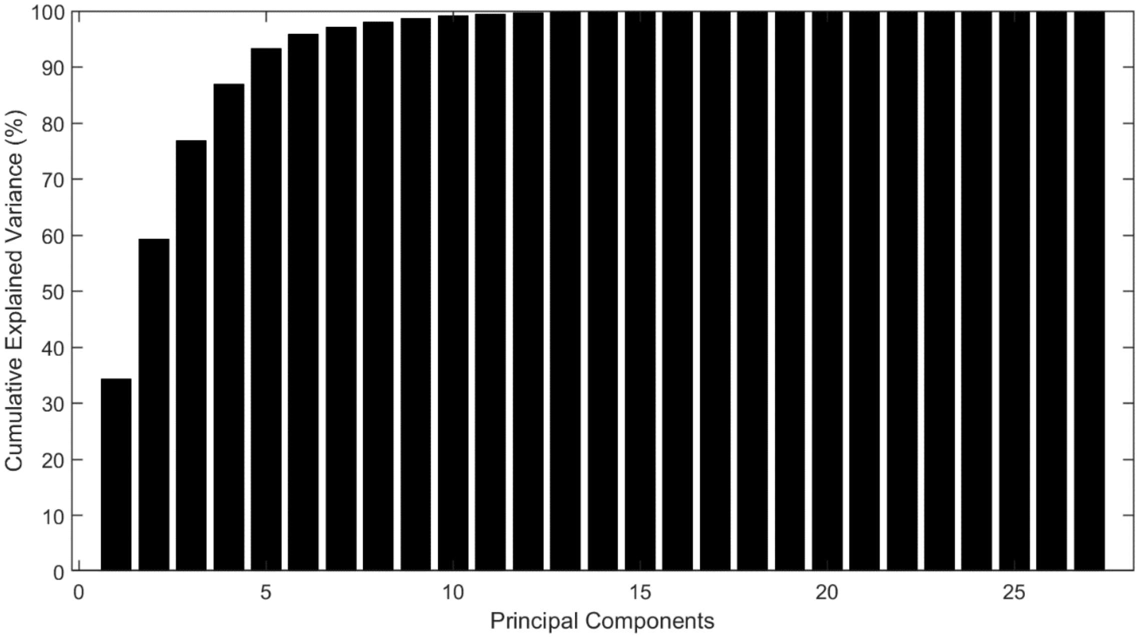

Figure 2.5 – Example of cumulative explained variance of each principal component from PCA

When PCA is incorporated into regression for process modelling, cross validation can be used to select the number of retained PCs; that is, models can be built using each number of PCs individually and the best performing model on a testing data set will give the number of retained PCs. This can help to avoid overfitting.

2.4.2 Partial least squares regression

Partial least squares regression (PLSR) takes PLS and combines it with LR in the same way as PCA does in PCR. In PCA, the scores are calculated such that they explain the most variance possible from the input data. In PLS, scores are calculated on both the input matrix X and the output matrix Y to maximise the explained variance in each. There is an additional objective of achieving maximal covariance between the scores from X and the scores from Y. PLS is described in detail in work by Hastie (2009).

PLS can be incorporated into the model building process in many ways. Zhang *et al.* (2018b) proposed incorporating PLSR into an extreme learning machine (see Section 2.4.4) ensemble (see Section 2.5) to create more reliable soft sensor models. In this case, PLSR was not used as the model prediction technique. PLSR was used to optimise the weights when aggregating the multiple ELM models into the final ensemble. Doing so improved model testing performance when compared with simply averaging the multiple ELM models. This work shows that PLS/PLSR has other applications within the model building process other than a pre-processing stage before the prediction model itself.

PLS has also been used as a pre-processing step to other techniques for process modelling applications, such as bagged neural networks (Zhou *et al.*, 2012) and least squares support vector machine (Li *et al.*, 2017b).

2.4.3 Artificial neural networks

When using data-driven techniques, the process complexity can govern the technique that is required. Linear regression, and extensions such as PCR/PLSR, are the most basic form of data-driven techniques, although using them in many real problems will often lead to inadequate fitting performance. This is particularly true for complex processes that show nonlinearities and cannot be modelled adequately using linear techniques. Therefore, there is a need for more complex algorithms that can create good models in difficult situations where nonlinear relationships exist between the input and output variables.

Artificial neural networks (NN) are among the well-known nonlinear machine learning techniques that have been applied to process modelling and control (Bhat and McAvoy, 1990; Chen *et al.*, 1990; Willis *et al.*, 1991), and soft sensing (Fernandez de Canete *et al.*, 2021). A detailed overview of neural networks is provided by Bishop (1995), covering a variety of topics including the different types of networks, algorithms and generalisation performance (the performance of a model on new data that was not used in model building). A single hidden layer feed forward network (SLFNN) is one of the most used neural networks in machine learning, and although more hidden layers can be used, a single hidden layer with sufficient hidden neurons can approximate any nonlinear, continuous function (Cybenko, 1989). The form of a SLFNN is illustrated in Figure 2.6.

The SLFNN contains three layers: an input, a hidden and an output layer. Each layer consists of nodes (often known as neurons or units) that are connected to every node in the other layers, with each connection having a weight associated to it. In the input layer, the number of input neurons is equal to the number of inputs, and in the output layer, it is equal to the number of outputs. The number of hidden neurons can be any number and it is a hyperparameter that requires optimising, which is typically done by cross validation. Additionally, each node has an activation function that is the same for nodes in each layer but can differ from layer to layer. These are usually linear functions for input and output layers, and a sigmoidal function for the hidden layer, however, other functions are often used, such as tanh (Bishop, 1995).

As shown in Figure 2.6, at each neuron (besides the input neurons), the outputs of the previous layer are multiplied by the connection weights, then summed and passed through the nonlinear activation function f_{hidden}. In the case of the common sigmoidal function,

$$f_{hidden}(x) = \frac{1}{1 + e^{-x}} \, . \tag{2.6}$$

For the *i*th hidden neuron with *J* input variables, the output of a hidden neuron can be mathematically described as

$$y_i^{hidden} = f_{hidden}\left(\sum_{j=1}^{J} w_j^{hidden} \, x_j + b_{hidden} \right), \tag{2.7}$$

where y_i^{hidden} is the output of the *i*th hidden layer, w_j is the connection weight for the *j*th input, and x_j is the *j*th input and b_{hidden} is a bias for the hidden layer.

The outputs of each hidden neuron are then connected to the output layer in the same way. In most process modelling cases, there is one output and so the output of the neural network, y_{output}, can be described as

$$y^{output} = f_{output}\left(\sum_{i=1}^{I} w_i^{output} \, y_i^{hidden} + b_{output} \right), \tag{2.8}$$

where f_{output} is the activation function of the output layer (most often linear), w_i^{output} is the connection weight to the output neuron from the *i*th hidden neuron, and b_{output} is a bias for the output layer.

Neural network modelling involves the training of the network weights (w_j^{hidden} & w_i^{output}) and bias using a learning algorithm. The first, and a still widely used learning algorithm, is the back-propagation algorithm that was proposed in 1986 by Rumelhart *et al.* (1986). The Levenberg-Marquardt algorithm (Marquardt, 1963) is one of the most popular and is a hybrid between a Gauss-Newton method and the gradient descent algorithm. The Levenberg-Marquardt algorithm has been shown to have improved performance over back-propagation (Kermani *et al.*, 2005). The Levenberg-Marquardt algorithm is described in detail by Bishop (1995).

Although neural networks possess universal approximation capability (Cybenko, 1989), poor performance can still be observed in many applications, due to a variety of issues, as discussed by Qin (1997). One of these main issues associated with neural network learning, and data-driven modelling in general, is overfitting. Overfitting is when the model performs well on the data it was trained on but displays poor performance on new, unseen data. The ability to perform well on unseen data is known as the generalisation capability/performance. A variety of techniques have been used to improve generalisation in neural network modelling, such as regularisation and early stopping (Bishop, 1995), and ensemble methods (Breiman, 1996; Zhang, 1999; Yang *et al.*, 2013).

In more recent times, neural networks have evolved and extensions such as deep learning (Shang et al., 2014; Yuan et al., 2020b), ensembles (Li et al., 2015; Yi et al., 2020) and echo state networks (Bo et al., 2020; He et al., 2020) have been widely used for process modelling applications.

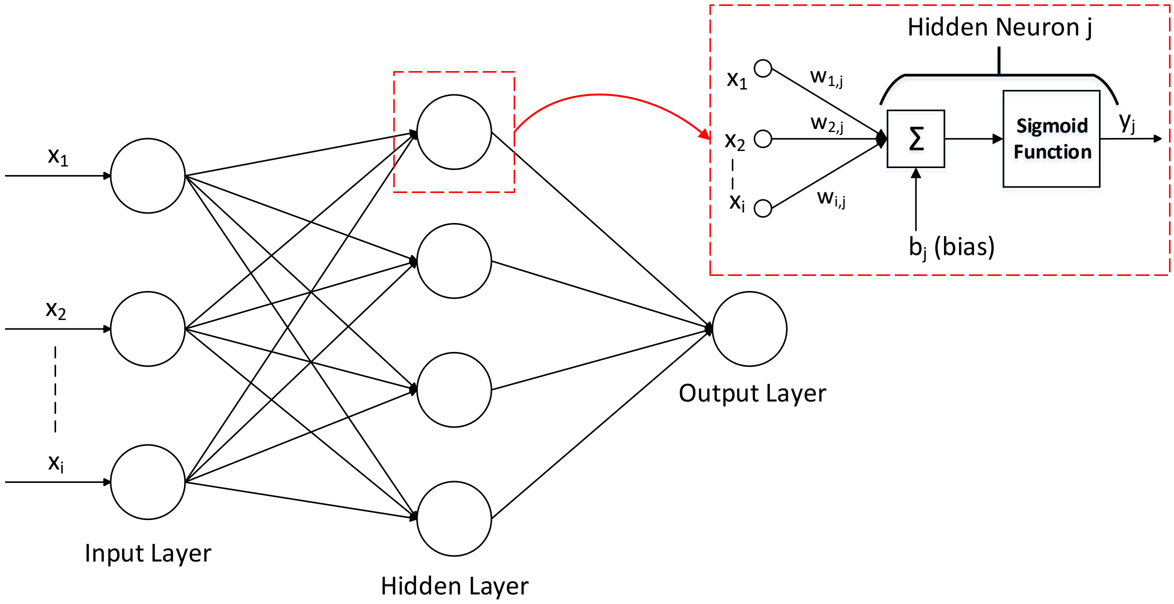

Figure 2.6 – Overview of a single hidden layer feedforward neural network

2.4.4 Extreme learning machine

Extreme Learning machine (ELM) is a modification of a single hidden layer feedforward neural network, as proposed by Huang *et al.* (2006), where the input weights and bias to the hidden layer are randomised, instead of being optimised via training. Traditional neural network training involves the use of backpropagation algorithms that are slow and can fall into local minima. ELM has been shown to overcome both of these issues through the input weight randomisation, resulting in significantly faster training and improved generalisation performance, while still possessing the universal approximation capability (Huang *et al.*, 2006). These advantages make ELM ideal for soft sensing purposes where on-line training or model maintenance is necessary. ELM has been used by many researchers for process modelling (Zhang and Zhang, 2011; Li *et al.*, 2013; He *et al.*, 2015; Li *et al.*, 2017a).

There have been many adaptation to ELM, such as on-line sequential ELM (Liang *et al.*, 2006), kernel ELM (Huang *et al.*, 2012) and unsupervised ELM (Huang *et al.*, 2014). Huang *et al.* (2015) presented a review of recent adaptations and applications of extreme learning machine, which discussed these in more detail.

2.5 Ensembles

One of the most widely used methods for improving reliability and robustness of data-driven models is to create an ensemble. This involves training multiple models and then combining their predictions in some way. The simplest form of an ensemble is to average the predictions of the multiple models. Valentini and Masulli (2002) explained the reasoning behind the advantages of ensembles, as well as the types of ensembles and methods used to create them. Two of the first, and still widely used, ensemble methods are bagging and boosting. Bagging (bootstrap aggregating) (Breiman, 1996) involves producing multiple, independent training data sets through random bootstrap sampling with replacement, with a model then trained using each data set. The predictions of each model are then aggregated through (weighted) averaging. The bootstrap resampling is a method of creating a diverse ensemble, a key factor in improving the generalisation performance of a model (Brown *et al.*, 2005). Bagging has been successfully applied to neural network ensembles to improve model robustness for nonlinear process modelling (Zhang, 1999; Niu *et al.*, 2011; Yang *et al.*, 2013; Li *et al.*, 2015). Zhang *et al.* (1997) demonstrated that bagged neural networks can be used to calculate confidence bounds for the model predictions, providing a measure of the model reliability. Bagging has also been used with other machine learning techniques for process modelling, such as extreme learning machine (Zhang *et al.*, 2018b) and support vector regression (Hu *et al.*, 2011).

Boosting involves iteratively training each model such that subsequent models are trained in areas where previous models were weak. The notion of boosting was introduced by Schapire (1990), with Freund and Schapire (1997) proposing the AdaBoost boosting algorithm, which still remains popular to this day. Bauer and Kohavi (1999) investigated the performance of boosting for voting classification and found that AdaBoost was not very robust to noise, which may be an issue for process modelling because process data often contains significant noise.

Other ensemble methods applied to data-driven modelling include sequential orthogonal training (Zhang and Morris, 1998), data fusion techniques (Ahmad and Zhang, 2005b), negative correlation learning (Liu and Yao, 1999; Zhang *et al.*, 2006), and forward selection/backward elimination combined with Bayesian methods for the model combination (Ahmad and Zhang, 2005a). Zhou *et al.* (2002) demonstrated a mathematical proof that, in many cases, it may be better to combine some of the total models, but not all of them. The authors stated that certain “bad” models may need a constraint that means they should be excluded from the ensemble. This shows that selective ensemble algorithms are better than the other commonly used approaches. Finally, Ahmad *et al.* (2009) presented a review on neural network ensemble techniques for process modelling and control, which covered a large amount of ensemble techniques and there applications.

2.6 Dynamic Modelling

To capture the dynamic nature of industrial processes, dynamic models can be created by incorporating past time samples into the model, which can account for time-lagged correlations between inputs and hence improve model performance. The two main methods to create data-driven dynamic models are based on internal or external dynamics. External dynamics-based methods involve augmenting the input data with time lagged samples of the inputs. Often the selected time lags are determined by correlation analysis, process knowledge or simply trial and error (Fortuna, 2007). Internal dynamics-based methods are typically in the form of a recurrent neural network (RNN), where the outputs of network layers are connected to the inputs of the layers. In particular, locally recurrent neural networks can be used for accurate long range predictions for nonlinear processes, as shown by Zhang *et al.* (1998) and Zhang and Morris (1999).

RNNs can be further extended to deep neural networks (DRNN), as shown by Chang and Li (2021), where the authors integrated DRNN into soft sensor development for a wastewater treatment process in order to better capture the temporal correlations in the data when compared with standard RNNs.

RNNs suffer from gradient vanishing and exploding problems (for descriptions of these issues, see work by Pascanu *et al.* (2013)), however, the long short-term memory (LSTM) network was developed to overcome these problems (Hochreiter & Schmidhuber, 1997). Several approaches have been developed to improve RNNs and LSTM networks for soft sensor modelling (Curreri et al., 2021; Moreira de Lima & Ugulino de Araújo, 2021). Yuan et al. (2020) proposed a supervised LSTM network to focus on learning the quality-relevant dynamics as opposed to just capturing the dynamics of the input variables. Addtionally, deep learning, particular convolutional neural networks, can enhance dyanmic soft sensor performance (Wang et al., 2019).

Chapter 3. Integrating Dynamic Slow Feature Analysis with Neural Networks for Enhancing Soft Sensor Performance

3.1 Introduction

Many industrial processes are extremely complicated and so developing mechanistic models can be extremely time consuming and expensive. Additionally, sufficient knowledge may not be available to produce a truly accurate mechanistic model. If sufficient process data is available, then data-driven modelling can overcome some, if not all, of these issues. The main advantages of data-driven over mechanistic modelling has been discussed in Section 2.2.

However, there are issues with data-driven soft sensors, such as lack of data covering the full process conditions, model drift, and the difficulty in creating a sufficiently robust model. One significant way to improve the accuracy and robustness of a soft sensor is in developing/selecting the right technique. Due to the complexity of industrial processes, many of the inherent relationships between process variables are nonlinear and so nonlinear models should be the main choice, although it is always problem dependent and if a less complex linear model is sufficient, then it should be employed.

The selection of nonlinear modelling techniques is vast and, with the era of machine learning and artificial intelligence rapidly growing alongside hardware improvements, the selection keeps increasing. Neural networks (NN) have always been one of the most used techniques for soft-sensing ever since they were first used for inferential estimation by Willis *et al.* (1991). Other machine learning algorithms that are commonly used include support vector machine (Wang *et al.*, 2020a) and extreme learning machine (Huang *et al.*, 2006; Xie *et al.*, 2020).

Selection of the right technique goes a long way to improving model robustness and accuracy. Choosing the correct technique, such as from those listed above, is one step. However, these techniques can be extended to further enhance the model performance. One way to do this is to combine the regression techniques with other statistical methods, often for dimensionality reduction, noise reduction or feature extraction. For example, applying principal component analysis (PCA) (Jolliffe, 2002) before a regression technique can remove collinearity in the data and improve model robustness, particularly when applied prior to a neural network due to their sensitivity to collinearity (Qin, 1997).

Another latent variable method that has gained attraction in the process systems engineering field in recent years is slow feature analysis (SFA). While PCA transforms the inputs into a new feature space with reduced collinearity, SFA transforms the input variables into a set of latent variables that are as slowly varying as possible (Wiskott and Sejnowski, 2002). These latent variables are known as slow features (SFs) and they can capture important underlying trends in the data, which is particularly useful for time series data where time varying trends and dynamics can be observed. The slowest SFs capture the most important trends, while the fastest ones mostly represent noise. This can be beneficial to process modelling because underlying dynamics of the process can be extracted with a de-noising effect when a reduced number of SFs is selected. Additionally, by retaining a reduced number of SFs to be used for modelling, the dimensionality can be reduced, leading to a reduction in model complexity. By using a reduced number of inputs that express inherent dynamics, combining SFA with a data-driven modelling technique can improve model generalisation performance.

SFA has been previously integrated within soft sensor design (Qin *et al.*, 2019; Yuan *et al.*, 2020b), as well as for process monitoring (Zhao and Huang, 2018; Qin and Zhao, 2019; Yu and Zhao, 2019; Zhang *et al.*, 2019; Zhang and Zhao, 2019; Zheng and Zhao, 2019; Xu and Ding, 2021), and process control performance monitoring (Shang *et al.*, 2016a). Other extensions of SFA have been applied in process system engineering, such as kernel slow feature discriminant analysis for nonlinear process fault diagnosis (Zhang *et al.*, 2015a), and a robust probabilistic SFA based regression model applied to process data contaminated with outliers (Fan *et al.*, 2018). Shang *et al.* (2015a) proposed a probabilistic SFA that used the SFs as inputs to a linear regression model, demonstrating improved soft sensor performance in two case studies when compared with other methods, such as PLS.

Other closely-related SFA applications include integrating SFA into an automatic detection and isolation of multiple oscillations algorithm (Wang and Zhao, 2020), combining canonical variate analysis with SFA to monitor process dynamics resulting from closed-loop process control (Zhang *et al.*, 2019), combining SFA with independent component analysis for time series feature extraction (Feng *et al.*, 2019), and creating a transition identification and trajectory-based monitoring scheme based on SFA (Wang *et al.*, 2020b).

SFA has also been used for dynamic feature extraction in other process monitoring applications. Zhao and Huang (2018) proposed a method combining SFA with cointegration analysis to produce a full-condition monitoring method that both the dynamic variations and the long term equilibrium relations. Recursive exponential SFA has been proposed to better extract nonlinear SFs when compared with standard SFA (Yu and Zhao, 2019). Other process monitoring based applications utilising SFA have been investigated in the areas of fault classification (Chai and Zhao, 2020) and batch process monitoring (Zhang and Zhao, 2019).

Since many processes display nonlinear characteristics, using LR with SFA will often lead to poor model performance. For example, Qin *et al.* (2019) developed a quality-relevant slow feature regression algorithm that aimed to improve soft sensor predictions when compared with traditional slow feature regression (SFR). However, an application on a complex process demonstrated that the prediction error was quite high even when the proposed method was better than the other methods that were used for comparison. One reason is that nonlinearities were not accounted for, as the authors themselves discussed in this work. This demonstrates the need to capture the nonlinearities in such processes, as well making use of the slow varying trends.

The method proposed in this chapter utilises dynamic linear SFA as a step prior to a neural network for dynamic soft sensing applications, with the aim of improving generalisation over traditional methods. Dynamic SFA is first applied to the process data and the number of retained SFs is selected using the scree plot, which is still widely used as a method for selecting retained principal components in PCA. The retained SFs are then used as inputs for a SLFNN. The efficacy of the proposed method is demonstrated by its application on two real industrial processes.

The rest of this chapter is organised as follows: Section 3.2 describes slow feature analysis with a numerical example illustration. Section 3.3 details the proposed method. Section 3.4 presents the application of the proposed method for soft sensor modelling in an industrial debutaniser column and Section 3.5 demonstrates a second application for a polymer melt index soft sensor in an industrial polymerisation process. Finally, the conclusions of this work are presented in Section 3.6. The work in this chapter is based on Corrigan and Zhang (2020).

3.2 Overview of Slow Feature Analysis

Slow feature analysis is an algorithm developed by Wiskott and Sejnowski (2002) that extracts a set of latent variables from input data that are as slowly varying as possible, without being constant. For example, if a signal that is a combination of sine signals of different frequencies is input to slow feature analysis, the slow varying variables to be extracted would be those original sine functions (see example in Section 3.2.1).

The main benefits of slow feature analysis are twofold. Firstly, these slow varying latent variables, known as SFs, can represent low dimensional driving forces within a data set. Here, the terms “driving force” and “underlying trend” are synonymous. They are low dimensional, uncorrelated, time series features that cause the variations shown in the original high dimensional input signals (e.g. the sine example described above and in Section 3.2.1).

Secondly, extracting SF scan reduce the amount of noise within the input data because slower features are inherently less noisy.

The algorithm takes an n-dimensional input signal x(t)=[$x_1(t)$ $x_2(t)$ … $x_n(t)$] and aims to find a function s(t) = g(x) where the output **s**(t) varies as slowly as possible, without being constant. This is because SFs that are constant provide little information about the driving forces behind data. The dimension of **s**(t) is up to n, i.e. $s_j(t)=g_j(\mathbf{x})$ with j=1, 2, …, n.

The optimisation problem for determining the j^{th} slow feature is described in the following:

$$\min_{g_j(.)} \langle \dot{s}_j^2 \rangle = var(\dot{s}_j) \;, \tag{3.1}$$

subject to the following three constraints, i.e. $\boldsymbol{s}$ has a zero mean, unit variance, and is decorrelated (identity covariance matrix):

$$\langle s \rangle = E(s) = 0\,, \tag{3.2}$$

$$\langle s_j^2 \rangle = var\big(s_j\big) = 1\,, \tag{3.3}$$

$$\langle s_j, s_i \rangle = cov(s_j, s_i) = 0, \qquad i \neq j \tag{3.4}$$

where $\dot{s}$ represents the first order derivative of the SFs with respect to time and $\langle s \rangle$ signifies temporal averaging given as,

$$\langle s \rangle = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} s(t)\, dt\,. \tag{3.5}$$

The aim of (3.1) is to minimise the temporal variation of the extracted SFs, that is, they vary as slowly as possible. Constraints (3.2) and (3.3) are there to ensure that a solution for s_j being constant is avoided, while additionally allowing a fair comparison of the slowness of each feature due to all features being scaled to unit variance. Constraint (3.4) ensures that the SFs are decorrelated and hence are independent, meaning that certain features are not simply repeated. This leads to a descending order of s_j where the slowest varying features have the lowest index, i.e. s_1 is the slowest feature. The problem is complicated because it is a variational calculus optimisation problem.

3.2.1 Linear SFA

In order to solve the difficult optimisation problem, Wiskott and Sejnowski (2002) proposed an algorithm that simplifies the problem greatly.

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

$$s(t) = Wx(t)\,. \tag{3.6}$$

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

$$\langle \dot{x}\dot{x}^T \rangle W = \langle xx^T \rangle W\Omega\,, \tag{3.7}$$

where $\langle \dot{x}\dot{x}^T \rangle$ is the covariance matrix of the first order derivative of *x(t)*, $\langle xx^T \rangle$ is the covariance matrix of *x*, and $\boldsymbol{\Omega}$ is a diagonal matrix of the generalised eigenvalues, which are the optimal objectives of the objective function. For discrete data of sampling interval *h*, such as process data, a first order difference approximation can be used to approximate the first order derivative, that is:

$$\dot{x}(t) \approx \frac{x(t) - x(t-h)}{h}\,. \tag{3.8}$$

The first step in solving the problem involves normalising the input signal to zero mean and unit variance. The normalised input, $\mathbf{x}(t)$, is then whitened to remove underlying correlations, giving the whitened matrix $\mathbf{z}(t)$. The next step is to carry out singular value decomposition (SVD) on the matrix $\langle \dot{\mathbf{z}}\dot{\mathbf{z}}^{\mathbf{T}} \rangle$, the covariance matrix of the first order derivate of $\mathbf{z}(t)$, giving

$$\langle \dot{z}\dot{z}^T \rangle = P_{SFA}\Omega {P_{SFA}}^T\,, \tag{3.9}$$

where P_{SFA} is the eigenvector and Ω is a diagonal matrix of the eigenvalues.

From the SVD, the SFs can be determined by the following:

$$s(t) = P_{SFA} z(t)\,. \tag{3.10}$$

The working of linear SFA can be nicely illustrated through a simple numerical example. Four inputs are defined by Equations (3.11) to (3.14) with t = 0 to 2π. Noises (v_1 to v_4) with distribution N(0,0.2) are added to these inputs. These noisy inputs are displayed in Figure 3.1.

$$x_1(t) = -5\sin(t) - 2\cos(5t) + v_1\,, \tag{3.11}$$

$$x_2(t) = 3\sin(\mathrm{t}) + 0.75\cos(5t) + v_2\,, \tag{3.12}$$

$$x_3(t) = -4\sin(t) + \cos(5t) + v_3\,, \tag{3.13}$$

$$x_4(t) = 6\sin(t) - \cos(5t) + v_4\,. \tag{3.14}$$

By performing linear SFA with these four input signals, the underlying driving forces behind the inputs can be extracted as SFs, which in this case are sin(t) and cos(5t). The SFs were derived by collecting the four inputs, as described by Equations (3.11) to (3.14), into a single matrix *x(t)* and applying the SFA method as detailed in Section 3.2.1. The derived SFs were therefore calculated by multiplying the derived P_{SFA} matrix by the whitened inputs $z_i(t)$, as shown in Equation (3.15):

$$\begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ s_4(t) \end{bmatrix} = \begin{bmatrix} -0.47 & -0.65 & 0.58 & -0.10 \\ 0.48 & 0.36 & 0.80 & -0.01 \\ -0.49 & 0.51 & 0.06 & -0.70 \\ 0.55 & -0.42 & -0.14 & -0.70 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \\ z_3(t) \\ z_4(t) \end{bmatrix} \tag{3.15}$$

where $z_i(t)$ is the whitened signal of the input $x_i(t)$, $s_j(t)$ is the jth derived slow feature and P_{SFA} is the eigenvector matrix derived from the SFA algorithm. These four SFs are displayed in Figure 3.2.

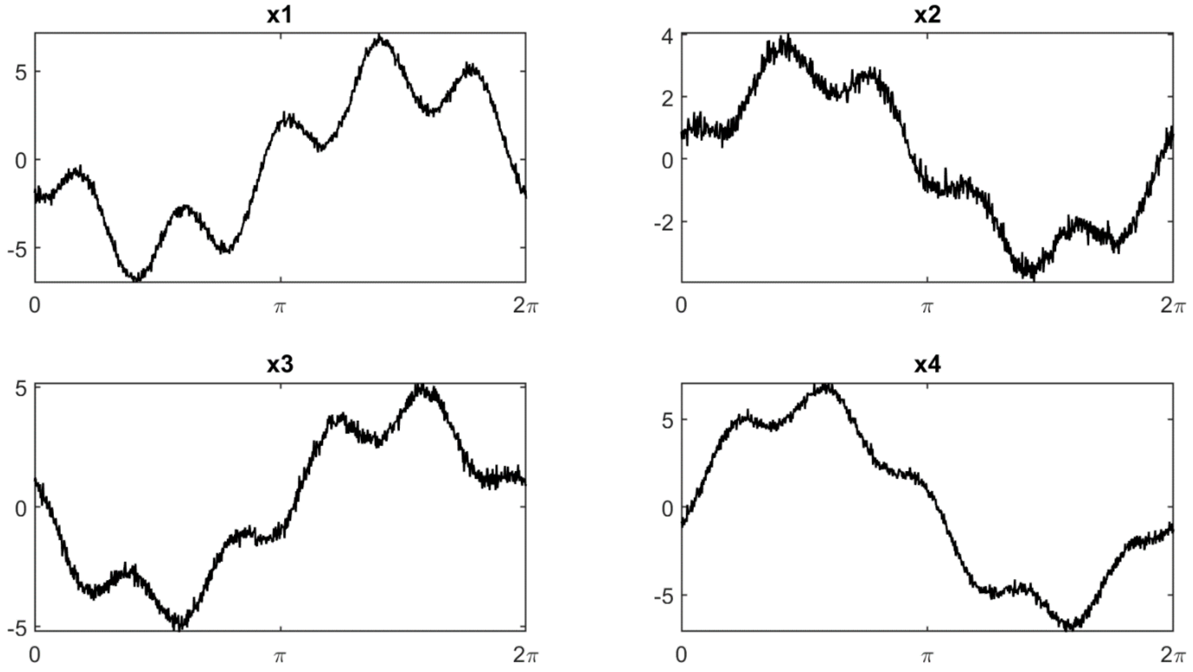

Figure 3.1 – Inputs for the numerical example

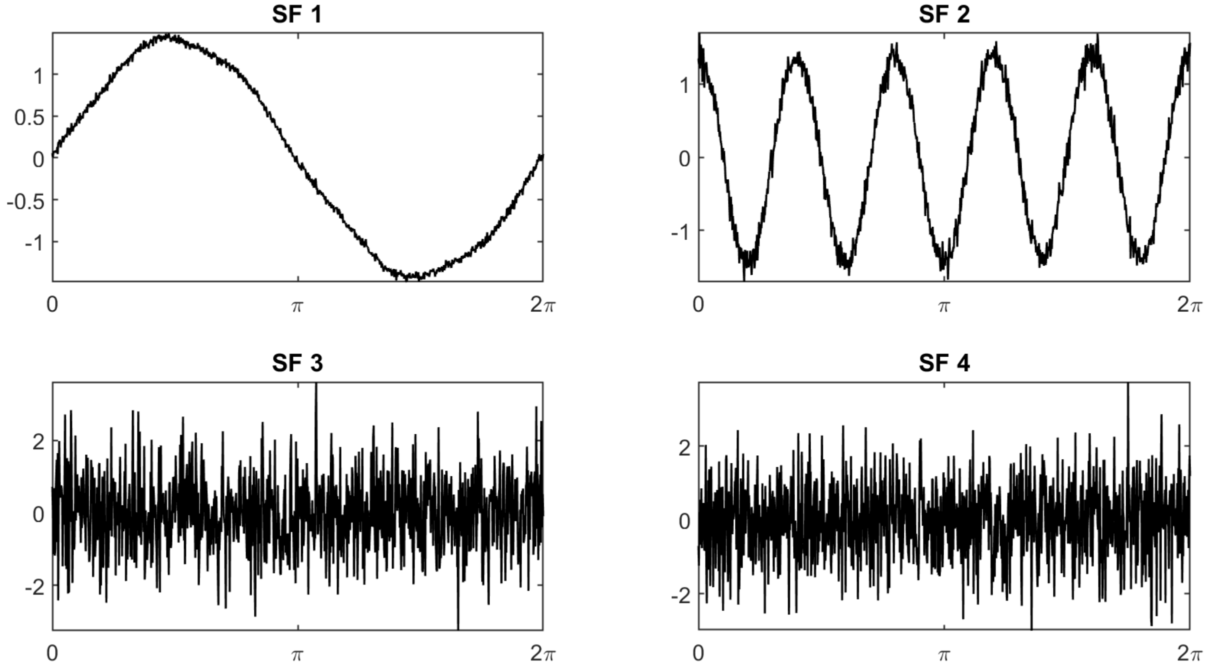

Figure 3.2 – Extracted slow features for numerical example

Figure 3.2 shows that the first two SFs are approximately sin(t) and cos(5t) respectively, however, there are some oscillations which appears to be the way that the algorithm has interpreted the noise in the inputs. When this example is repeated without any noise, the first two SFs are exactly sin(t) and cos(5t), demonstrating that linear SFA can indeed extract the underlying driving forces behind a set of inputs. The reason that the last two SFs are purely noisy is because linear SFA has already extracted the two driving forces in the first two SFs, leaving the remaining SFs as noise since no driving forces remain to be extracted.

Without knowing the exact equations for the inputs, it may not be obvious that there are only two driving forces, and so looking at the determining the number of dominant SFs can help this. The dominant SFs are simply the number of SFs that best capture underlying trends, with the remaining SFs being mostly noise. To determine the number of dominant SFs, inspection of a plot of the eigenvalues, found from the Ω matrix in Equations (3.7) or (3.9), can be carried out. The number of dominant SFs can be determined by the SFs up until an “elbow” in this plot. The selection of dominant SFs is discussed in more detail in Section 3.3.

For the example described in this section, Figure 3.3 shows a clear elbow in the eigenvalues after two SFs and therefore there are only two dominant SFs, representing the two driving forces of sin(t) and cos(5t).

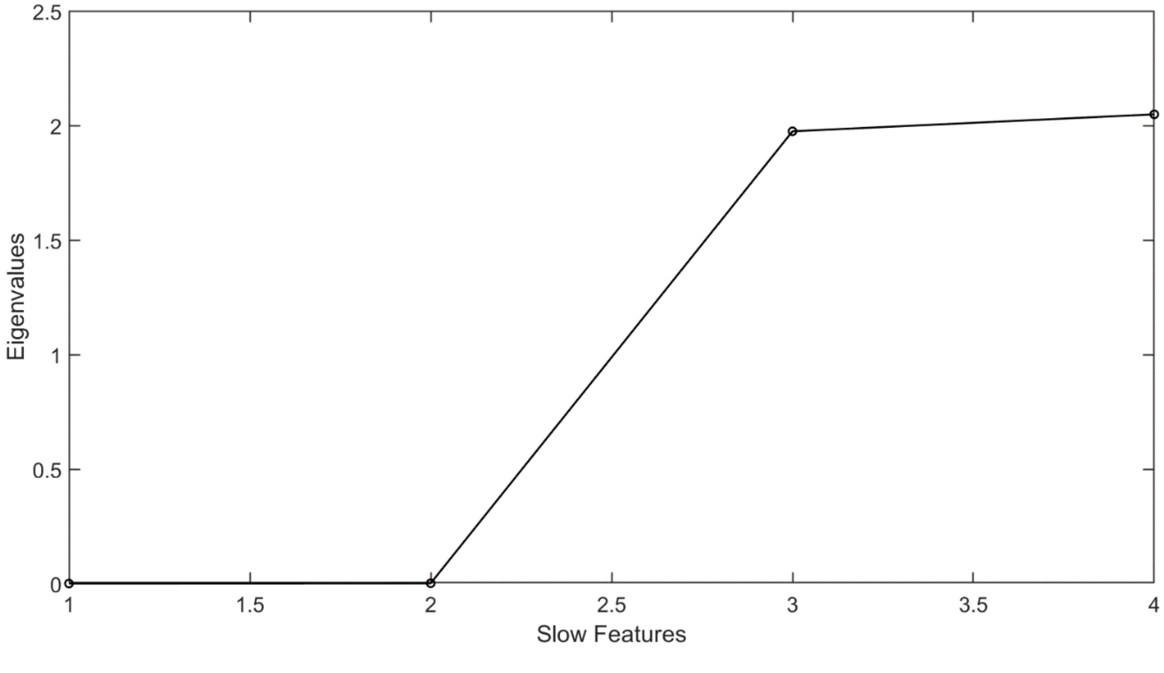

Figure 3.3 – Eigenvalue plot for the sine example

3.2.2 Dynamic SFA

The relationship between the inputs and outputs can involve significant time delay for many industrial processes and so time lagged process variables can be included to improve the extraction of SFs.

Dynamic SFA is simply the inclusion of time lagged process inputs into the data matrix for SFA. For a selected time lag, *d*, the input matrix for SFA is given by the following:

$$X(t) = \begin{bmatrix} x(t) & \cdots & x(t-d) \\ \vdots & \ddots & \vdots \\ x(t+N-1) & \cdots & x(t+N-d-1) \end{bmatrix} \tag{3.16}$$

where $x(t)$ is a vector of inputs at time *t* and *N* is the number of samples. The selection of time lag, *d*, can be carried out through correlation analysis, cross validation, or utilising process knowledge (Zhang et al., 2015b; Zhao and Huang., 2018). In this study, cross validation is used to determine the time lag. A range of time lags are considered and the one giving the least model error on the testing data is selected as the appropriate time lag. This approach is taken to avoid overfitting.

3.3 Proposed Methodology: Combining Slow Feature Analysis with a Neural Network (SFA-NN)

In this section, the proposed SFA-NN method is detailed. A simplified block diagram showing the transformation of data in the method of SFA-NN is illustrated in Figure 3.4. The proposed methodology contains the following steps.

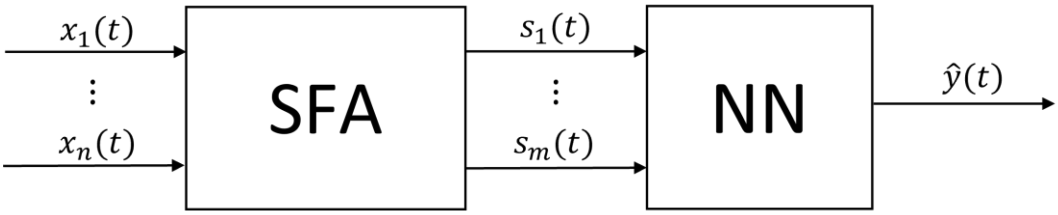

Figure 3.4 – Simplified block diagram of SFA-NN

Step 1 – Dynamic Data Preparation:

Matrices of the process input variables ***X*** and output variables ***Y*** are collected. Time lagged inputs up to the selected time lag *d* are added to the input data matrix for dynamic modelling, as detailed in Section 3.2.2.

Step 2 – Data Partitioning and Scaling:

The data matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ are partitioned into a training and testing (TT) set ($\boldsymbol{X_{TT}}$, $\boldsymbol{Y_{TT}}$) and an unseen validation set ($\boldsymbol{X_V}$, $\boldsymbol{Y_V}$). The TT data set is then normalised to have zero mean and unit variance. The means and standard deviations from the TT data set are used to normalise the unseen validation data set.

Step 3 – Applying Linear SFA:

The linear SFA algorithm, as described in Section 3.2.1, is applied to the normalised $\boldsymbol{X_{TT}}$ matrix to derive the P_{SFA} matrix and extract the SFs $\boldsymbol{s_{TT}}$.

Step 4 – Selection of Dominant Slow Features:

Given a set of *m* extracted SFs, it is important to select the number of dominant SFs, *M*, that best capture the dynamics of the process since many of the faster features represent mostly noise. Including the faster features can decrease generalisation performance because the model is fitting the noise as opposed to the underlying trends. Additionally, decreasing the number of SFs reduces the model complexity as the neural network model will have a smaller number of inputs.

There are few standard procedures for the selection of the dominant SFs but there are some ways this can be accomplished that have been attempted in previous work. One standard method for selecting latent variables in general is using cross validation. This method has been applied to slow feature regression in previous work (Shang *et al.*, 2015a; Qin *et al.*, 2019). Cross validation can be effective when the number of latent variables is the only parameter to be determined. However, when using neural networks for regression, cross validation is typically used for selecting the optimal number of hidden neurons. Therefore, additionally using cross validation for slow feature selection can lead to poor generalisation, especially for complex processes where it can be difficult to obtain the optimal values for such hyper parameters.

A slowness criterion was developed by Shang *et al.* (2015c) that was based on a de-noised reconstruction of the inputs. This method suggested discarding SFs that are faster than all the input variables.

Looking towards PCA can provide inspiration for selecting dominant SFs due to the similarities between SFA and PCA. These similarities arise from the fact that eigenvalues are extracted in both algorithms, albeit via different pathways, and that they provide key information about the latent variables. In PCA, the eigenvalues detail the variance explained by each PC, whereas in SFA, the eigenvalues relate to the slowness of each SF. Jolliffe (2002) described some of the commonly used methods of PC selection, such as a cumulative percentage of total variation cut off, Kaiser's rule (Kaiser, 1960), the scree plot and cross validation. Of these methods that could be translated over to SFA, the scree plot, as first proposed by Cattell (1966), shows promise. A scree plot involves a plot of the eigenvalues and visually inspecting the plot to see if there are any significant changes in direction, often referred to as "elbows", or significant changes in magnitude, referred to as "gaps". The number of retained latent variables is then selected by the features up until the first elbow.

In this work, using the scree plot for dominant slow feature selection is proposed. After an elbow/gap in the scree plot for the SFA eigenvalues, it can be said that any SFs after this point contain too much noise and do not contain enough relevant information. To obtain the eigenvalues for the scree plot, the Ω matrix from Equation (3.9) can be extracted and this matrix contains these eigenvalues. Figure 3.5 illustrates an example of the scree plot for SFA. In this example, the first elbow occurs at 6 SFs and therefore 6 dominant SFs would be selected.

Utilising the scree plot is not without its downfalls however. By nature, it is a subjective method and so there can be issues in reliability due to individual opinions in what consitutes the first elbow. Despite this issue, the scree plot is still widely used for PC selection because it gives a good solution more often than not. In Figure 3.5, the elbow point is very clear and so there is little issue in this case, however, it is not always so clear cut and so in such cases it may be necessary to use this method to obtain an approximate solution that can be verified or refined with other methods, such as cross validation. The application examples in the next section show that selecting the number of domainant SFs using the scree plot gives good modelling performance.

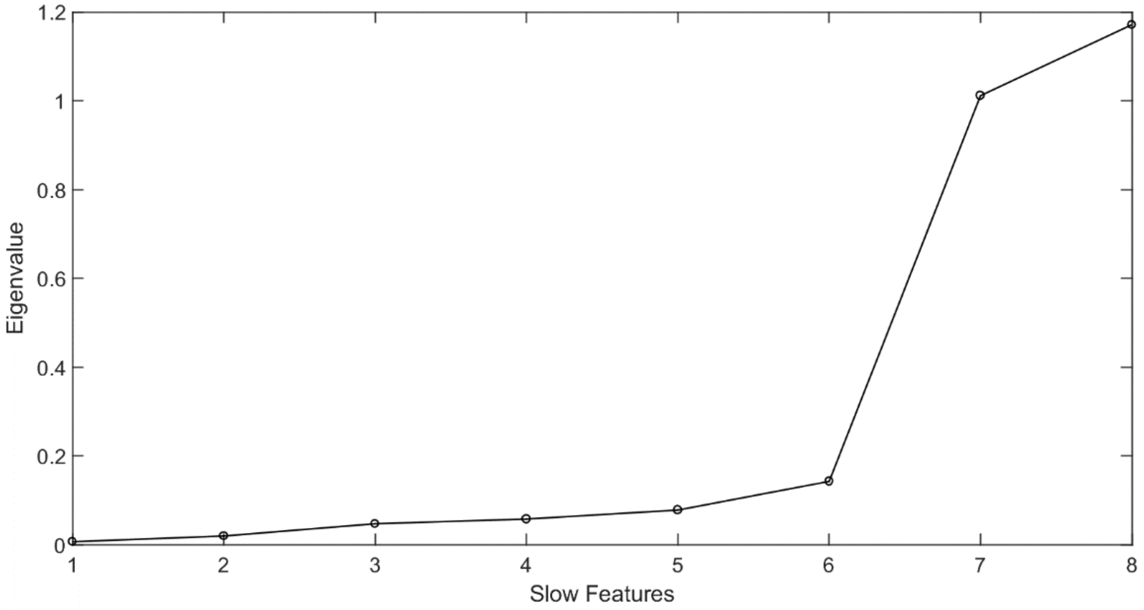

Figure 3.5 – Eigenvalue trend for slow features used for dominant slow features selection

Step 5 - Neural Network Training:

Following the application of SFA, the M retained SFs together with the corresponding target values are then further partitioned into training and testing data sets. 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. The optimal number of hidden neurons is then selected using cross validation on the testing data set. The neural network used here is a SLFNN that is trained using the Levenberg-Marquardt algorithm with regularisation and early stopping, to aid in preventing overfitting. During network training, the neural network model errors on the testing data are monitored and training is terminated when the testing errors stop decreasing.

Step 6 – Model Evaluation:

The normalised validation data set $\boldsymbol{X_V}$ is sphered (synonymous to whitening) to give $\boldsymbol{Z_V}$. Sphering removes correlations between signals and can be done via algorithms such as PCA. Normalised refers to the scaling of the original data to zero mean and unit variance. Slow features for this data set are calculated based on the previously derived P_{SFA} matrix: $s_V = P_{SFA}Z_V$. These validation SFs are applied to the trained neural network model to assess the model's performance on unseen validation data to assess its generalisation capability. The root mean squared error (RMSE) can then be calculated to measure the model accuracy. RMSE is chosen as the metric for comparison due to it being in the units of the original measurement and so its magnitude is easily interpretable. Furthermore, due to the squaring of the errors, RMSE gives a higher weight to larger errors, which is desirable because large errors are unwanted in soft sensor predictions.

Figure 3.6 provides a summary of the key steps in the SFA-NN model development methodology.

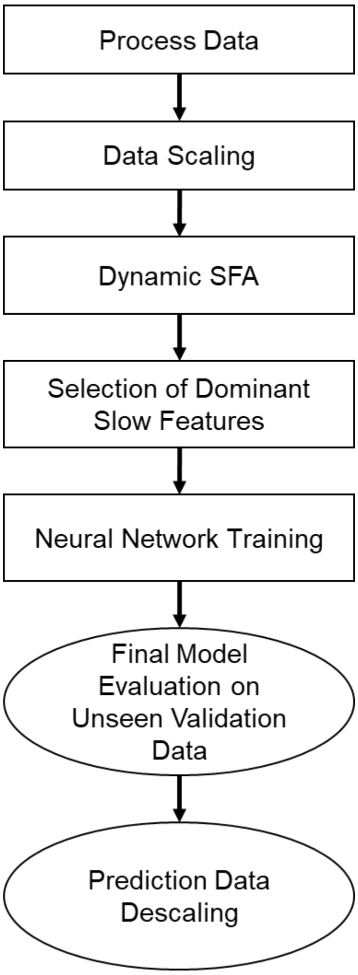
Process Data
Data Scaling
Dynamic SFA
Selection of Dominant Slow Features
Neural Network Training
Final Model Evaluation on Unseen Validation Data
Prediction Data Descaling

Figure 3.6 – Simplified methodology of the development of SFA-NN models

3.4 Case Study 1: An Industrial Debutaniser Column

3.4.1 Process description

The process used for this first case study is a debutaniser column that is a part of an industrial desulfuring and naphtha splitting plant, as shown in Figure 3.7. Propane (C3) and butane (C4) are removed as overheads from the debutaniser column. This process has been used previously for soft sensor development by Fortuna (2007). One of the key requirements of the column is to minimise the butane concentration in the Naphtha splitter feed coming from the bottom of the column. Therefore, measurements of this butane concentration are required as this stream leaves the bottom of the column, as indicated by location N2 in Figure 3.7. However, this butane concentration is measured by a gas chromatograph on the deisopentaniser column overheads, as shown by location A2 in Figure 3.7, which is significantly later in the process. This substantial downstream location and the inherent delay of gas chromatograph led to measurements with a large time delay, typically around 45 minutes, which is suboptimal for control purposes. A soft sensor using inputs to the debutaniser column is of interest so that measurements with a significantly reduced time delay can be produced to improve process monitoring and control.

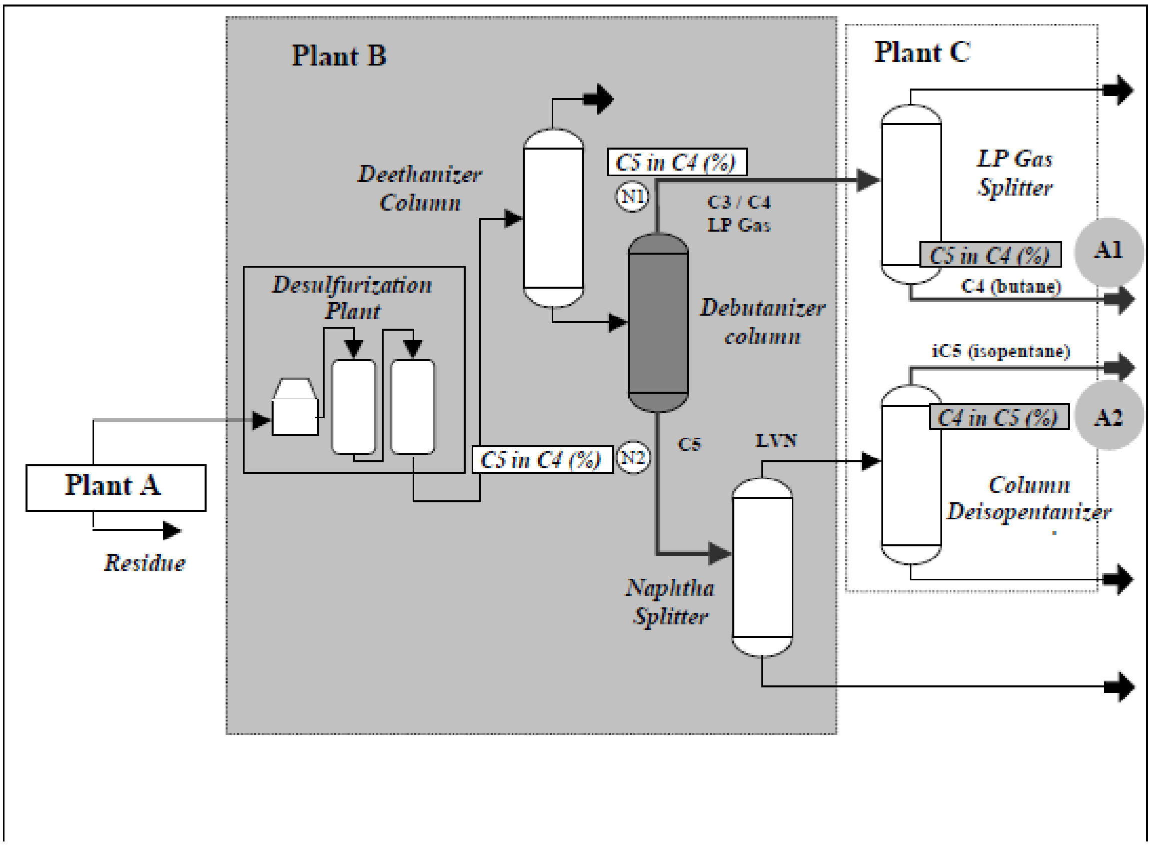

Figure 3.7 – Schematic of the debutaniser column and associated plants (Fortuna, 2007)

The process inputs used for the soft sensor design are measurements of various variables around the debutaniser column as described in Table 3.1 and illustrated in Figure 3.8, to provide background understanding of the process.

Table 3.1 – Variables used for the soft sensor design (Fortuna, 2007)

Soft Sensor Inputs	Variable Description
u_1	Column top temperature
u_2	Column top pressure
u_3	Reflux flow rate
u_4	Flow rate to the next unit
u_5	6th tray temperature
u_6	Column bottom temperature 1
u_7	Column bottom temperature 2

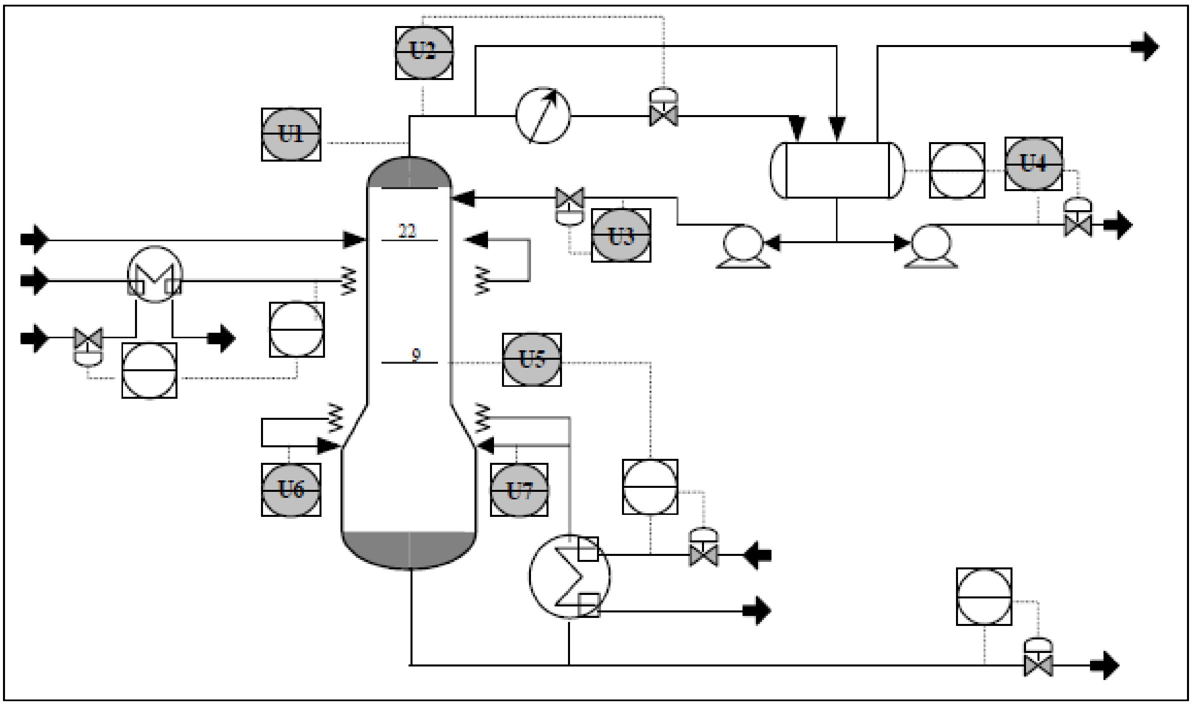

Figure 3.8 – Simplified process and instrumentation diagram of the debutaniser column with the soft sensor inputs highlighted (Fortuna, 2007)

3.4.2 Modelling of Butane Concentration

The process data used for soft sensor development contain 2394 samples with a sampling time of 6 minutes (Fortuna, 2007). The data is split into 60% for training, 20% for testing and 20% for unseen validation. The validation data was taken from samples 1650 to 2122. The process inputs data and normalised butane concentration data are displayed in Figures 3.9 and 3.10 respectively to highlight the amount of variation and noise in the data, which leads to poor soft sensor performance with many classical data-driven modelling methods.

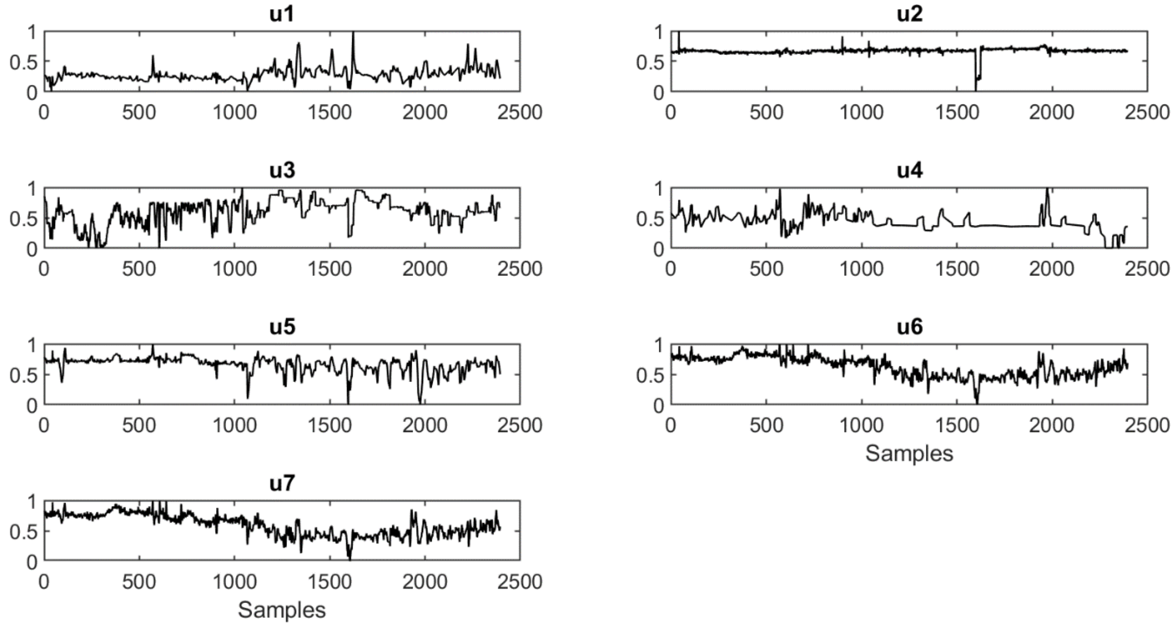

Figure 3.9 – All seven process variables for the full data set showing large variations

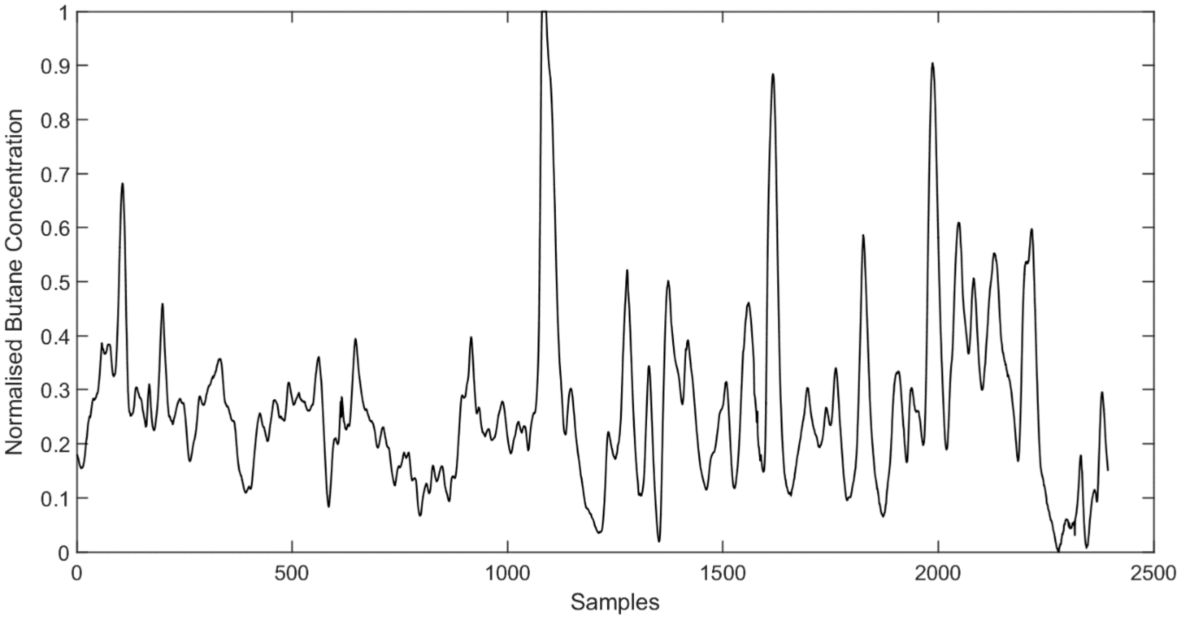

Figure 3.10 – Normalised butane concentration data

In complex chemical processes such as this, there can be time delays between inputs and outputs and so it is necessary to consider time lagged inputs. In this process, time-lagged inputs up to a discrete time lag of *d* were included into the model input. The proposed dynamic model structure is shown in Equation (3.17).

$$\hat{y}(t) = f(\, u_1(t)\ \dots\ u_7(t), \dots, u_1(t-d) \dots u_7(t-d)\,) \tag{3.17}$$

where u_i is the i^{th} process output, t is the discrete time, d is the time lag, and $\hat{y}$ is the soft sensor prediction.

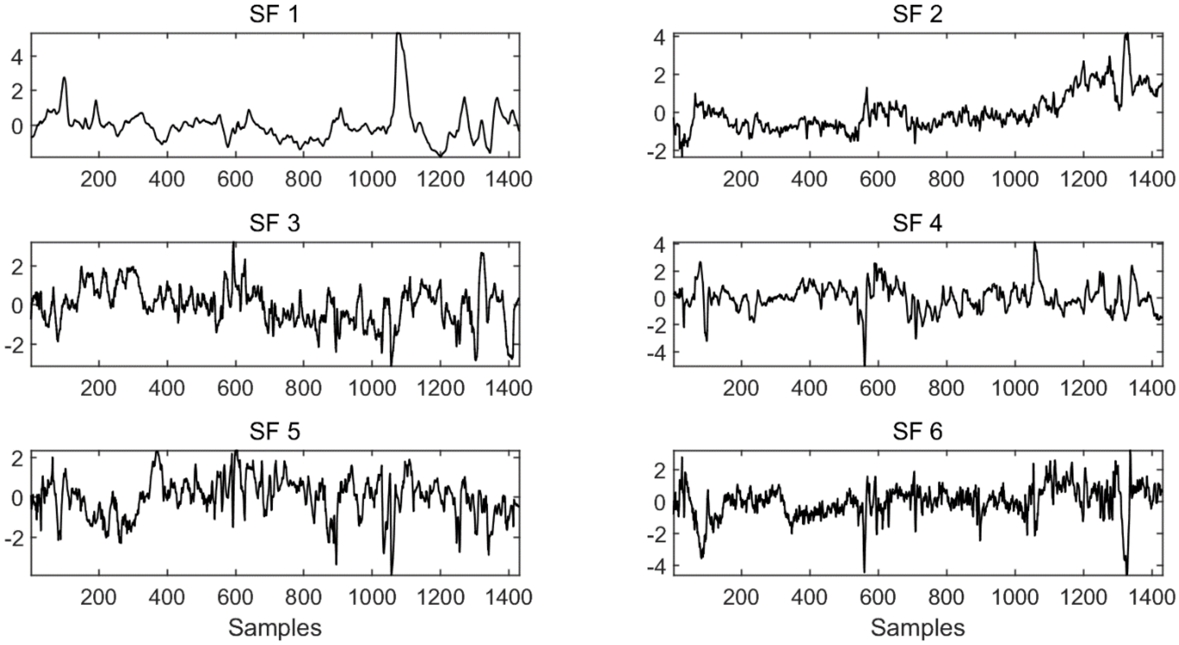

Figure 3.11 – The first 6 extracted slow features on the training and testing data sets for the debutaniser case study

Based on the proposed SFA-NN method, a soft sensor model is developed for butane concentration for each time lag *d* from 0 to 10. The model with the lowest RMSE on the testing data is considered as having the best value of time lag. To illustrate how SFA captures slow varying trends in the data, the first 6 extracted SFs for the training and testing data, when *d* = 0, are presented in Figure 3.11. It can be clearly seen that the first slow feature varies the least compared with the other five, with the amount of variation increasing with each slow feature, demonstrating that SFA does indeed extract slow varying trends with reduced noise. Several other similar algorithms are employed for comparison with the proposed method: the popular latent variable method PCA (Jolliffe, 2002) combined with neural networks (PCA-NN), the standard slow feature regression (SFR) algorithm (Shang *et al.*, 2015b), partial least squares regression (PLSR) (Wold *et al.*, 1984), and a single neural network (single NN). For PCA-NN, PCA is first applied to the TT data and the first *k* principal components (PCs) are retained using the scree plot method (Cattell, 1966). These retained PCs are then used as the inputs to a neural network. The neural networks used for single NN and PCA-NN are SLFNNs trained using the Levenberg-Marquardt algorithm with regularisation and early stopping. Regularisation and early stopping are necessary for this case study due to the complexity of the system and measurement noises in the data. Especially in this case study, significant regularisation was required (regularisation parameter of 0.5) to achieve good generalisation. Sigmoidal and linear activation functions are applied to the hidden and output layers respectively. Cross validation with the RMSE on testing data is used to select the number of hidden neurons. In the SFR algorithm, the method is the same as SFA-NN except that linear regression is used instead of a neural network. The soft sensor prediction results on unseen validation data, for each of the applied methods, are shown in Table 3.2 with the best results highlighted in bold. The proposed method shows significant improvement in generalisation performance over the other methods.

Table 3.2 – Soft sensor prediction performance for butane concentration on unseen validation data

Methods	RMSE	R^2	Hyper-parameters
Single NN	0.0896	0.7249	d = 10, #*Inputs* = 77
PCA-NN	0.0831	0.7634	d = 10, k = 7
SFR	0.0855	0.7493	d = 10, M = 18
PLSR	0.0863	0.7448	d = 10, k = 36
SFA-NN	**0.0802**	**0.7795**	d = 10, M = 18

Comparing SFR with SFA-NN demonstrates the needs for nonlinear regression methods for more complex processes as linear regression becomes insufficient for accurate predictions. Another interesting result to point out is that the performance of PCA-NN is second to the proposed method, showing the merit of employing latent variable methods as a precursor to a neural network. However, SFA-NN has better performance than PCA-NN, highlighting the advantages of using SFA over PCA. The R^2 on training and testing data for PCA-NN is 0.6950 and 0.6811 respectively, showing that, despite the good generalisation for this method, the selected number of PCs do not retain enough information for good predictions. For comparison, the training and testing R^2 for SFA-NN is 0.9289 and 0.9224 respectively, demonstrating that even better model performance can be achieved across different data sets, ensuring better generalisation performance. This issue arises from the use of the scree plot since it appears that the scree plot underestimates the number of PCs when the number of inputs is large (e.g. in the d = 10 case) leading to insufficient relevant information to produce good predictions. Whereas in SFA, although the scree plot also gives a relatively small number of SFs, this seems beneficial for SFA because too many inputs contain more noise and less relevant information. While in PCA, retaining more PCs means retaining more variance and so more relevant information. This highlights that SFA can capture more relevant information, and so produce better predictions, with fewer latent variables.

As an additional way to verify the soft sensor performance, visual inspection of these predictions on the unseen validation data can be performed through Figures 3.12 to 3.14. These figures demonstrate that the proposed method is better than other methods shown in Table 3.2.

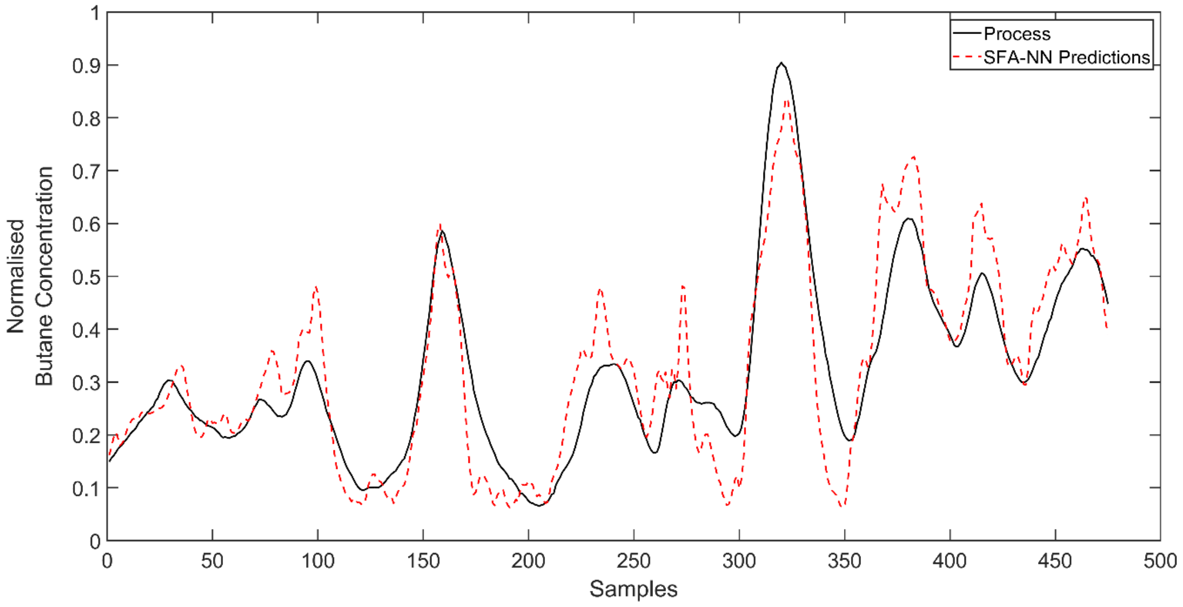

Figure 3.12 – Normalised butane concentration predictions on unseen validation data for the proposed SFA-NN method

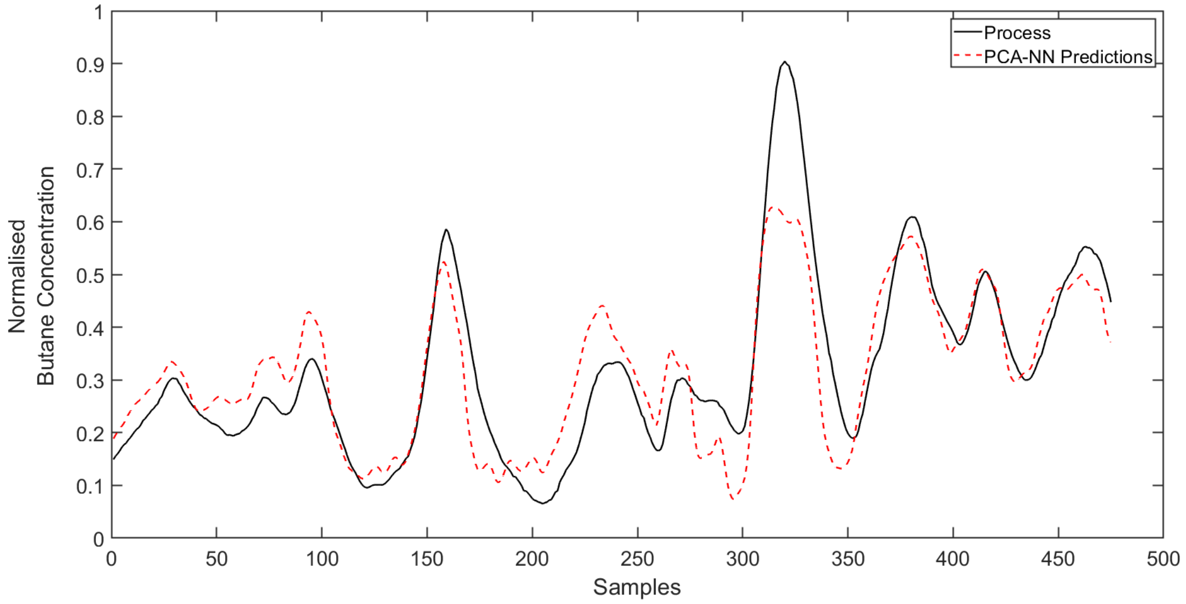

Figure 3.13 – Normalised butane concentration predictions on unseen validation data for PCA-NN

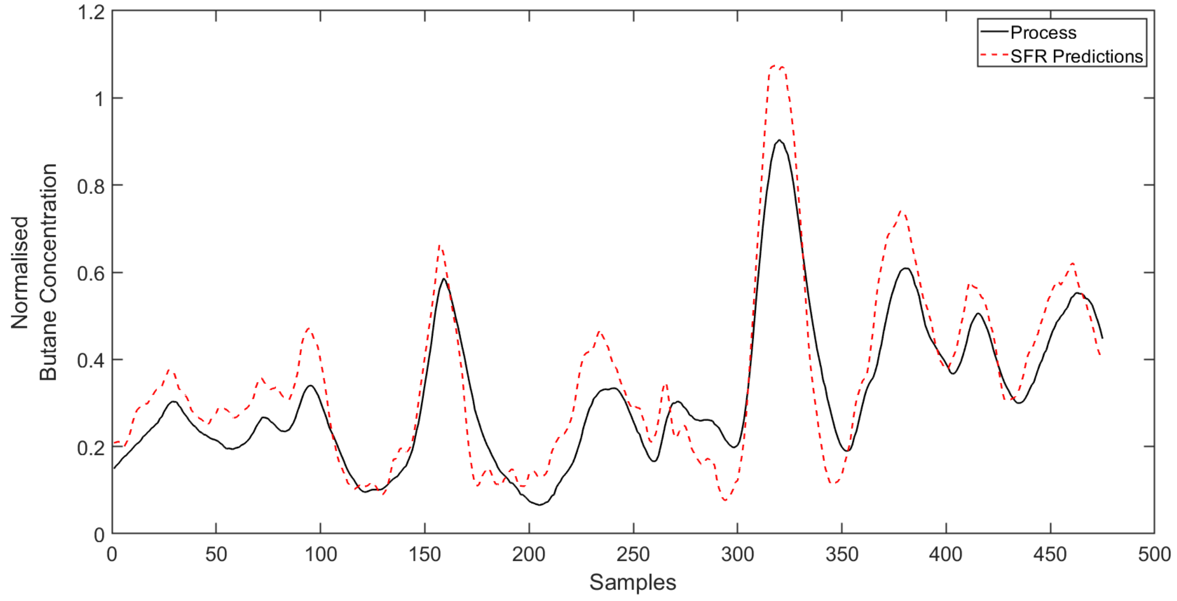

Figure 3.14 – Normalised butane concentration predictions on unseen validation for SFR

When comparing PCA and SFA in this case study, time lagged inputs are included to bring in the dynamics and improve model performance, however, this could blur the differences between PCA and SFA as one of the key benefits of SFA is to capture slowly varying dynamics. Therefore, SFA-NN was compared with PCA-NN with no time lagged inputs included (i.e. d = 0). These results are displayed in Table 3.3. The RMSE using SFA-NN is 6.15% better than

PCA-NN when $d = 0$, showing that including time lagged inputs into the model does not blur the differences between the two methods and that SFA-NN is the better method in both cases.

Table 3.3 – Comparison of SFA-NN with PCA-NN for $d = 0$ on the unseen validation data

Methods	RMSE	R^2	Hyper-parameters
PCA-NN	0.1190	0.5142	$d = 0, k = 6$
SFA-NN	**0.1119**	**0.5707**	$d = 0, M = 6$

To assess the merit of the scree plot for dominant slow feature selection, the generalisation performance for SFA-NN and SFR using the scree plot was compared with another slow feature selection method, the slowness of input reconstructions criterion proposed by Shang *et al.* (2015c). The results of this comparison are displayed in Table 3.4. For SFA-NN, the proposed scree plot gave an improvement in RMSE on validation data of 2.58% and for SFR, a small improvement of 0.23%. This shows that the scree plot method produces better generalisation for the proposed SFA-NN method and SFR. A key point to note here is that in both cases, the difference in RMSE is very small, highlighting that the selection of dominant SFs is not too crucial. The benefit of the scree plot is that it gives a lower number of dominant SFs, 18 versus 61, especially for problems like this where many time lagged inputs are considered and so the input matrix for the neural network is large. Retaining a lower number of dominant SFs further reduces model complexity as well as reducing noise. Additionally, the scree plot is simple and does not require further calculations as is the case with the slowness of input reconstructions method. The subjective nature of the scree plot remains its biggest potential downfall, however, with the small differences in RMSE shown in Table 3.4, this does not prove a significant issue.

Table 3.4 – Comparison of RMSE on unseen validation data using different dominant slow feature selection methods

SF Selection Method	RMSE for SFA-NN	RMSE for SFR	M
Scree Plot	**0.0802**	**0.0855**	18
Slowness of Input Reconstruction Criterion	0.0823	0.0857	61

3.5 Case Study 2: An Industrial Polymerisation Process

In this section, a second real industrial process is used to further demonstrate the performance of the proposed method and show that model robustness can be improved in a variety of problems. The following polymerisation process poses different challenges when compared with the first process, in particular the very large number of process inputs and the relatively small amount of data available.

3.5.1 Process Description

The process used in this section is based on an industrial propylene polymerisation process based in China, which has been used in previous work using bootstrap aggregated neural networks (Zhang *et al.*, 2006). Figure 3.15 shows a simplified diagram of the region of interest in the process consisting of two continuously stirred tank reactors (CSTR) and two fluidised bed reactors (FBR). The feed to the first CSTR's feed contains a mixture of propylene, hydrogen, and a catalyst. An important quality variable is the melt index (MI) of polypropylene in the reactor, which is affected by variables such as the composition of reactant, catalyst properties and reactor temperature. The hydrogen in the feed can regulate the molecular weight of propylene and adjusting the hydrogen in the feed rate can be used to control the MI of polypropylene. The MI is difficult to measure because it requires offline analysis that leads to measurements being recorded every two hours. This infrequent sampling and time delay in measurement leads to suboptimal control of the MI. Being able to produce a soft sensor from process variables that are frequently measured with minimal delay will yield improvements in product quality through better control of the MI.

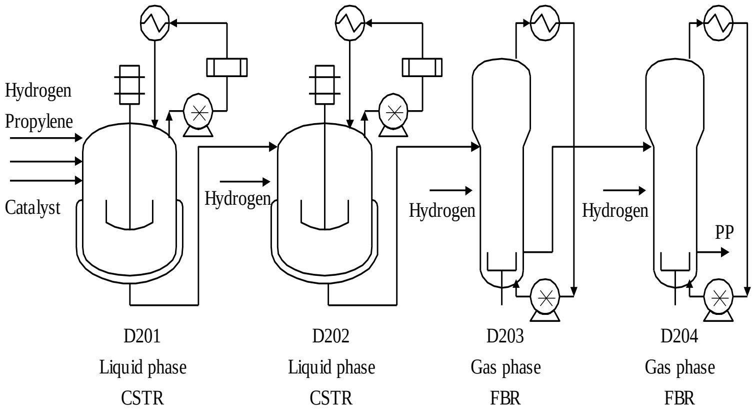

Figure 3.15 – A simplified diagram of the industrial polymerisation process (Zhang et al., 2006).

3.5.2 Modelling of MI

The data provided covered 31 days with MI measurements in reactors 1 and 4 for every 2 hours, and measurements of 30 process variables for every half an hour. For this case study, a soft sensor is built for the MI in reactor 1. The data is split into 50% for training, 30% for testing and 20% for unseen validation.

Once again, a dynamic model was used for the modelling to capture time delayed correlations to improve predictions. The dynamic model form is shown in Equation (3.18).

$$\hat{y}(t) = f\big(\, u_1(t), \dots, u_{30}(t), u_1(t-1), \dots, u_{30}(t-1), \dots, u_1(t - d), \dots, u_{30}(t-d)\big) \tag{3.18}$$

where u_i is the i^{th} process output, t is the discrete time, and $\hat{y}$ is the predicted output.

The large number of process inputs, with the addition of time lagged inputs for the dynamic model, leads to poor model performance. So, utilising the dimensionality reduction capability of latent variable methods can improve model prediction performance.

As with the first case study, soft sensor models were created for values of *d*, from 0 to 5 in this case, for the proposed method and the three other methods that were used for comparison. The first 6 extracted SFs for the training and testing data set with *d* = 3 are presented in Figure 3.16. As with the debutaniser case study, this shows how the variation and levels of noise increase as the slow feature number increases.

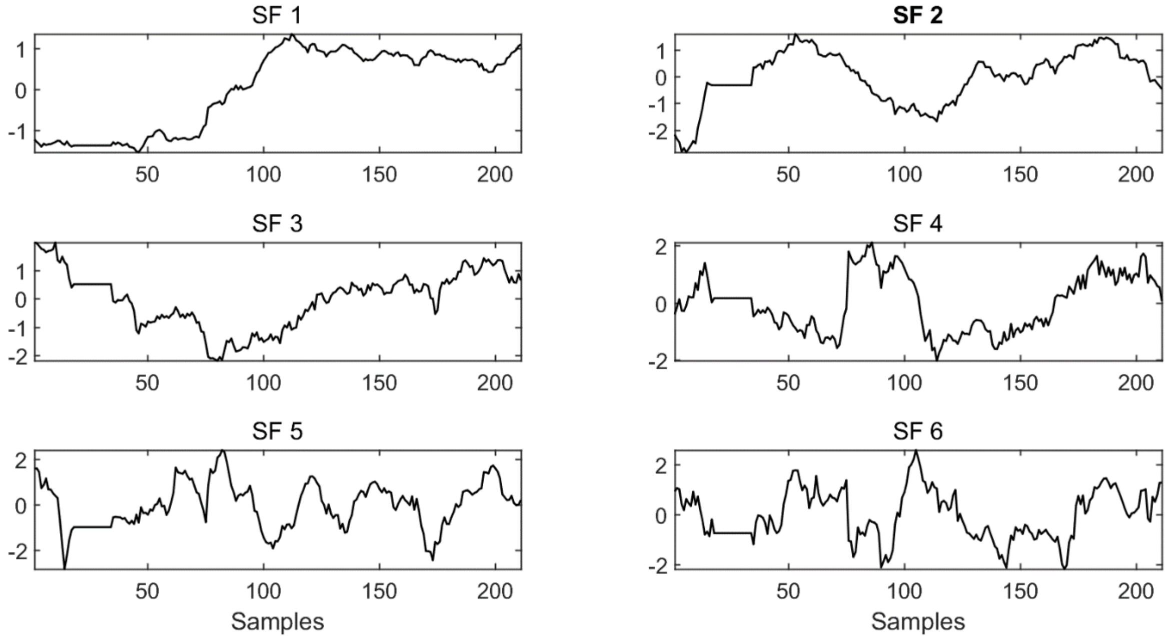

Figure 3.16 – The first 6 extracted slow features on the training and testing data set for the polymerisation case study

Table 3.5 – Soft sensor prediction results of melt index on the unseen validation data set

Methods	RMSE	R^2	Hyper-parameters
Single NN	16.7372	0.9489	d = 0, *#Inputs* = 30
PCA-NN	20.4307	0.9239	d = 4, k = 31
PLSR	18.2818	0.9390	*d = 3, k = 11*
SFR	16.2411	0.9519	d = 4, M = 17
SFA-NN	**15.5195**	**0.9561**	d = 3, M = 15

Table 3.5 shows the prediction results on the unseen validation data set for each of the five methods. The proposed method produces the best generalisation performance (highlighted in bold) and with the fewest number of latent variables, leading to a reduction in model complexity. SFR is the second-best method for this case study, likely due to the fact that the process does not have much of a nonlinear relationship between the quality variable and process variables, especially when compared with the first case study. However, the difference of 4.54% is still significant enough to justify the use of a neural network. Figures 3.17 to 3.20 illustrate the predictions on the unseen validation data against the actual melt index data, for four of the best methods, to further verify the results in Table 3.5.

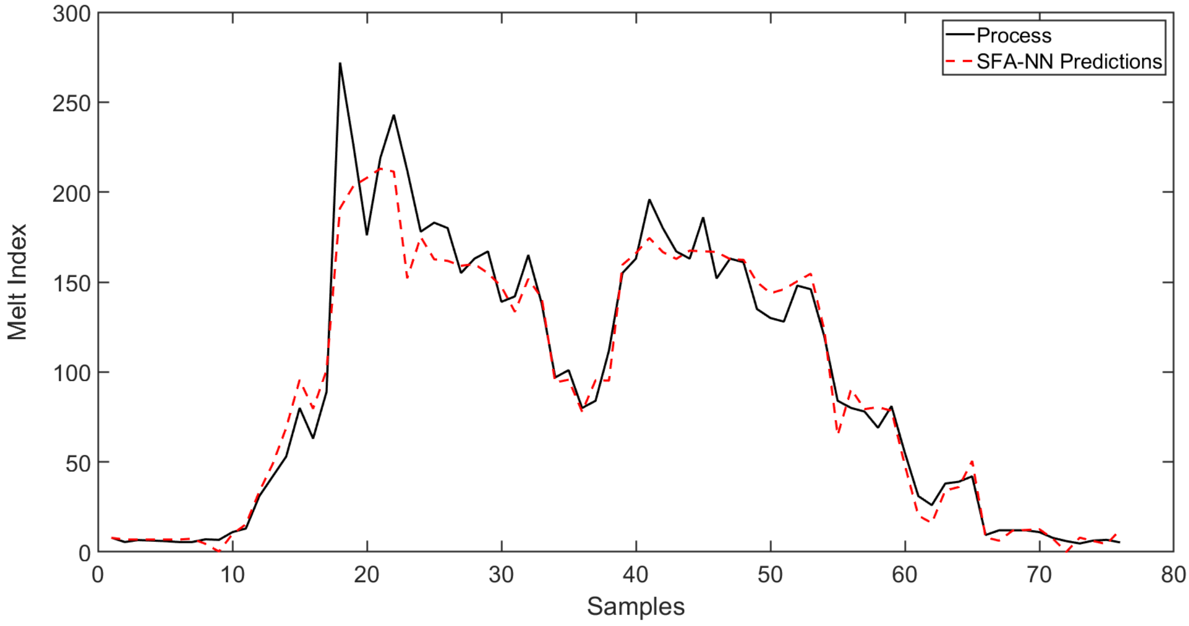

Figure 3.17 – Melt index soft sensor predictions on the unseen validation data set for the proposed SFA-NN (units omitted for confidentiality)

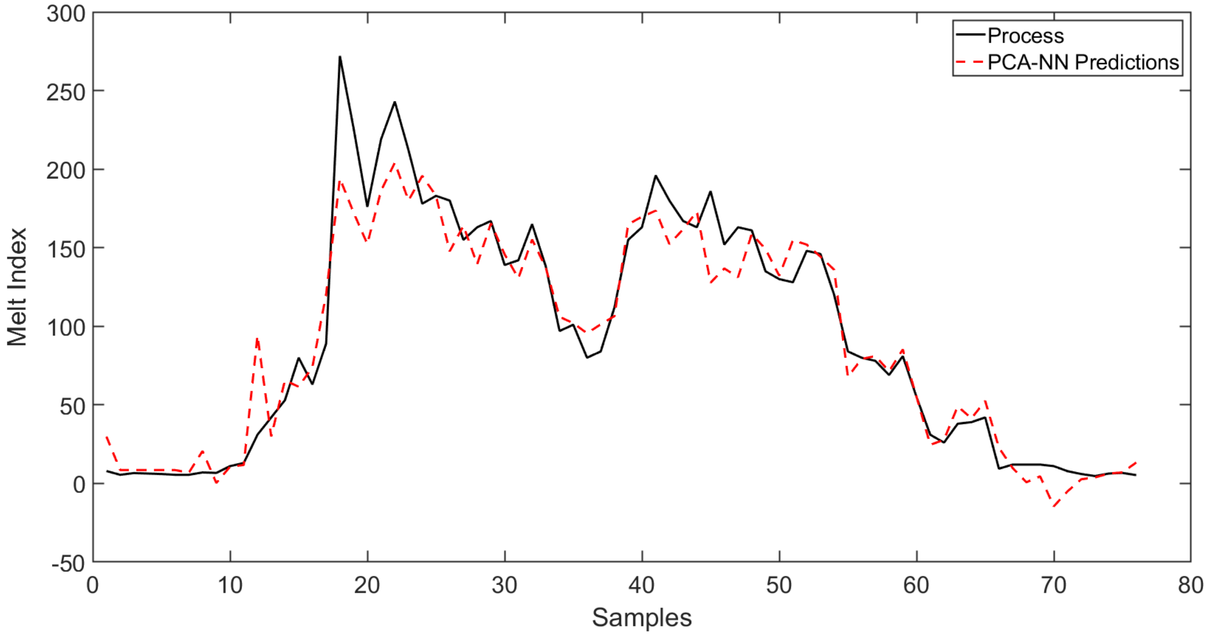

Figure 3.18 – Melt index soft sensor predictions on the unseen validation data set for PCA-NN (units omitted for confidentiality)

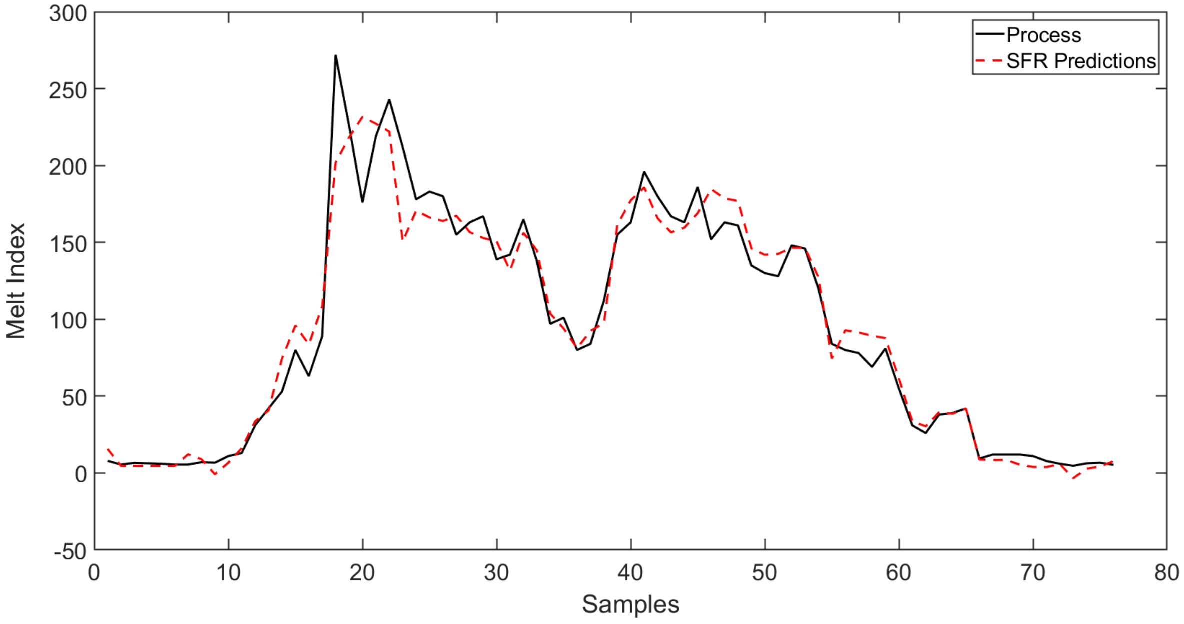

Figure 3.19 – Melt index soft sensor predictions on the unseen validation data set for SFR (units omitted for confidentiality)

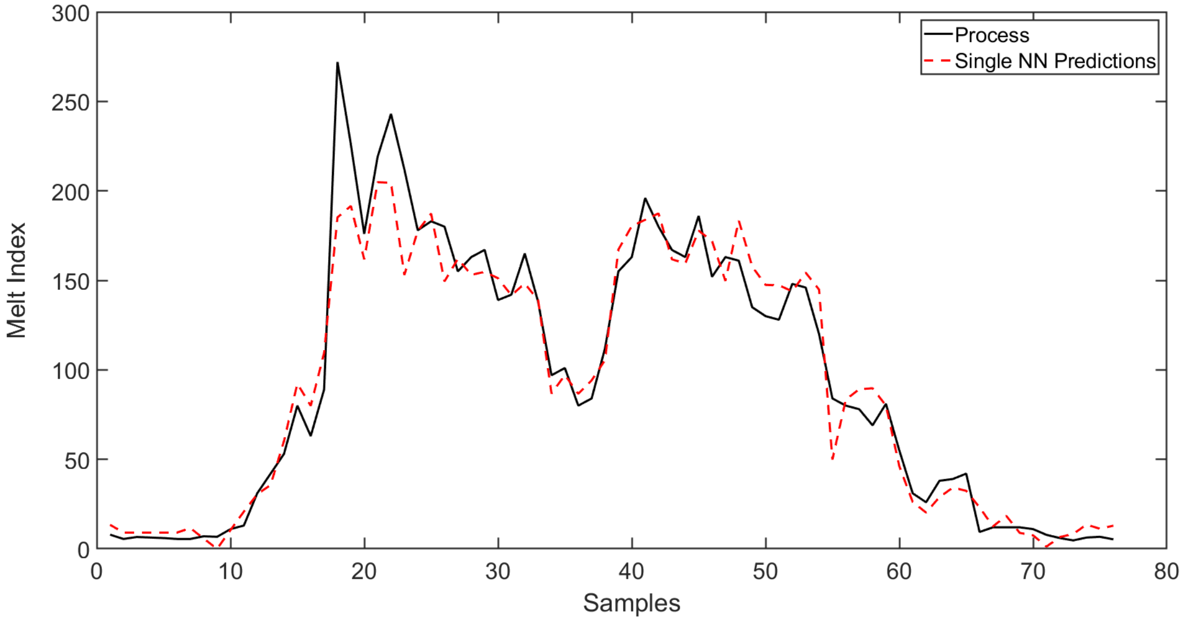

Figure 3.20 – Melt index soft sensor predictions on the unseen validation data set for single NN (units omitted for confidentiality)

As with the first case study, the scree plot for dominant SF selection is compared with the slowness of input reconstructions criterion, with these results presented in Table 3.6.

Table 3.6 – RMSE on the unseen validation data set with two dominant slow feature selection methods

SF Selection Method	RMSE for SFA-NN	RMSE for SFR	*M*
Scree Plot	**15.5195**	**16.2411**	17
Slowness of Input Reconstructions Criterion	18.8657	16.6463	59

In this case, the scree plot selection method gives better generalisation for both SFA-NN and SFR, with improvements in RMSE of 17.7% and 2.43% respectively, when compared with the selection method using slowness of input reconstructions. The reduction in RMSE is much larger for SFA-NN in this case study compared with the previous one. When comparing the number of dominant SFs of both selection methods, it seems that slowness criterion retains too many SFs, especially for processes such as this with many inputs and time lags included.

3.6 Conclusions

A novel method for enhancing soft sensor performance through integrating slow feature analysis and neural networks is proposed in this paper. Unlike other applications of SFA for soft sensor development, the use of neural networks with SFs as inputs can improve generalisation performance of soft sensors for complex processes where nonlinear relationships between input and output variables are present. The proposed method achieves this through utilising dynamic linear SFA to capture underlying dynamic trends in the data and then using a neural network as a more powerful machine learning technique to model the nonlinear relationships between the dominant SFs and product quality variables. This overcomes the pitfalls of other methods that make use of linear regression with SFA (SFR), where the prediction results are often insufficient for complex nonlinear processes.

The proposed SFA-NN method is applied to soft sensor development for two industrial processes: inferential estimation of butane concentration in a debutaniser column, and inferential estimation of polymer melt index in an industrial polymerisation process. In both case studies, SFA-NN is compared with PCA-NN, a single NN, traditional SFR and PLSR. The results on these case studies demonstrate the superior generalisation performance of the proposed method. Comparing with SFR, the proposed SFA-NN method produced a reduction in RMSE on the unseen validation data of 6.40% and 4.45% for case studies 1 and 2 respectively. This demonstrates the need for a nonlinear technique such as neural networks to deal with the nonlinearity in complex industrial processes. Comparing with single NN, SFA-

NN led to a reduction in RMSE on the unseen validation data of 11.1% and 7.28% for the first and second case studies respectively. This result shows the improvement in generalisation performance produced when dynamic SFA is used prior to neural network training. This improvement occurs due to SFA capturing slow varying trends in the data that express integral process dynamics and only retaining a reduced number of SFs that contain relevant information without much noise.

Linear SFA has limitations in situations where the underlying driving forces are nonlinear and so extensions of the proposed method can be explored, such as utilising more complex versions of SFA instead of linear SFA, to further improve generalisation capability for chemical process applications. This leads into the next chapter which expands linear SFA to a nonlinear form: kernel SFA.

Chapter 4. Using Dynamic Kernel Slow Feature with a Neural Network to Further Improve Soft Sensor Performance

4.1 Introduction

A limitation of SFA applications is in the use of linear SFA with its inability to truly extract driving forces behind data in nonlinear situations. A solution to this problem is to extend SFA to a nonlinear version using kernels. One of the main benefits of incorporating kernels is that a nonlinear mapping can be carried out without the mapping being explicitly calculated, saving computational effort (Schölkopf *et al.*, 1998). Using kernels to improve multivariate statistical methods is nothing new and kernel PCA (Schölkopf *et al.*, 1998) has been widely and effectively applied to soft-sensing (Li *et al.*, 2012; Yuan *et al.*, 2014; Chen *et al.*, 2018; Tang *et al.*, 2018). Incorporating a kernel into SFA has been applied for process monitoring (Zhang *et al.*, 2017) and fault detection (Zhang *et al.*, 2015b), but it is untested for soft-sensing/predictive modelling.

In addition to pre-processing techniques, including external dynamics into the model can account for time-lagged correlations between inputs and output, leading to improved model performance.

In this work, a novel process modelling method for soft sensor development is proposed by incorporating dynamic kernel SFA (KSFA) with a neural network, with the aim of improving soft sensor robustness and accuracy. First, dynamic KSFA is applied to the process inputs to extract nonlinear SFs. Then, a reduced number of SFs are used as the inputs to a neural network. The proposed method is applied to a numerical example and a benchmark fed-batch penicillin process to demonstrate its effectiveness when compared with other similar techniques, including KSFA with linear regression.

The chapter is organised into the following sections: the dynamic KSFA algorithm is described in Section 4.2, the proposed soft sensor design method, named DKSFANN, is presented in Section 4.3. Section 4.4 contains the two case studies: a nonlinear numerical example and a benchmark fed-batch penicillin process simulation. The conclusions of the work are presented in Section 4.6.

4.2 Dynamic Kernel Slow Feature Analysis

4.2.1 Kernel Slow Feature Analysis

As described in Chapter 3, the solution to linear SFA (LSFA) defines the function $\boldsymbol{g}(\mathbf{x})$ as simply a linear function with weights $\boldsymbol{W}$:

$$s(t) = \boldsymbol{W}\boldsymbol{x}(t)\,. \tag{4.1}$$

LSFA fails to extract the underlying trends of a data set when the relationship between the data and the underlying trends is nonlinear. An extension of the linear algorithm is to have $\boldsymbol{g}(\mathbf{x})$ be a nonlinear function $\boldsymbol{f}$, so that nonlinear SFs can be extracted:

$$\mathbf{s}(t) = \boldsymbol{f}\big(\mathbf{x}(t)\big)\,. \tag{4.2}$$

One way to determine the nonlinear mapping $\boldsymbol{f}$ is to employ a nonlinear expansion, such as a quadratic expansion. However, such a polynomial expansion will only be effective in certain situations. Therefore, a more effective approach is to employ the kernel trick by introducing kernel principal component analysis (KPCA) into the SFA algorithm to map the inputs into a high-dimensional feature space (Zhang *et al.*, 2015b), thus creating a KSFA algorithm. The key benefit of the kernel trick is that it is not necessary to determine the nonlinear mapping $\boldsymbol{f}$ explicitly, the only required calculations come from the dot products in the use of the kernel function on the input space (Schölkopf *et al.*, 1998).

KPCA is applied to the inputs $\boldsymbol{x}(t)$, using a kernel function $\boldsymbol{k}$**,** to extract a higher dimension of kernel principal components $\boldsymbol{K}(t)'$, which are then sphered to meet the SFA constraints (3.2)-(3.4), giving $\boldsymbol{K}(t)$. This then transforms Equation (4.2) to a linear combination of a weights vector $\boldsymbol{W}$ and the normalised kernel principal components:

$$\boldsymbol{s}(t) = \boldsymbol{W}\boldsymbol{K}(t)\,. \tag{4.3}$$

The problem has now been reformulated to a linear one, just like Equation (4.1), except that now the nonlinearity is captured through the kernel. The optimisation problem (3.1) is therefore to optimise the weights such that

$$\langle \dot{s}_j^2 \rangle = \boldsymbol{W^T} \langle \dot{\boldsymbol{K}} \dot{\boldsymbol{K}}^T \rangle \boldsymbol{W} \tag{4.4}$$

is minimal, where $\langle \dot{\boldsymbol{K}} \dot{\boldsymbol{K}}^{\boldsymbol{T}} \rangle$ is the covariance matrix of $\dot{\boldsymbol{K}}$, calculated as follows:

$$\langle \dot{\boldsymbol{K}} \dot{\boldsymbol{K}}^{\boldsymbol{T}} \rangle = \frac{\sum_{i=1}^{P} (\dot{\boldsymbol{K}}_i - \mu_{\dot{K}})(\dot{\boldsymbol{K}}_i - \mu_{\dot{K}})}{N-1}, \tag{4.5}$$

where *N* is the number of data points, $\mu_{\dot{\mathbf{K}}}$ is the mean of $\dot{\boldsymbol{K}}$, and $\dot{\boldsymbol{K}}$ is the first order derivative of ***K***, which is calculated using a first order difference approximation due to the discrete sampling interval *h* used in the sampled process data, that is:

$$\dot{\mathbf{K}}(t) \approx \frac{\mathbf{K}(t) - \mathbf{K}(t-h)}{h}. \tag{4.6}$$

This temporal derivative approximate follows the use of it in other works on SFA (Shang *et al.*, 2015c; Huang *et al.*, 2020; Jia *et al.*, 2020; Wang *et al.*, 2021b; Xu and Ding, 2021).

This optimisation problem can be reduced to the following eigenvalue problem, which is easily solved by performing singular value decomposition (SVD) on $\langle \dot{\boldsymbol{K}} \dot{\boldsymbol{K}}^T \rangle$:

$$\langle \dot{\boldsymbol{K}} \dot{\boldsymbol{K}}^T \rangle = \boldsymbol{W^T \Omega W} \ , \tag{4.7}$$

where $\boldsymbol{\Omega}$ is a diagonal matrix of the eigenvalues, which are the optimal values of the objective function, corresponding to the eigenvectors which equate to the weight vectors ***W***.

Therefore, the slow feature outputs can be calculated using the kernel principal components and the derived weights vector via Equation (4.3). There is a question of the choice of kernel function ***k***, as certain functions lead to better extraction of the driving forces than others. Schölkopf *et al.* (1998) discussed how Mercer's theorem states that ***k*** must be a continuous kernel of a positive integral operator for there to be a situation where the kernel function acts as a dot product. To satisfy Mercer's theorem, three main kernel functions are typically used:

$$\text{Polynomial: } \boldsymbol{k}(a, \mathrm{b}) = \langle a, \mathrm{b} \rangle^{\delta} \ , \tag{4.8}$$

$$\text{Sigmoidal: } \boldsymbol{k}(a, \mathrm{b}) = \tanh(\beta_0 \langle a, \mathrm{b} \rangle + \beta_1) \ , \tag{4.9}$$

$$\text{Gaussian: } \boldsymbol{k}(a, \mathrm{b}) = \exp\left(-\frac{\|a-b\|^2}{2\sigma^2}\right) \ , \tag{4.10}$$

where δ is the degree of the polynomial, β_0 and $\beta_1$are the slope and intercept for the sigmoidal function respectively, and σ is the standard deviation that determines the width of the Gaussian kernel. These parameters are user specified prior to using the function. The polynomial and Gaussian kernels always satisfy Mercer's theorem, while the sigmoidal kernel only does so for certain values of β_0 and β_1 (Lee *et al.*, 2004b). There appears to be no certain method for selecting the appropriate kernel function and it can be difficult to choose, however, prior knowledge of the process can help the decision. Sprekeler *et al.* (2014) used high order polynomials for the kernel, but they do state that radial basis functions (Gaussian) can be more robust, depending on the input data. The downfall of the Gaussian kernel is that it is often more computationally expensive than polynomials, but this also depends on the input data. The Gaussian kernel is the most widely used in the process systems engineering field (Yuan *et al.*, 2014; Zhang *et al.*, 2015b; Zhang *et al.*, 2018a) and so it is also the choice for the proposed method in this paper. However, the optimal choice of kernel function remains an open problem.

4.2.2 Dynamic Kernel Slow Feature Analysis

Often it is assumed that the current sample of a process variable is independent from previous samples, however, industrial processes commonly have dynamic characteristics where the correlations between process inputs and outputs have time lags. Therefore, these dynamic correlations should be considered by including time lagged samples into the input data matrix, up to a chosen time lag, *d*. So, the augmented input vector at time *t* is:

$$\mathbf{x}_d(t) = [\mathbf{x}(t) \ \ \mathbf{x}(t-1) \ \cdots \mathbf{x}(t-d)] \, , \tag{4.11}$$

where $\mathbf{x}(t)$ is a vector of inputs at time *t*.

Suppose $\mathbf{X} = [\mathbf{x}(1)^{\mathrm{T}} \ \mathbf{x}(2)^{\mathrm{T}} \ \ldots \ \mathbf{x}(N)^{\mathrm{T}}]^{\mathrm{T}}$ is a matrix of process variables with *N* samples, then the augmented matrix $\mathbf{X}_d$ is obtained by adding the past *d* values of the process variables:

$$\mathbf{X}_d = \begin{bmatrix} \mathbf{x}_d(d+1) \\ \mathbf{x}_d(d+2) \\ \vdots \\ \mathbf{x}_d(N) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(d+1) & \mathbf{x}(d) & \cdots & \mathbf{x}(1) \\ \mathbf{x}(d+2) & \mathbf{x}(d+1) & \cdots & \mathbf{x}(2) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}(N) & \mathbf{x}(N-1) & \cdots & \mathbf{x}(N-d) \end{bmatrix} \tag{4.12}$$

KSFA is extended to dynamic KSFA (DKSFA) by using the matrix $\mathbf{X}_d$ to perform KSFA (Zhang *et al.*, 2015b). The selection of *d* can be carried out through correlation analysis, cross validation, or utilising process knowledge.

4.3 Soft-sensing Using Kernel Slow Feature Analysis and Neural Network

Although KSFA has been incorporated into several recent works on process monitoring (Zhang *et al.*, 2017) and process fault detection (Zhang *et al.*, 2015b; Zhang *et al.*, 2018a; Zhang *et al.*, 2021), the effect of the nonlinear SFs from KSFA as inputs to a nonlinear regression model has not yet been investigated. Soft sensor design differs from process monitoring and other process applications by using regression/machine learning-type methods to create predictions of difficult to measure key product quality variables through other easy to measure process variables. KSFA improves soft sensor performance in complex cases where other techniques are poor by extracting nonlinear driving forces in the data, which provides important process information to the regression model that is not present from the original input data. By extracting slow varying features, the effect of noise is also reduced. By extracting nonlinear driving forces in the data, with lower levels of noise than the process inputs, and using them as the model inputs as opposed to just the original noisy process variables, better soft sensor predictions are expected. One potential drawback of using kernel slow feature analysis is that the driving forces are not extracted effectively, and so misleading information is input to the model. Furthermore, selecting the correct hyper-parameters to successfully extract the driving forces is difficult, especially with the multiple hyper-parameters involved in KSFA.

Soft sensors and other types of data-driven models can be created by using DKSFA as a precursor to a regression technique. Typical choices of a regression technique are linear regression, for situations where the relationship is linear, and neural networks (Bishop, 1995), for when the relationship is nonlinear. The approach in this paper uses a neural network so that it can produce more accurate models in complex nonlinear industrial processes. The soft sensor design method based on DKSFA with neural network (DKSFANN) is described in the following sequence:

Model Training:

1. Normalise a training data matrix of n inputs to zero mean and unit variance and then obtain the dynamic matrix in Equation (4.12) for a given d.
2. Using a Gaussian kernel function with a kernel width σ, apply KPCA to this normalised input matrix to extract L kernel principal components in the matrix $\boldsymbol{K}(t)'$, where $L \geq n$. The application of KPCA, unlike linear PCA, allows for the extraction of a higher number of principal components (PCs) than the input dimensionality, which is necessary to effectively extract nonlinear driving forces from SFA. The parameters σ and L are both determined by cross validation on testing data.
3. Sphere $\boldsymbol{K}(t)'$ to give $\boldsymbol{K}(t)$, and then calculate the covariance matrix $\langle \dot{\boldsymbol{K}}\dot{\boldsymbol{K}}^T \rangle$.
4. Solve the eigenvalue problem in Equation (4.7) by performing SVD on $\langle \dot{\boldsymbol{K}}\dot{\boldsymbol{K}}^T \rangle$ to calculate the weights vector $\boldsymbol{W}$**.**
5. Calculate the SFs $\mathbf{s}(t)$ from Equation (4.3).
6. Determine the number of dominant SFs M by cross validation. The use of other methods to determine the dominant SFs is discussed in Section 3.3.
7. Use the M dominant SFs as the inputs to a single hidden layer feed-forward neural network (SLFNN). A SLFNN is used because Cybenko (1989) proved that a SLFNN using a sigmoidal activation function can approximate any nonlinear function, with a sufficient number of hidden neurons.
8. The neural network is trained using the Levenberg-Marquardt algorithm with regularisation and a sigmoidal activation function is used in the hidden layer. The optimal number of hidden neurons N_H is found by cross validation.
9. The output of the neural network is the soft sensor prediction ($\hat{\boldsymbol{y}}$).

Model Testing:

i. Collect new input data and normalise it using the same mean and variance as used for the training data in model training step 1. Then augment the matrix with time lag d.

ii. Compute the kernel principal components of dimension L on the new data using the eigenvectors determined from KPCA in training step 2, with the same Gaussian kernel of width σ**.**

iii. Sphere the kernel principal components using the same sphering matrix as in training step 3.

iv. Calculate the new SFs from Equation (4.3), using the same weights *W*, as computed in model training step 4.

v. Apply the new *M* SFs to the previously trained neural network with N_H hidden neurons to produce a soft sensor prediction on the new data.

The above steps describe the creation of a soft sensor using DKSFANN. A basic summary of the key steps is illustrated in Figure 4.1.

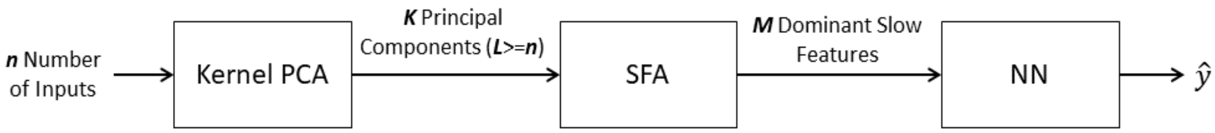
n Number of Inputs
Kernel PCA
K Principal Components (*L*>=*n*)
SFA
M Dominant Slow Features
NN
$\hat{y}$

Figure 4.1 – Simple diagram of the DKSFANN soft sensor modelling data flow

4.4 Case Study 1 – Numerical Example

In the following two sections, two case studies are used to demonstrate the performance of the proposed method in comparison with several other similar methods. One case study is a numerical example based on sine functions so that it can best demonstrate the theoretical advantages of KSFA, as well as evaluate the regression performance. The other case study is a benchmark simulated industrial fed-batch penicillin process which is used to evaluate the proposed method for industrial applications.

For this first case study, a numerical example with 3 inputs signals and 1 output signal is considered. The inputs were created by making nonlinear functions of 2 driving forces: *sin*(*πt/*250) and *sin*(*πt/*50). The aim is for KSFA to extract the driving forces as the SFs for a nonlinear situation where LSFA cannot. The inputs signals are described by the following:

$$x_1(t) = 1.5e^{\sin(\frac{\pi}{250}t)} - e^{\sin\left(\frac{\pi}{50}t\right)} + v_1 \, , \tag{4.13}$$

$$x_2(t) = e^{-\sin(\frac{\pi}{250}t)} + 1.5e^{-\sin(\frac{\pi}{50}t)} + v_2 \, , \tag{4.14}$$

$$x_3(t) = 3e^{-\sin(\frac{\pi}{50}t)} + e^{-\sin(\frac{\pi}{250}t)} + v_3 \, , \tag{4.15}$$

where x_1, x_2 and x_3 are the input functions, t is the discrete time, and v_1, v_2 and v_3 are random Gaussian noise.

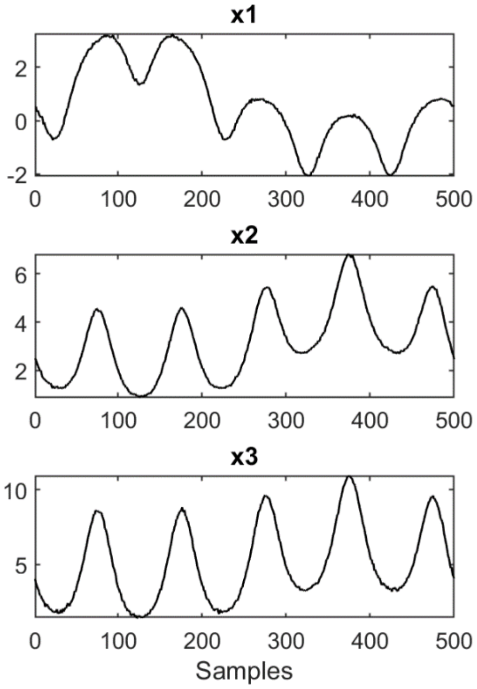

Figure 4.2 – Input signals for the numerical example

The output signal is a nonlinear dynamic function of the driving forces, with added random noise v_4:

$$y1(t) = 3\sin\left(\frac{\pi}{50}t\right) + 2\sin\,(\frac{\pi}{250}(t-2))^2\sin\,(\frac{\pi}{50}(t-2))^2 + 1.5\sin\,(\frac{\pi}{250}(t-1))^2 + v_4\ . \tag{4.16}$$

500 samples were created with the discrete time t in the range [0, 500]. The random noises added are as follows: v_1= N(0,0.035), v_2=N(0,0.035), v_3=N(0,0.075), v_4=N(0,0.01). Figure 4.2 illustrates the input signals from Equations (4.13) to (4.15), whilst Figure 4.3 shows the output from Equation (4.16).

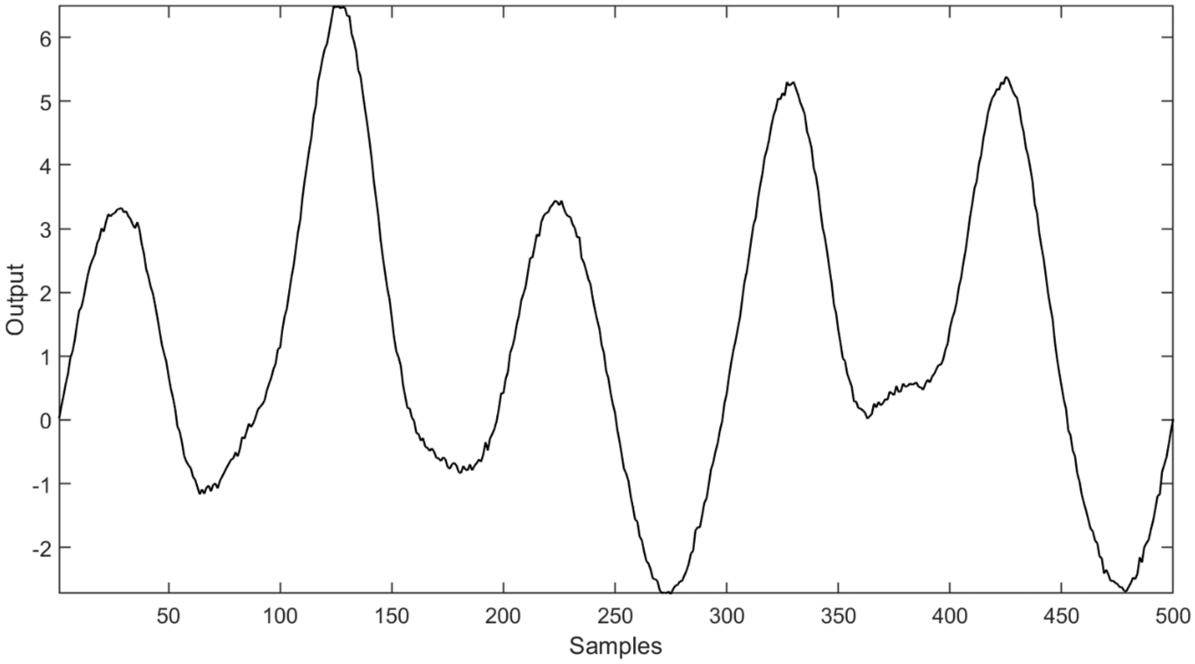

Figure 4.3 – Output signal for the numerical example

4.4.1 Extraction of Driving Forces

The main reason for using KSFA is for the extraction of the driving forces behind a set of input signals in nonlinear cases where LSFA fails to do so effectively. To evaluate whether the extracted SFs are equivalent to the driving forces, Pearson correlation coefficients (R) are calculated, where a coefficient of 1 indicates a perfect match between the slow feature and driving force. For both KSFA and LSFA, the correlation coefficients were calculated for this numerical example with noise to assess whether the driving forces behind the input signals, $sin(\pi t/250)$ and $sin(\pi t/50)$, were successfully extracted in the SFs. The correlation coefficients for KSFA and LSFA are presented in Table 4.1. These results show that the correlation coefficients are higher for KSFA, for both $sin(\pi t/250)$ and $sin(\pi t/50)$, when compared with LSFA.

The significance of the improvements in correlation coefficient is demonstrated through Figure 4.4, which provides an illustration of the comparison between the extracted driving forces for KSFA and LSFA. This confirms the results exhibited through the correlation coefficients and shows much better extracted driving forces from KSFA than LSFA. The lack of exact driving force extraction (correlation coefficient equal to 1) when using KSFA is down to the noise in the input signals. If noise was not present, perfect extraction of the driving forces is achieved, as illustrated in Figure 4.5.

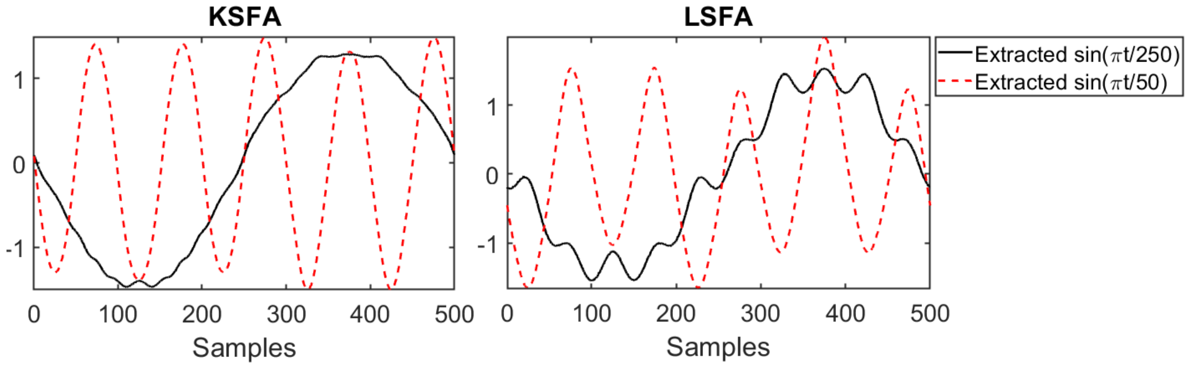

Figure 4.4 – Comparison of the driving forces extracted in the first two slow features for KSFA and LSFA

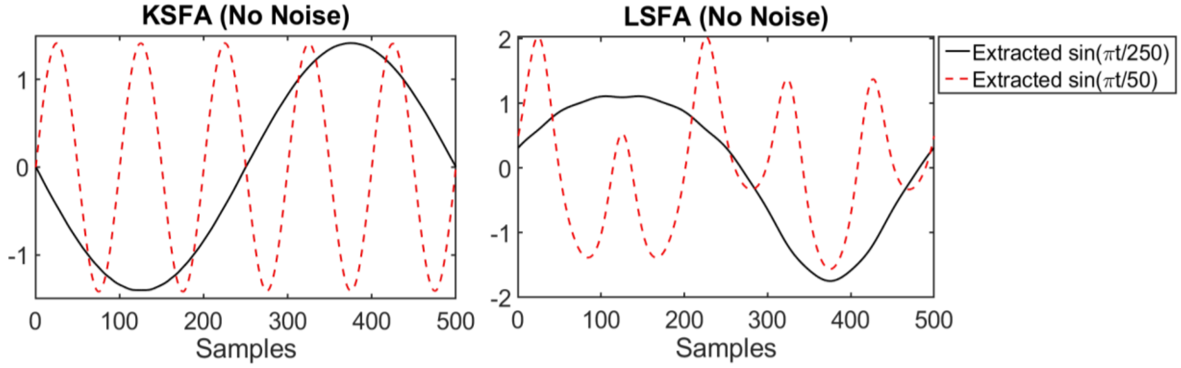

Figure 4.5 – Comparison of the driving forces extracted in the first two slow features for KSFA and LSFA, if no noise is present in the input signals

Table 4.1 – Correlation coefficients (R) between the extracted and true driving forces

Driving force	KSFA	LSFA
$sin(\pi t/250)$	0.9965	0.9721
$sin(\pi t/50)$	0.9961	0.9706

4.4.2 Regression Results

With KSFA demonstrating effective extraction of the underlying driving forces in nonlinear cases where LSFA cannot, the attention now turns to how this can be used to improve regression performance. By extracting nonlinear driving forces/trends behind the data with reduced noise, KSFA can input this key information to the regression technique, leading to improved prediction performance. Without the SF extraction, the inputs to regression do not contain as much useful information that is also potentially distorted by noise. To demonstrate this regression performance, dynamic KSFA was paired with both a SLFNN (DKSFANN), as described in Section 4.3, and linear regression (DKSFR). In these models, the extracted driving forces from KSFA are used as the inputs to the regression technique. Note that lagged input variables are used in the SF extraction step, so the models can be considered as dynamic models. The reason for using both linear regression and a neural network after KSFA is to show that a nonlinear based regression method, such as a neural network, is often necessary in nonlinear cases. For example, in this numerical case study, the output signal is a nonlinear dynamic function of the two driving forces. Therefore, even when KSFA extracts the driving forces in the SFs, linear regression will not be able to capture the relationship between these SFs and the output signal, whereas the neural network will typically be able to. Other similar techniques were included for comparison: dynamic partial least squares regression (DPLSR), dynamic principal component regression (DPCR), dynamic linear slow feature regression (DLSFR), dynamic kernel PCR (DKPCR), dynamic kernel SFR (DKSFR), dynamic single hidden layer feedforward neural network (DNN), dynamic principal component analysis with NN (DPCANN), dynamic linear slow feature analysis with NN (DLSFANN), and dynamic kernel principal component analysis with a neural network (DKPCANN). All these methods involved applying the pre-treatment technique (SFA, PCA etc.), when applicable, to the normalised input signals and then using the outputs of the pre-treatment technique as the inputs to the regression method, either linear regression or a neural network. For example, for DPCANN, the input matrix is augmented to include external dynamics in the form of previous time samples for the inputs. Then PCA is applied to this augmented matrix and the first k principal components are used as the inputs to a neural network. Selection of k PCs is done by cross validation to match the method used for the selection of M for KSFA-based methods. The scree plot was not employed in this chapter because there were multiple hyperparameters for KSFA that are dependent and so the scree plot changes as the other parameters, such as σ, changes. Additionally, all the above techniques utilise external dynamics in their inputs, in the same method as for dynamic KSFA, as described in Section 4.2.2.

The neural networks are trained using the Levenberg-Marquart algorithm with a regularisation parameter of 0.1 to help in preventing overfitting. The PCA based techniques provide a good comparison with the SFA based techniques because the KPCA/PCA layer performs feature extraction, as does the KSFA/SFA layer.

To test the effect of the different dynamic approaches, a second order NARX neural network was developed. This NARX network uses lagged inputs and output and the network inputs and is trained to create one-step ahead predictions. One-step ahead predictions are given in the following form:

$$\hat{y}(t) = f\big(y(t-1), y(t-2)\,, x_1(t-1), x_2(t-1), x_3(t-1), x_1(t-2), x_2(t-2), x_3(t-2)\big)\,, \tag{4.17}$$

where $\hat{y}(t)$ is the prediction of the output y at time t, and x_1 to x_3 are the three inputs.

For model validation, the NARX network is assessed in simulation mode to create multi-step ahead predictions to form a long-range predictive model. One step ahead predictions are not a fair comparison to the other techniques because lagged outputs are not included in those models, only lagged inputs are used. Therefore, when comparing any techniques with the NARX neural network, only the multi-step ahead predictions on the validation data provide a fair comparison because they are long-range predictions (i.e., no output values included as model inputs). The multi-step ahead predictions, or long-range predictions, of the NARX neural network are calculated as follows:

$$\hat{y}(t) = f\big(\hat{y}(t-1), \hat{y}(t-2)\,, x_1(t-1), x_2(t-1), x_3(t-1),\ x_1(t-2), x_2(t-2), x_3(t-2)\big)\,. \tag{4.18}$$

The 500 samples created for the input and output signals were split into subsets. The first 60% for training, the next 20% for testing and the final 20% for unseen validation data. Training data is used to create the models (e.g. calculate the eigenvectors in PCA and SFA or train the weights of neural network). Testing data is used to determine the optimal hyper-parameters, such as the number of hidden neurons (N_H), via cross validation on the root mean squared error (RMSE) of the testing data. The unseen validation data take no part in the creation of the model and are only used as a data set to properly test the model's generalisation performance. To assess how including the dynamics affect the model performance for each technique, models were created for time lags, d, of 0 to 2, with the optimal d selected by cross validation on the testing data as well.

The results of the model performance for each technique are presented in Table 4.2. The R^2 and RMSE values for training, testing and unseen validation data sets are presented, although only the RMSE values on the unseen validation data are used to assess which technique produces the best model. The R^2 is included as it provides a performance metric that can be more easily interpreted as to how good the model performance actually is. The results show that DKSFANN produces the best model for this case study, with a validation RMSE that is 31.7% lower than the second-best technique, DKPCANN. The significance of this difference can be shown by visually comparing the predictions on the validation data, as illustrated in Figure 4.7 for DKSFANN and Figure 4.8 for DKPCANN. Often when using only the validation RMSE as the model performance metric, poor performance on the testing or training data would not be factored into selecting the best model and so it is important to check the RMSE on training and testing to check that the performance on validation data was not in fact by a model with poor generalisation. Looking at the performance on training and testing data for DKSFANN shows that this model gives good performance across all data sets. In comparison with DKPCANN, the training and testing RMSE values for DKSFANN are 13.6% and 42.1% lower respectively. This further adds to the conclusion that the DKSFANN model is significantly better than the compared models. Theoretically, DKPCANN and DKSFANN are very similar; they both incorporate external dynamics using the same method, both use a kernel-based feature extraction algorithm based on eigenvalues, and both have single hidden layer neural networks as the regression technique. However, KSFA can almost perfectly extract the driving forces behind the input signals (Figure 4.4), in the presence of noise, leading to more important information being supplied to the neural network since the output is derived from these driving forces. This leads to improved model generalisation. In comparison, the PCs extracted from KPCA contain some kind of sine-based function (Figure 4.6), however, it is not a good representation of the driving forces as extracted by KSFA, accounting for the lower generalisation performance.

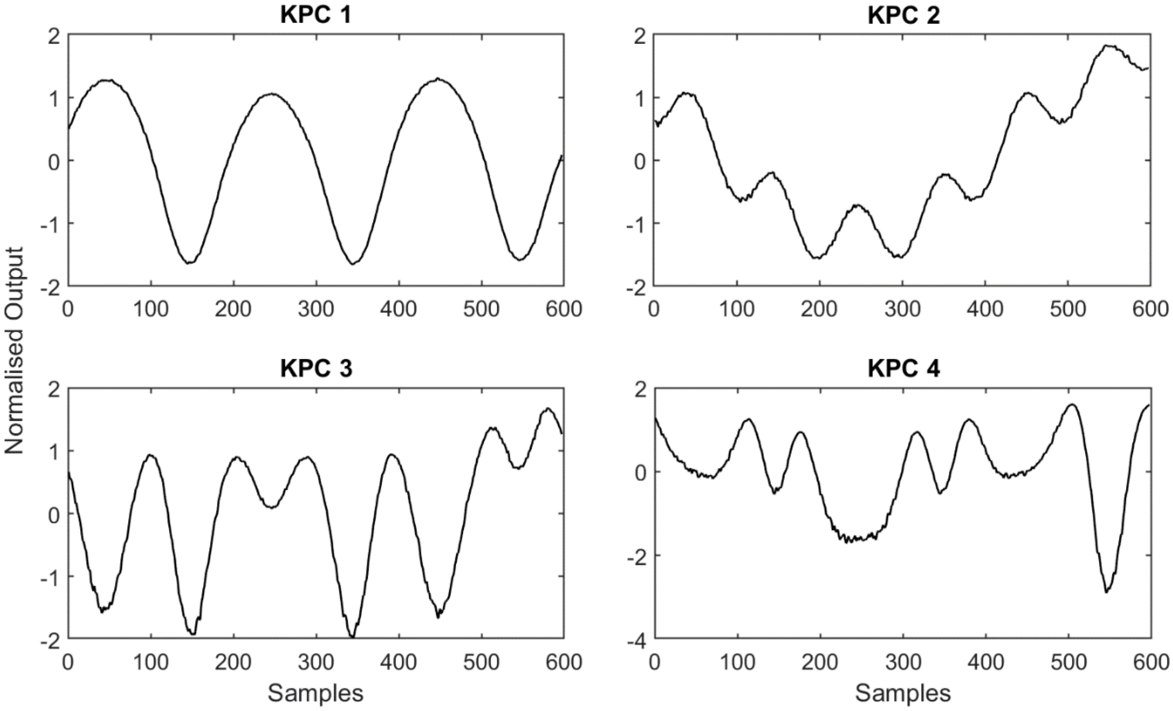

Figure 4.6 – First 4 extracted features from KPCA for the DKPCANN model

Table 4.2 – Regression results for the numerical case study

	Training		Testing		Validation		
Method	R^2	RMSE	R^2	RMSE	R^2	RMSE	Hyper parameters
DLR	0.8452	0.3928	0.0015	2.2973	0.0202	1.5001	d = 2
DPLSR	0.7611	0.4880	0.0317	1.1566	0.6418	0.7508	d = 2, k = 1
DPCR	0.7073	0.5401	0.1257	0.8890	0.7266	0.6559	d = 2, k = 1
DLSFR	0.0580	0.9983	0.1056	0.8985	0.0223	1.2525	d = 1, M = 1
DKPCR	0.9925	0.0867	0.9276	0.2135	0.9804	0.1527	d = 2, k =15, σ = 4.5, L = 15
DKSFR	0.9917	0.0910	0.9424	0.1906	0.9907	0.1158	d = 2, M = 10, σ = 4.5, L = 15
DNN	0.9616	0.1957	0.4031	0.6133	0.9481	0.2859	d = 2, N_H = 35
NARX NN*	*0.9626*	*0.1227*	*0.9707*	*0.1382*	0.4890	0.9063	d = 2, N_H = 5
DPCANN	0.7713	0.4774	0.7387	0.4058	0.8456	0.4929	d =1, k = 3, N_H = 26
DLSFANN	0.9031	0.3107	0.7046	0.4315	0.8375	0.5057	d = 1, M =2, N_H = 32
DKCPANN	0.9969	0.0557	0.9318	0.2073	0.9919	0.1132	d = 2, k = 8, σ =6.5, N_H = 7, L = 12
DKSFANN	**0.9977**	**0.0481**	**0.9771**	**0.1200**	**0.9962**	**0.0773**	**d = 2, M = 5, σ =5, N_H = 35, L = 15**

*NARX NN uses 1 step ahead predictions for training and testing (incorporates output signal, which the other methods do not), therefore the metrics for training and testing are not fair to compare to other methods. Only comparing the multistep ahead predictions on validation data using the NARX NN is fair.

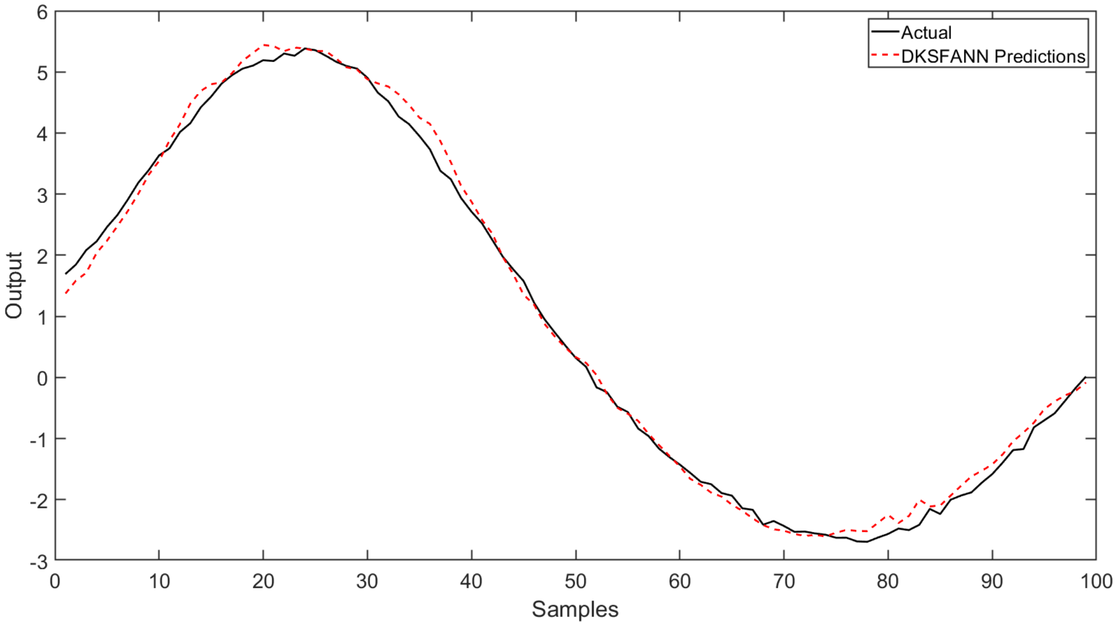

Figure 4.7 – Predictions on unseen validation data using DKSFANN for the numerical case study

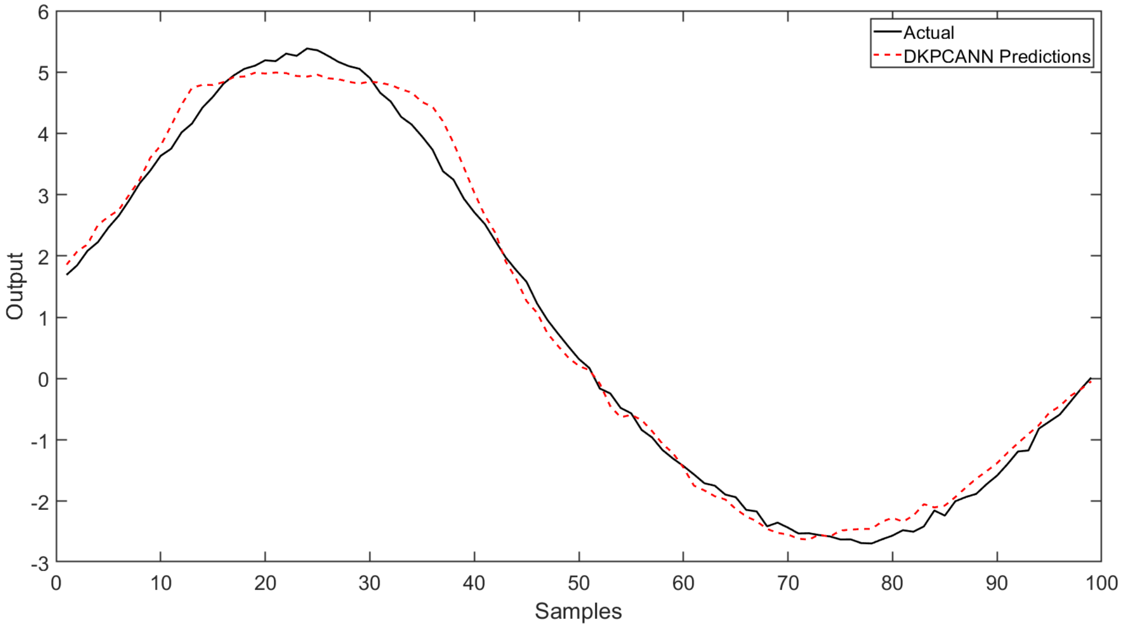

Figure 4.8 – Predictions on unseen validation data using DKPCANN for the numerical case study

Another result worth noting is that DKSFR is the best of the linear regression-based models, particularly when compared with the non-kernel techniques. Even comparing between the two kernel techniques, the validation RMSE of DKSFR is 24.2% lower than that of DKPCR. As with the corresponding comparison of the NN-based techniques, the improvement of DKSFR over DKPCR (and the other LR-based techniques) can be accounted for by the extraction of the underlying driving forces by KSFA, which provide more meaningful, less noisy information to regression than the original input signals.

One more important comparison is between the KSFA-based and the SFA-based models, as this comparison shows the importance of correctly extracting underlying driving forces behind data and the impact this can have on model performance. DKSFR outperforms DLSFR with a validation RMSE that is 90.8% lower, while DKSFANN has a validation RMSE 84.7% lower than DLSFANN. Therefore, on average, KSFA-based methods improve model accuracy by 87.8% compared with LSFA-based methods, demonstrating the significant benefits of properly extracting the driving forces on modelling performance and the massive impact it can have on modelling performance.

Very low R^2 values are seen for DLSFR (and a few other instances). Additionally, the RMSEs in these cases are also a lot larger than other techniques, showing that it is just poor fitting in this case. This is likely due to the low number of retained slow features (M=1), which lead to optimal testing performance but was in fact poor on training.

The effect of incorporating external dynamics, and the number of time lagged samples *d,* on the model generalisation performance is clear in this example. Out of the 11 methods in Table 4.2, the optimal value of *d* was 2 for all but 3 techniques, where it was 1. This shows that, for this example, incorporating the dynamics had a significant impact across the different techniques. This is to be expected as the output signal contains time lagged relationships between the inputs, or the driving forces, and the output, meaning that incorporating time lagged inputs can uncover these time lagged correlations. Comparing different forms of dynamic models, the NARX NN with long-range predictions performed poorly compared with DNN. This is likely due to the propagation of errors associated with using the NARX model trained for one step ahead predictions (see Equations (4.17) and (4.18)). Model complexity can be another factor in determining the optimal model, especially when the number of inputs is large and so dimensionality reduction can be an important tool for the feature extraction techniques. DKSFANN retained only 5 SFs to be the inputs to the neural network, compared with 8 PCs for the DKPCANN model. This demonstrates that DKSFANN reduced dimensionality more then DKPCANN while still producing better model performance.

These results demonstrate that the more accurate extraction of the underlying driving forces behind the input signals, using KSFA-based methods, leads to significantly improved model generalisation performance when compared with poorly extracting the driving forces, using LSFA-based models, and other similar techniques. This improved performance from

DKSFANN is due to these nonlinear underlying driving forces/trends, with less noise, providing more useful information for the neural network to create more accurate predictions.

4.5 Case Study 2 - Industrial Penicillin Production

The proposed method was next applied to an industrial case study based on a fed-batch penicillin production process that was created using a simulator known as PenSim 2.0 (Birol *et al.*, 2002), which is a commonly used benchmark process for soft-sensing (Zhang *et al.*, 2017; Gopakumar *et al.*, 2018; Yuan *et al.*, 2020a) and process monitoring (Sun *et al.*, 2011; Hong *et al.*, 2014). A detailed description of the process and the mechanistic model behind it is described by Birol *et al.* (2002). Figure 4.9 shows a process diagram of the penicillin production process.

Production of penicillin is an important process for the pharmaceutical industry and establishing effective process control and optimisation is critical. Creating a soft sensor to predict key quality variables, such as the concentration of penicillin, is therefore highly valuable. However, the process is extremely complex and so it provides a good case study for nonlinear methods.

To effectively evaluate the proposed method, 9 variables of those measured were selected for use as the model inputs, as illustrated in Figure 4.10, with the penicillin concentration as the output, displayed in Figure 4.11. The CO_2 concentration was omitted from the model inputs because it was extremely noisy and so provided unreliable measurements which could lead to poor model performance. Additionally, the biomass concentration was omitted to create a more realistic case study because typically biomass concentration requires offline measurements and so the frequent measurements, without time delay, that the simulation produced, would not be realistically available in an industrial process of this kind.

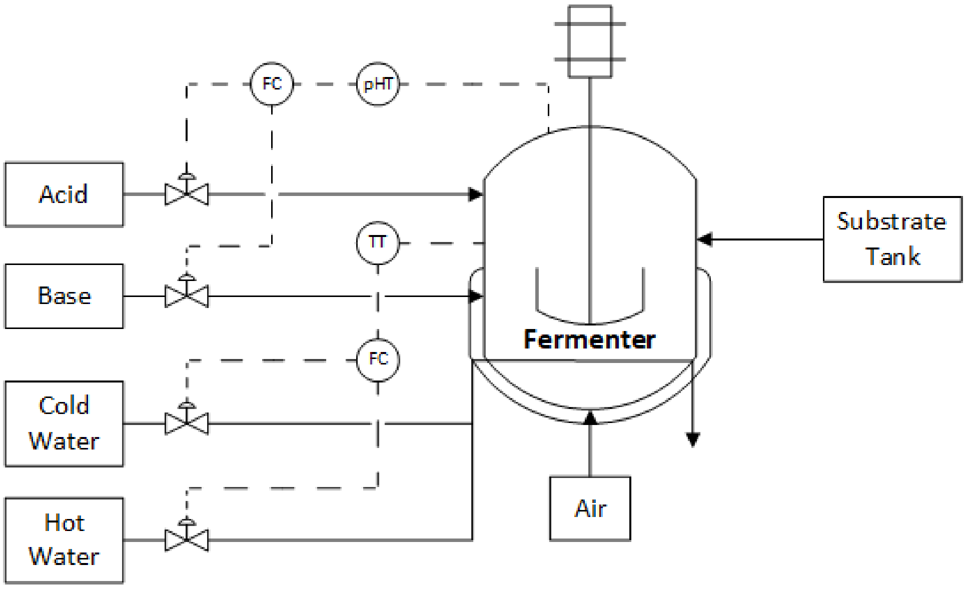

Figure 4.9 – Process diagram of the fed-batch penicillin process

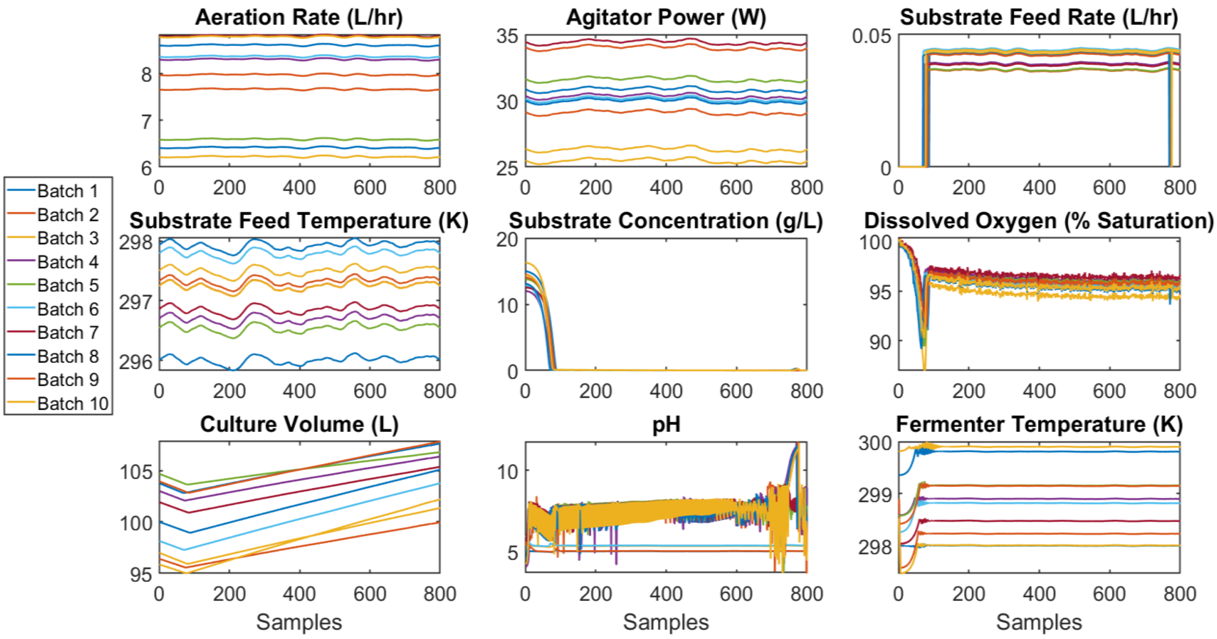

Figure 4.10 – Full input data across 10 batches for the industrial penicillin process case study

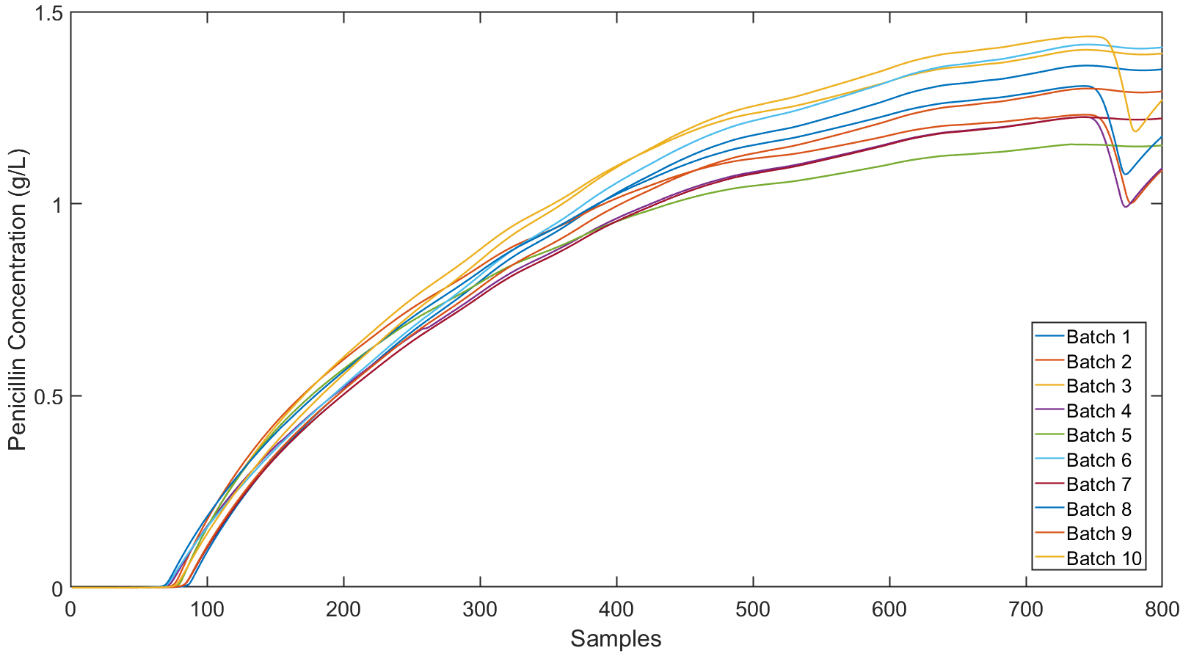

Figure 4.11 – Output data (penicillin concentration) for all 10 batches

In this study, 10 batches of data were created with a sampling time of 0.5 hr and a batch time of 400 hr, creating a total of 8000 samples with 800 samples per batch. To create a varied set of conditions across the batches, a range of acceptable values for each initial condition and setpoint was used, as provided by PenSim 2.0 (Birol *et al.*, 2002). These ranges are displayed in Table 4.3, and 10 random values within each condition's range were selected to create the 10 diverse batches. The 10 batches were randomly split into 6 for training, 2 for testing and 2 for validation. Just as with the first case study, models were created for *d* of 0 to 2, with the best model selected by the lowest RMSE on the testing data. All methods were trained in the same manner as in Section 4.4.

Table 4.3 – The ranges of values used for each initial condition and setpoint to generate the batch data

Initial conditions	**Range of values**
Substrate concentration (g/L)	12-18
Dissolved oxygen concentration (mmol/L)	1.16
Biomass concentration (g/L)	0.05-0.15
Penicillin concentration (g/L)	0
Culture volume (L)	95-105
Carbon dioxide concentration (mmol/L)	0.5-1
pH	4-6
Fermenter temperature (K)	298-300
Generated heat (kcal)	0
Setpoints	
Aeration rate (L/hr)	6-10
Agitator power (W)	25-35
Substrate feed flowrate (L/hr)	0.035-0.045
Substrate feed temperature (K)	296-298
Temperature setpoint (K)	298-300
pH setpoint	5-6

Table 4.4 provides the prediction results for each method. The most accurate soft sensor model was created by DKSFANN, with a validation RMSE 32.3% lower than the next best method, DKPCANN. These results can be confirmed by visual inspection, as demonstrated in Figures 4.14 and 4.15, where the predictions of DKSFANN clearly follow the trend of penicillin concentration better, albeit with an offset throughout. Additionally, the testing RMSE of DKSFANN is 30.9% lower than that of DKPCANN, confirming that the improved generalisation is not just a one off on one data subset. DKSFANN outperforms the other techniques due to extracting nonlinear underlying trends in the data through DKSFA, and inputting this crucial information into the neural network, leading to more accurate predictions. The significant improvements in model performance of the neural network-based methods, when compared with the linear regression-based methods, highlights the complexity of the penicillin process and the need for the nonlinear regression method. Just as with the numerical case study, the KSFA-based methods produce more robust models than the LSFA-based methods, with a 34.6% improvement from DLSFANN to DKSFANN and 44.3% from DLSFR

to DKSFR. Figures 4.12 and 4.13 show the first 6 extracted SFs on training data for both LSFA and KSFA respectively. These figures demonstrate that the SFs from KSFA contain different and more complex trends that do not simply correlate to the input variables. As shown in Figure 4.10, this process runs in batch mode initially until the amount of biomass reaches a certain level and it then changes to fed-batch mode by having the main reaction with continuous feeding of substrate into the reactor. It is interesting to note from Figure 4.13 that all the 6 nonlinear SFs have captured the switching between the batch and fed-batch modes. In contrast, the 2nd and the 4th linear SFs fail to capture this, as shown in Figure 4.12. Thus, the nonlinear SFs can be considered as being more representative than the linear ones in this penicillin production process. These underlying trends from KSFA are what help improve the soft sensor generalisation performance.

Certain data quality issues, particularly noise, are present in the input data. The key advantages of KSFA/SFA are the ability to extract slow varying features, meaning that noise is filtered out and is only present in the latter, faster SFs - which are omitted when the dominant (M) SFs are selected. By reducing the impact of noise, while also capturing key underlying trends, the SFA-based models improve performance over the other models. This can be visualised by comparing the neural network inputs for DNN vs DKSFANN: Figure 4.10 is the inputs that go into the neural network for DNN, lots of data quality issues are present. Comparing this with Figure 4.13, which shows some of the inputs to the neural network in DKSFANN, illustrates that the extracted SFs reduce the data quality issues over the original inputs.

Table 4.4 – Regression results for the penicillin process case study

	Training		Testing		Validation		
Method	R^2	RMSE	R^2	RMSE	R^2	RMSE	Hyper parameters
DLR	0.8707	0.1574	0.1198	0.6853	0.2932	0.5097	$d = 2$
DPLSR	0.5386	0.3018	0.5550	0.3053	0.4250	0.3426	$d = 2$, $k = 1$
DPCR	0.4685	0.3208	0.4515	0.3280	0.4632	0.3284	$d = 2$, $k = 4$
DLSFR	0.1882	0.3919	0.2063	0.4479	0.3619	0.3969	$d = 2$, $M = 2$
DKPCR	0.7348	0.2041	0.4812	0.3177	0.4682	0.3281	$d = 1$, $k = 10$, σ = 5.5, L = 9
DKSFR	0.7956	0.1983	0.6352	0.2760	0.6491	0.2933	$d = 1$, $M = 10$, σ = 5, L = 9
DNN	0.7225	0.2185	0.1569	0.5038	0.7479	0.2221	$d = 0$, $N_H = 4$
NARX NN*	*0.9392*	*0.0937*	*0.8983*	*0.1321*	0.6697	0.2883	$d = 2$, $N_H = 12$
DPCANN	0.8793	0.1519	0.8725	0.1493	0.7124	0.2352	$d = 0$, $k = 5$, $N_H = 29$
DLSFANN	0.9581	0.0850	0.7311	0.2301	0.6764	0.2637	$d = 2$, M =22, $N_H = 25$
DKCPANN	0.9859	0.0490	0.7671	0.2001	0.7682	0.2190	$d = 1$, $k = 9$, σ = 6.25, N_H = 28, L = 9
DKSFANN	**0.9817**	**0.0544**	**0.8870**	**0.1382**	**0.9005**	**0.1482**	**d = 2, M = 13, σ =6.5, N_H = 39, L = 9**

As with the first case study, the effect of d on the generalisation performance is investigated. This time, the optimal d was 0 for 18% of the methods, 1 for 27% and 2 for 55%. Therefore, introducing the time lagged samples improved generalisation performance for 82% of the methods. This shows that incorporating the time lagged values is significant for enhancing model performance, which is likely due to correlations between the time delayed process variables. It is also worth noting that this increase in model robustness due to increasing d is not negligible. For example, for DKSFANN the validation RMSE for a d of 2 was 0.1482 when compared with 0.1959 for d of 0, a 24.3% improvement. Comparing the NARX NN to DNN, the multi-step ahead predictions were worse than the DNN validation predictions. As with the first case study, this confirms that the external dynamic method used for DNN is better than a NARX model. As with Table 4.1, low R^2 are observed due to poor fitting when the number of retained features is too low, additionally for linear techniques which are not suitable for a nonlinear case such as this.

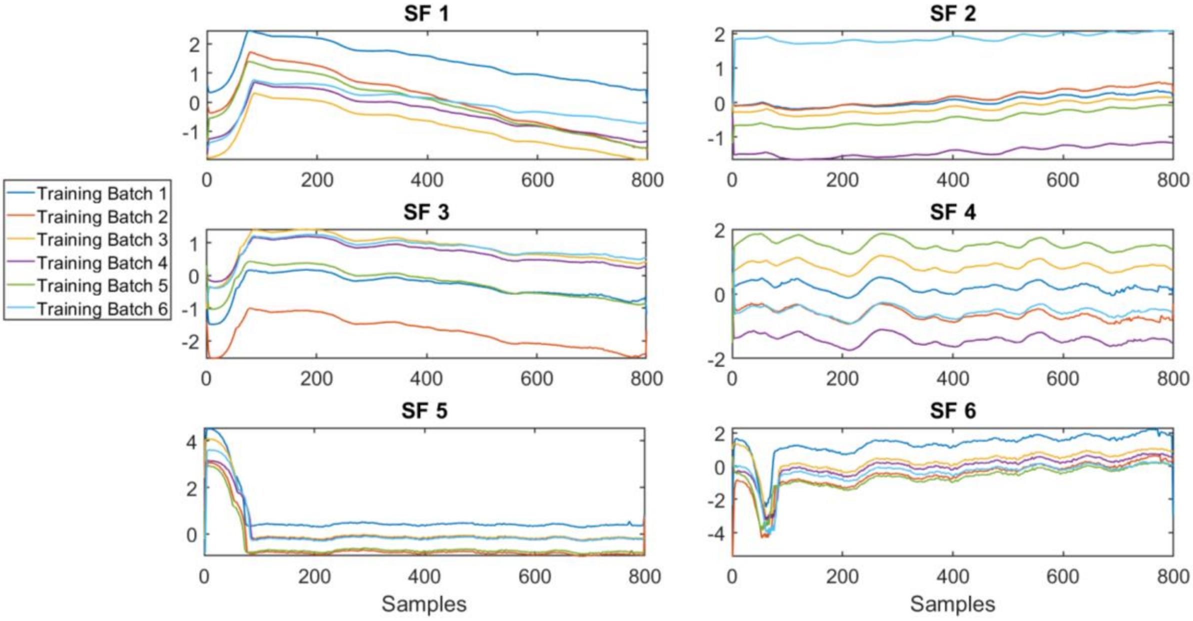

Figure 4.12 – First 6 extracted slow features on training data using LSFA

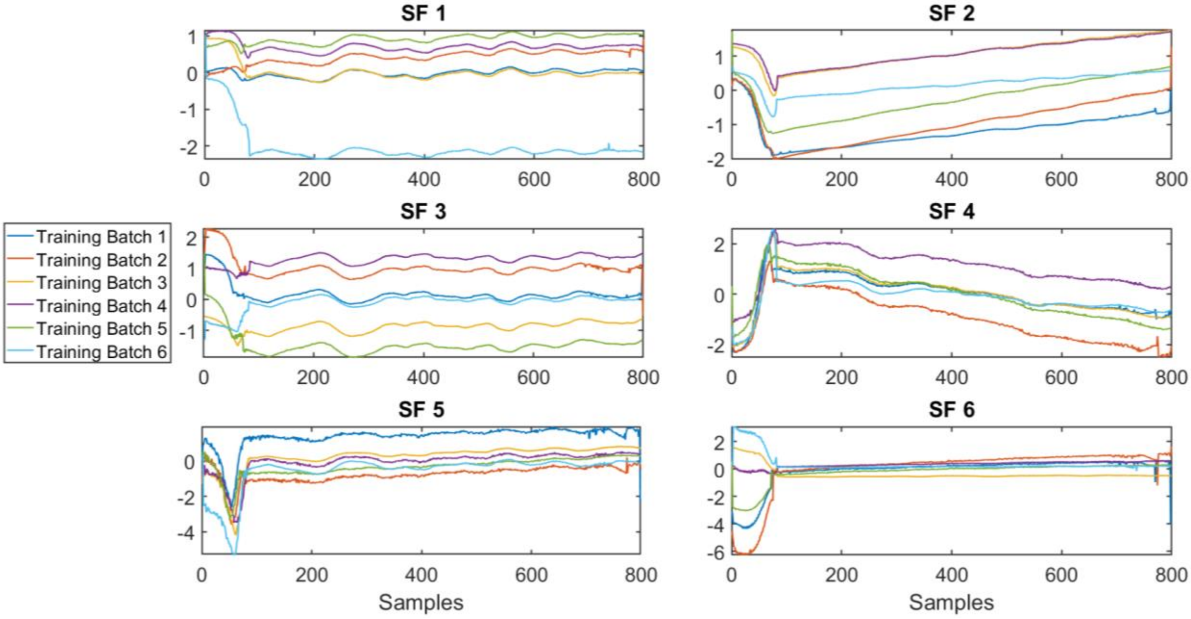

Figure 4.13 – First 6 extracted slow features on training data using KSFA

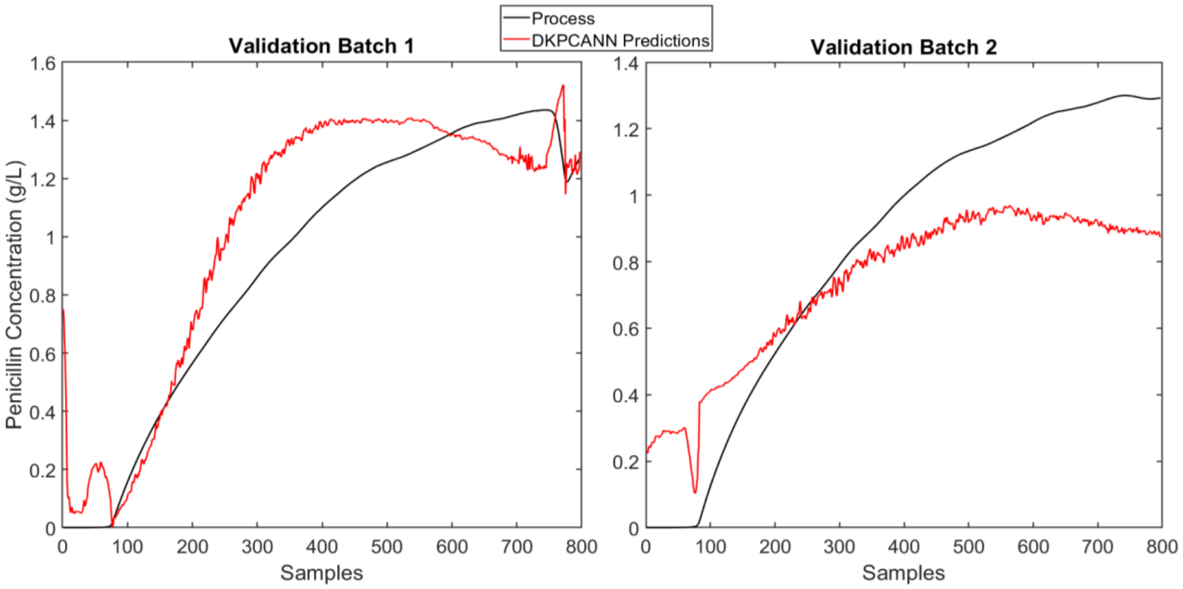

Figure 4.14 – Predictions on unseen validation data using DKPCANN for the penicillin case study

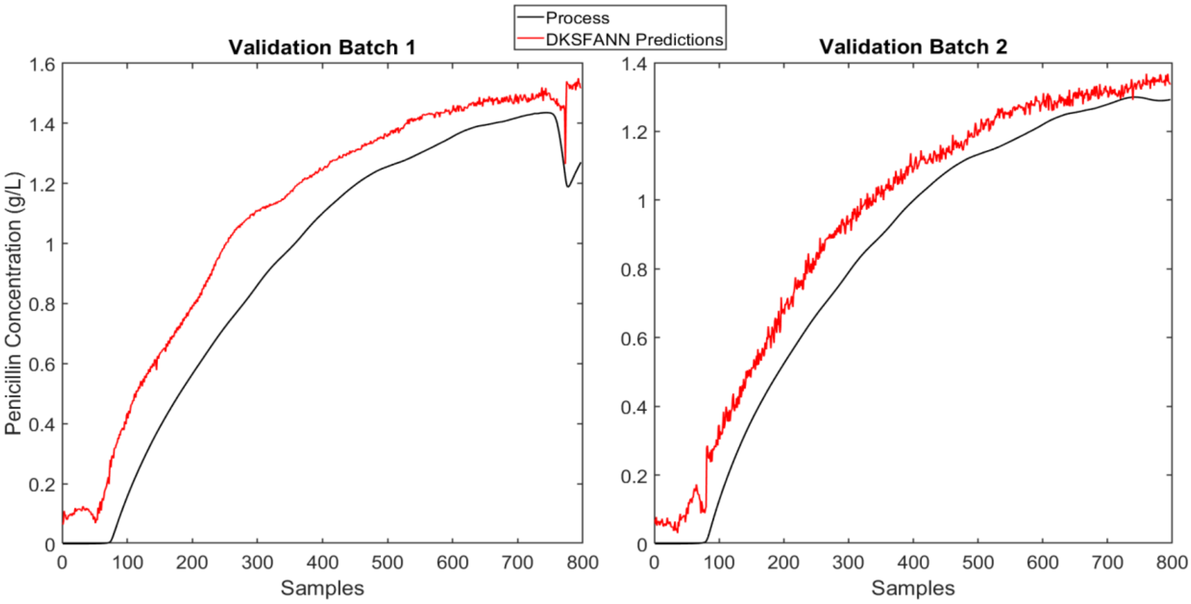

Figure 4.15 – Predictions on unseen validation data using DKSFANN for the penicillin case study

4.6 Conclusions

In this work, a data-driven modelling approach using kernel slow feature analysis and a neural network, referred to as DKSFANN, has been presented for improving soft sensor performance. The DKSFANN based method is used to extract driving forces and underlying trends in nonlinear situations where linear SFA fails to do so. It does this through the use of KSFA, which employs a kernel function to map the input signals into a higher dimensional feature space, so that SFA can extract the nonlinear SFs from the features in this new space. Then the nonlinear SFs are used as inputs to a neural network to create model predictions. The nonlinear SFs contain crucial underlying trends, with reduced noise, from the process and so, when input to the neural network, produce a more robust soft sensor. A neural network is utilised over linear regression for more complex situations where the relationships between the underlying trends/driving forces and the outputs are nonlinear.

The efficacy of this proposed method is evaluated on two case studies. Firstly, a nonlinear numerical example demonstrates the effectiveness of KSFA in extracting the driving forces behind a set of inputs, for both noisy and non-noisy signals, when compared with LSFA. Correlation coefficients are used to determine how effectively the driving forces were extracted in the SFs. The results clearly show that KSFA extracts the driving forces very well, where LSFA fails to do so because the inputs are nonlinear functions of the driving forces. Furthermore, DKSFANN outperforms all other methods for model robustness, with a RMSE on unseen validation data that is 31.7% lower than the next best method. The second case study is an industrial fed-batch penicillin simulation, which provides a complex process where creating a robust soft sensor is difficult. The PenSim 2.0 (Birol *et al.*, 2002) simulator was utilized to produce the data for this case study, where 9 input variables were used to predict penicillin concentration. As with the first case study, the DKSFANN model produced the most robust soft sensor by 32.3% when compared with the next best method. Both case studies demonstrate the efficacy of the proposed DKSFANN method for improving soft sensor robustness through the extraction of nonlinear driving forces.

Having developed effective nonlinear techniques for process modelling, the direction of this thesis now moves to using these techniques for applications on Sellafield Ltd’s nuclear waste vitrification process.

Chapter 5. Data-Driven Predictive Model for Average Pour Rate of the High-Level Waste from the Melter

5.1 Introduction

5.1.1 Nuclear waste vitrification

With the diminishing use of fossil fuels as a part of the attempts to reduce the global warming effect, nuclear energy is important as a viable alternative to fossil fuels, especially with electricity demand ever rising. When compared to renewable energy sources, such as wind and solar, nuclear energy has the advantage of continuous energy production. The major issues with nuclear energy come from the potential for major catastrophes (Chernobyl and Fukushima incidents), large construction and decommissioning costs, and the difficulty of nuclear waste management. In dealing with the highly radioactive waste arising from nuclear fuel reprocessing, vitrification is one of the advanced options that is utilised. Vitrification involves producing a glass product from the nuclear waste, which is of a lower volume and more radioactively stable (Taylor, 1985).

Spent fuel from nuclear reactors is reprocessed to maximise process efficiency and reduce waste. Around 3% of the fuel cannot be reprocessed, leading to a highly radioactive waste that needs to be properly dealt with. The nuclear waste vitrification process at Sellafield Ltd takes the Highly Active Liquor (HAL) waste arising from nuclear fuel reprocessing and converts it to a High-Level Waste (HLW) glass product. The HAL contains a mixture of fission products, additives, and impurities.

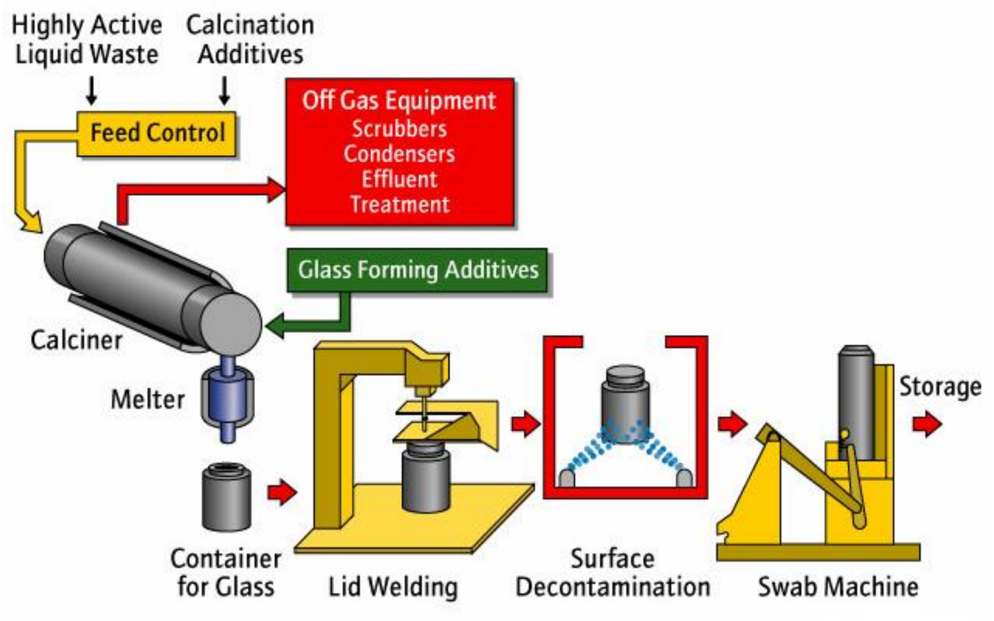

Figure 5.1 – Nuclear waste vitrification process

As illustrated in Figure 5.1, the HAL is combined with additives and fed into the calciner. The main calcination additives are lithium nitrate and sugar, which inhibit the formation of refractory oxides and reduce ruthenium volatility respectively. The calciner is a rotating, electrically heated furnace that is inclined by a few degrees. The calcination converts the HAL into a calcine powder by evaporation, drying and de-nitration. An off-gas system comprises of dust scrubbers, condensers, and NOx absorbers. Liquid effluents from the off-gas system are recycled to HAL storage.

The calcine is mixed with glass forming additives and melted at temperatures of at least 1000 °C to produce a homogenous mixture with the aid of air sparging. The glass mixture is then poured into stainless steel containers that are allowed to cool, welded shut and then decontaminated. The containers are then sent to the interim Vitrified Product Store (VPS).

Donald *et al.* (1997) provided a comprehensive review of the immobilisation of high-level nuclear waste, including the background of nuclear waste vitrification, types of HLW glass, and the processing stages. Additionally, previous work on determining the chemical durability of glass was discussed in this paper. Emphasis was placed on the importance of having adequate experimental designs for producing data for chemical durability.

Goel *et al.* (2019) presented an overview of the challenges, both short and long term, of HLW glass. In particular, selecting the correct composition of glass to ensure good durability and its effect on the alteration rate is discussed. The authors highlighted that theoretical understanding of the effect of glass composition on the mechanism and kinetics of corrosion is critical.

Chemical durability of the vitrified waste is a key variable because it determines how much radioactivity will leach into the environment when the containers eventually erode and encounter ground water. One of the main reasons for using vitrification for immobilisation of the HLW is because of the high chemical durability of the glass (Ojovan and Lee, 2011). The choice of glass is for many reasons, not just chemical durability, such as cost and ease of processing. Borosilicate glass is usually chosen because it reduces the melting point and improves the elemental oxides incorporation, while still having good chemical durability (Ferguson, 2013). A pure silica glass would give the best chemical durability; however, the required melting temperatures are much higher and so the operating costs make the process unfeasible (Backhouse, 2017). Understanding how the composition of the HLW and the types of glass affect the durability is critical to reducing process costs and improving the environmental safety. The effect of the composition of the borosilicate glass on the durability has been previously investigated. Harrison (2014) looked at the main compositional variations used in the UK and found that, in general, certain elements such as aluminium and silicon improve the durability.

Utton *et al.* (2012) explored the effect of the composition and pH of simulated groundwater solutions on the chemical durability of the waste glass. However, the authors found that the durability was similar for simulated groundwater of different compositions and pH when tested on Magnox HLW glass.

Work has also been done on chemical durability modelling using mechanistic models (Frugier *et al.*, 2009), however, mechanistic models are generally time and effort demanding. Kaunga *et al.* (2013) used bootstrap aggregated neural networks to model the chemical durability of HLW glass and demonstrated the reliability of this method over other common data-driven modelling techniques.

5.1.2 Melter pouring

Knowledge of the average pour rate of glass from the melter to the container is critical for process operations to be able to control glass behaviour. The pour rate should not be too low so that pouring takes too long and to prevent the glass not even pouring properly due to the high viscosity. Additionally, accumulation in the melter heel can occur. Knowing the pour rate before the pour can help avoid these issues, as well as preventing overfilling/underfilling of containers, which is very costly.

The pour rate is directly related to the viscosity of the glass, a higher viscosity leads to a lower pour rate, and vice versa. Previous attempts to predict pour rate were by modelling the viscosity of the glass, and then inferring pour rate from this. This is done because viscosity cannot be measured on the vitrification plant. Mechanistic models for viscosity of the glass are very complex and time consuming in development, with many assumptions often required to create the model. Data-driven efforts have been previously utilised to overcome these issues. Hrma (2008) used viscosity data for nuclear waste glasses to fit a simplified form of the Arrhenius equation to model the viscosity, with good results observed, however, the model is only valid for certain temperature and composition ranges. Ferguson *et al.* (2011) used bootstrap aggregated neural networks to model the viscosity of vitrified highly active waste for a range of compositions and temperatures to understand how these variables will affect a new feed. The model produced promising results; however, it was developed on a small number of experiments of simulated waste for a limited range of compositions. These experiments are costly and so creating a model with sufficient data to cover a wide range of conditions is difficult.

Determining viscosity also does not directly produce a specific value of pour rate, it only gives an idea of the pour rate due to the correlation between pour rate and viscosity. Creating a model to predict the pour rate directly will be more useful. Utilising large amounts of process data available (compared with limited experimental data) to create a robust model of the pour rate may be a better approach than predicting viscosity, with the limitations of previous attempts.

5.2 Data Collection and Pre-Treatment

Process data from Sellafield Ltd’s waste vitrification plant was collected in the form of once-per-pour (batch) measurements, including the variable of interest: average pour rate, that covered 162 batches over a period of 7 months.

Additionally, time series data for this entire plant operation period was provided to investigate whether any additional information could be extracted that would improve modelling performance over just using the once-per-batch data. This time series data is extensive and contains measurements such as temperature, which is known to affect the viscosity and hence the pour rate.

5.2.1 Variable selection - once-per-batch variables

Variables from the once-per-batch (OPB) measurements were selected as model inputs based on process knowledge of when measurements were available and their potential impact on the pour rate. OPB measurements are defined as single measurements made for one batch. Since the pour rate prediction is required at the start of a batch, only variables which have measurements available at or before this time can be incorporated into the model. The selected OPB variables are displayed in Table 5.1.

Table 5.1 – OPB variables selected as model inputs

OPB variables included in the model
Soak Time
Initiation Time
Target Weight
Melter Level when Target Weight Achieved
Nozzle Temperature when Pour Starts
I6 Power when Pour Starts
WO Incorporation Weight
Fuel Type Variable S
Fuel Type Variable L
Fuel Type Variable D

5.2.2 Variable selection - time series variables

Additionally, time series (TS) data was provided, which contained 23 different process variables that covered the time span of the 162 batches. Two variables (variables 8 & 9) were immediately removed due to containing mostly values of zero. Variables 21 and 22 were also omitted due to producing the same value before every batch. This left 19 TS variables that could be incorporated into the model.

Plots of the TS variables, with the batch start and end times superimposed, highlighted how these variables changed prior to, and at the start of, a batch. Based on this visual inspection, various trends pre-batch were found for certain groups of variables. Some variables displayed a more complex, nonlinear trend pre-batch and therefore a pre-determined number of samples pre-batch was included into the model to capture as much information on the state of these variables as possible. These variables are henceforth known as full trend variables, an example of which is shown in Figure 5.2 (scales removed for confidentiality). The pre-batch sample window for the full trend variables was set to 200 samples. This was determined from visual inspection of the data to include the meaningful information whilst also keeping model complexity down. Furthermore, the sampling rate was reduced by 10 times for this sample window to reduce the effect of noise and reduce model complexity. Since the TS data needs to be unfolded to be used as inputs to the model for predicting the OPB pour rate measurement, reducing the number of these samples to be included has a large impact on computational demand, since variable size affects computational effort much more then sample size.

There were three other types of trends in the TS data from pre-batch: a roughly linear trend (Figure 5.3), a step trend (Figure 5.4), and a roughly constant trend (Figure 5.5). To capture these trends in the fewest samples possible (to reduce computational effort), 1 to 2 samples were used from pre-batch. For the variables with a linear pre-batch trend, a sample window was defined by visual inspection such that the linear trend would be mostly contained within this window for all batches. The sample window was defined for each linear trend variable individually. To capture these linear trends, an average of a certain number of samples at the start of the window, and at the end of the window, was taken to produce 2 data points per batch, for each linear trend variable. 2 data points is sufficient to capture information for a linear trend. Several samples were averaged to account for the noise in the data, and this number of samples was defined by visual inspection and was individual to each variable as well. The step trends were handled similarly to the linear trends except that an average of samples was taken before the step and then after the step. For the constant trend variables, an average of several samples pre-batch was taken to capture the operating point of that variable just before the start of the batch. The averaging of samples in these cases was to reduce the impact of noise.

The classification of TS variables according to the type of trend is listed below:

- Full trend variables = 1,2,3,4.
- Roughly linear trend variables = 5, 15, 16, 17, 18, 19.
- Step Trend Variables = 10, 14, 23.
- Roughly constant trend variables = 6, 7, 11, 12, 13, 20.

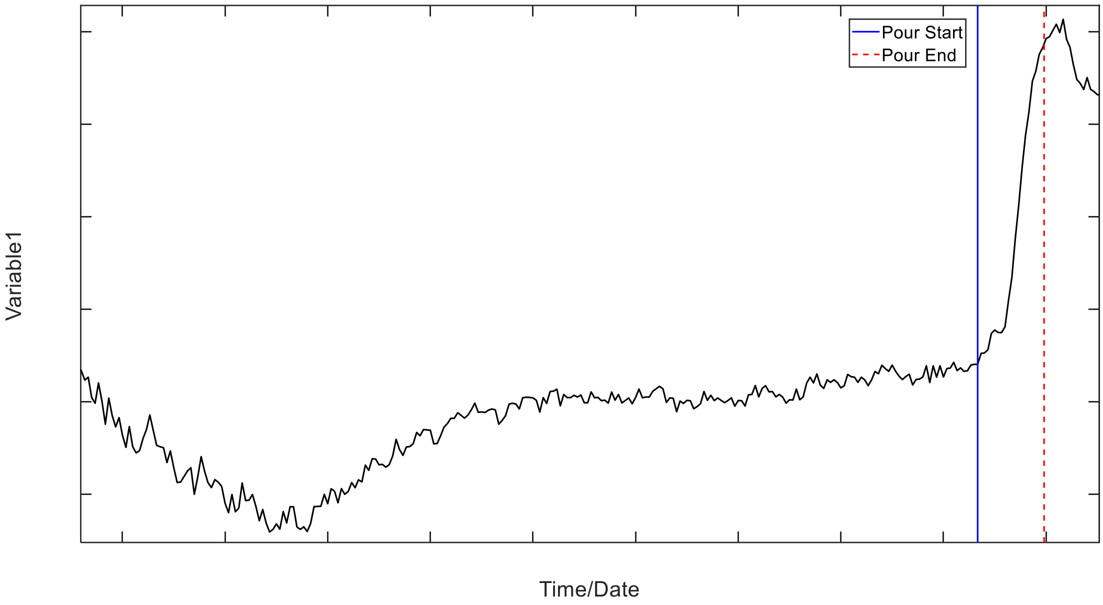

Figure 5.2 – TS variable 1 prior to a batch - example of full trend pre-batch

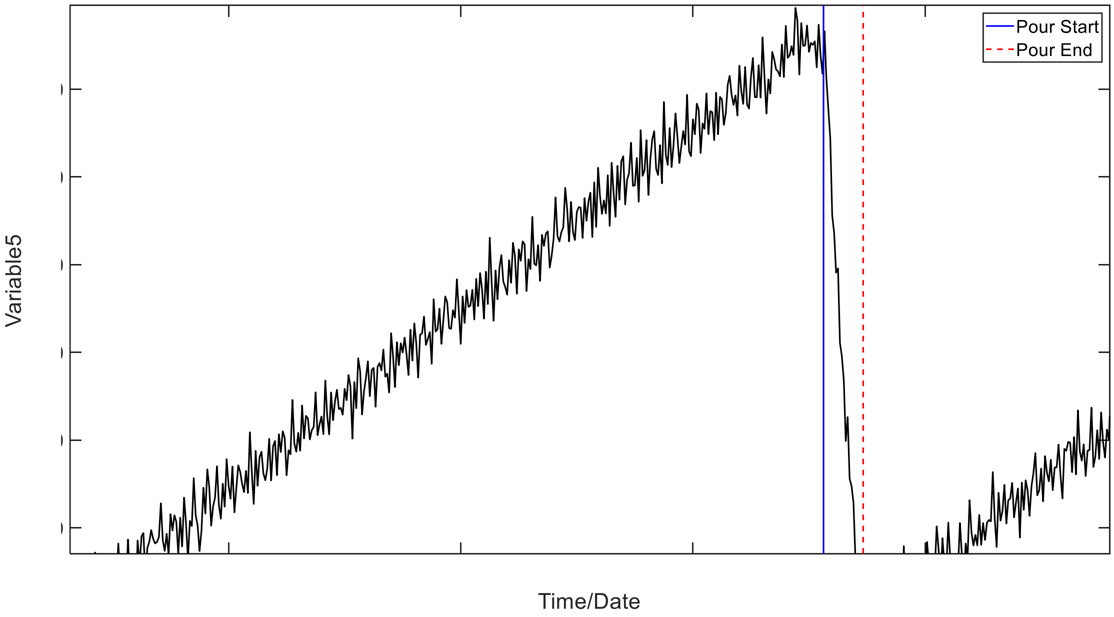

Figure 5.3 – TS variable 5 prior to a batch – example of linear trend pre-batch

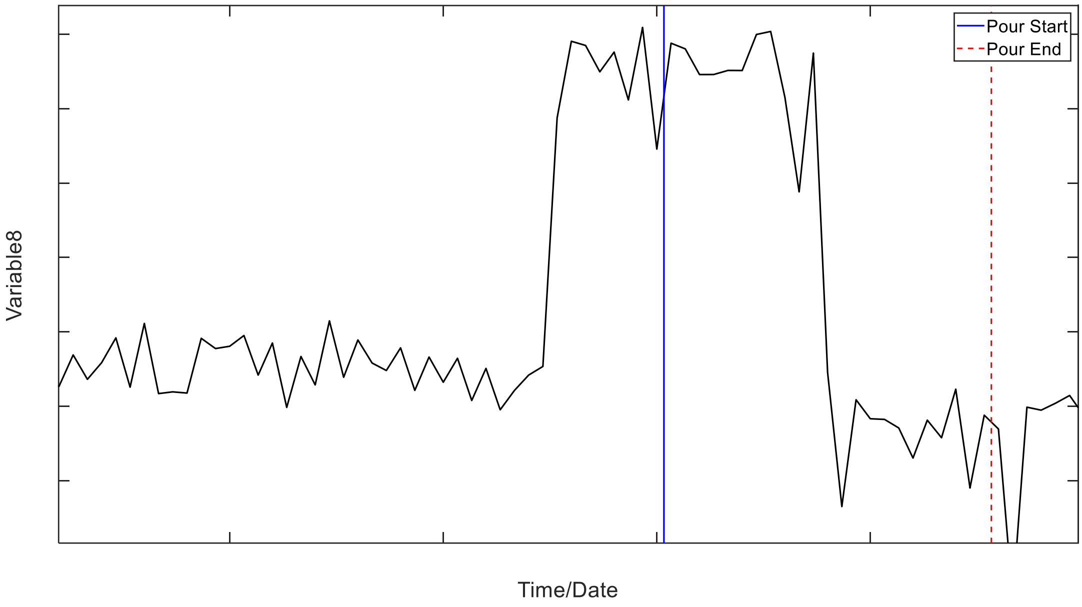

Figure 5.4 – TS variable 8 prior to a batch – example of step trend pre-batch

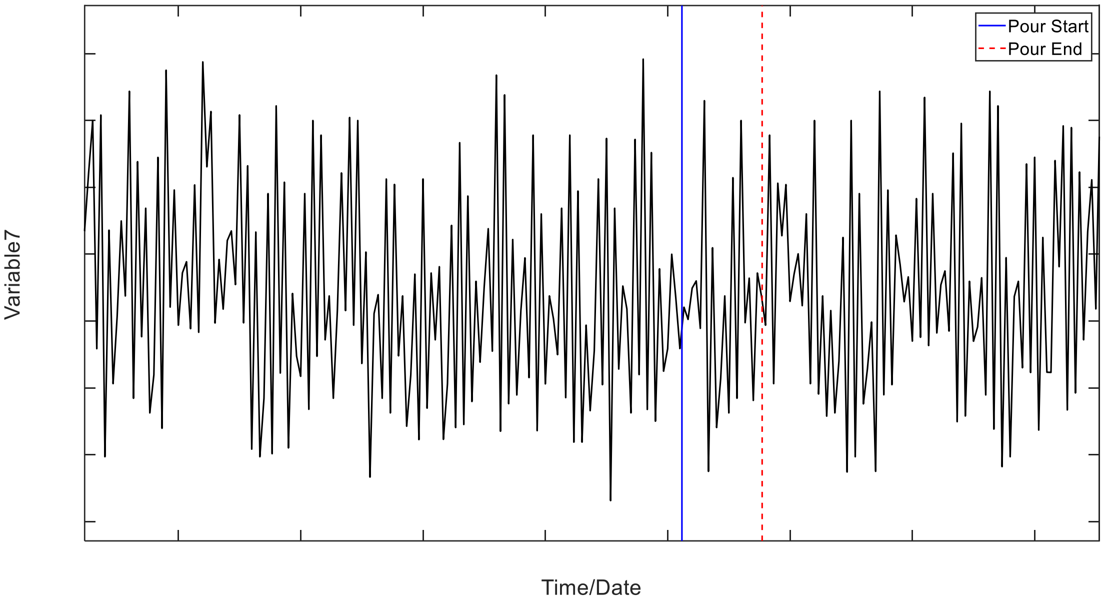

Figure 5.5 – TS variable 7 prior to a batch – example of a roughly constant trend pre-batch

5.2.3 Outlier detection and missing data

Outlier detection was carried out using two methods. Firstly, visual inspection of the data was carried out to remove any obvious outliers. This led to the removal of batches 4,5,6 from the OPB data due to having a significant number of outliers in sequence for TS variable 2. Batch 29 was removed due to having a clear outlier in the average pour rate. Since this is a OPB measurement and so is the process output, it would not be feasible to replace this measurement in any way.

Secondly, PCA was performed on the normalised OPB input data, with the outliers from visual inspection removed. Hotelling's T-squared statistic was calculated and plotted to attempt to detect any remaining outliers. As displayed in Figure 5.6, outliers were detected based on a 99.9% confidence limit to ensure that as much data was retained as possible due to the limited number of batches available. From this T^2 chart, 3 more batches were removed, corresponding to batches 1, 19, 46 from the original OPB data. Batches 79, 80 (relating to campaign E) and 53 were removed due to missing data.

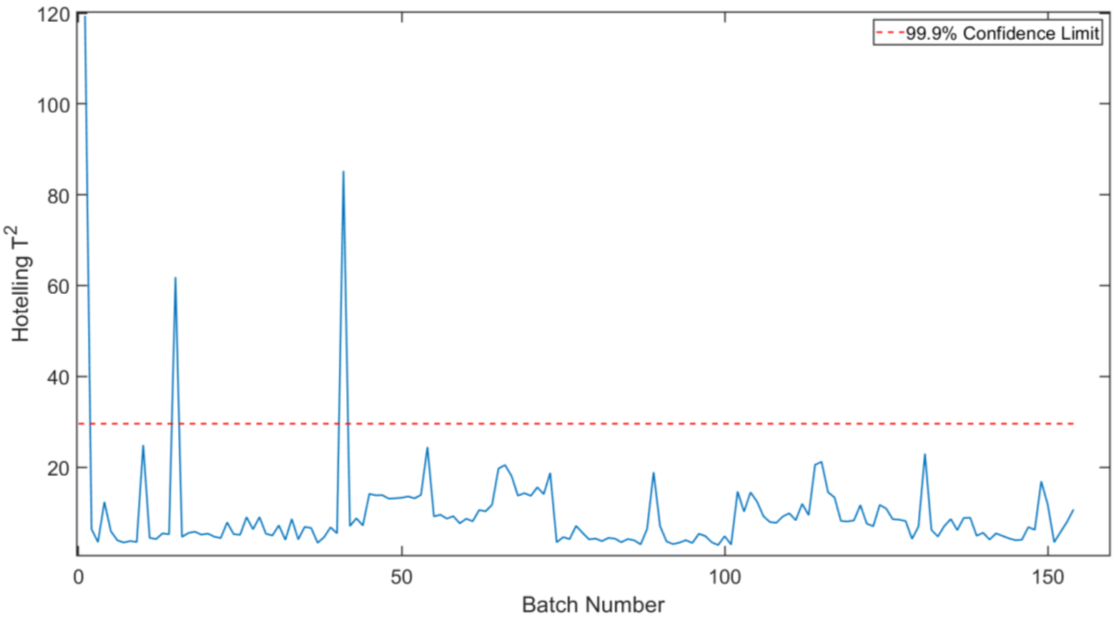

Figure 5.6 – Hotelling T^2 on the OPB Input Data, with a 99.9% confidence limit

5.2.4 Creating the input matrix for modelling

Before regression can be performed, the different inputs must be brought together into one matrix. Since many of the TS inputs contain more than one sample, batch wise unfolding was necessary so that the time-based samples could be included because the output (average pour rate) is only a OPB measurement.

Batch-Wise Unfolding of TS Data

Several approaches have been used previously to rearrange batch data so it can be used for regression techniques or other statistical pre-processing techniques. Nomikos and MacGregor (1995) originally proposed batch-wise unfolding as the most suitable method and it has remained the preferred choice up to recent process modelling/monitoring work (Zhang *et al.*, 2021).

Batch-wise unfolding involves converting 3D batch data into a 2D matrix so that it can then be used for regression techniques or other statistical pre-processing techniques. Consider a batch process with measurement data of I batches, J process variables, and K samples. Batch-wise unfolding converts the batch data matrix (I x J x K) into 2D matrix (I x JK), as demonstrated in Figure 5.7. In this way, each batch constitutes one row containing all the variables at all time points. This means that the time-varying behaviour of the process variables is captured.

Alternatively, time-wise unfolding creates a K x IJ matrix, such that each time point contains batches x variables. Variable-wise unfolding is the third option, which creates a matrix of batches x times for each variable (J x IK). Time-wise unfolding can be used to analyse sample variability, while variable-wise unfolding highlights the variability amongst variables. Both of these methods ignore the dynamics of the batch process and so can miss out on key temporal variations, hence batch-wise unfolding is the best option to capture the most information for process modelling (Yoo *et al.*, 2004).

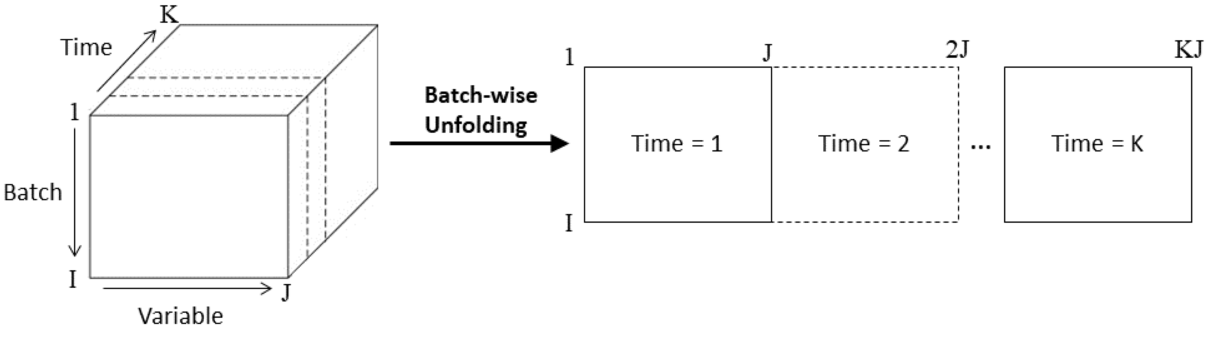

Figure 5.7 – Batch-wise unfolding of 3D batch data into 2D matrix (adapted from (Nomikos and MacGregor, 1995))

Batch-wise unfolding was carried out on each TS variable, whether it was a designated as a full, linear or step trend variable. The constant trend TS variables have one sample per batch (i.e., K = 1) and so batch-wise unfolding is not required.

Incorporation of the OPB with TS data

The batch-wise unfolded TS data was then combined with the OPB variables to create the total set of inputs for the model.

5.3 Linear Regression-based Modelling

A variety of regression techniques were applied to produce the model with the best generalisation performance. This involved starting with basic linear regression techniques, and then increasing the model complexity if the model performance was not satisfactory.

Before creating a model, the data must first be split into training, testing and validation data sets. Training data is used to train the model, testing data is used to determine hyper-parameters, and validation data is unseen throughout the model training process and so it is used to properly validate the model performance. The data was randomly split into 70% training, 15% testing and 15% validation in this case. A relatively large proportion of the data was used for training due to the small amount of batch samples available (154 batches after data pre-treatment). After splitting, the data was normalised to zero mean and unit variance.

The training data consisted of 108 samples with 118 inputs. Due to the number of inputs being higher than the number of samples, conventional linear regression (LR) would not work in this case due to an ill-conditioned $X'X$ matrix, where X is the input matrix. Therefore, attempting to reduce the dimensionality is important for this reason, as well as for reducing computational effort. Principal component regression (PCR) was employed to reduce the dimensionality, which involves applying principal component analysis (PCA) to X and then reducing the number of inputs in the form of retained principal components k, that are then used as the inputs to linear regression. The number of retained principal components k was determined by the lowest RMSE on the testing data.

5.3.1 Linear slow feature regression integrated with PCA

To attempt to improve model performance further, linear slow feature analysis (LSFA) was utilised. Applying LSFA is only meaningful for TS data and so the only logical place to use LSFA in this problem was with the full trend TS variables, because there are no underlying trends behind linear/step trends. LSFA can extract underlying driving forces behind data in the form of slowly varying latent variables, known as SFs. By capturing the underlying driving forces and slowly varying trends, with reduced noise, more meaningful information can be passed to the regression technique, leading to increased model accuracy. LSFA also reduces dimensionality (similar to PCA) through retained a reduced number of SFs (*M*), however, due to there only being 4 full trend variables, any dimensionality reduction would be insignificant compared to the total number of inputs. To prevent the ill-conditioned $X'X$ matrix when using LSFA with LR (LSFR), PCA was applied after LSFA (in the same way as in PCR) had been employed for the full trend TS variables.

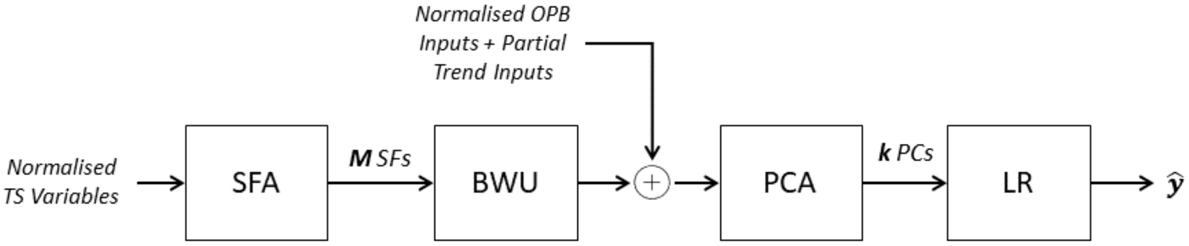

Figure 5.8 – Simple block diagram of the LSFA-PCR method

The results of the PCA and LSFA-PCR models are shown in Table 5.2. LSFA-PCR outperforms PCR across all data sets, with a 19.7% decrease in RMSE on unseen validation data predictions. The increase in generalisation for LSFA-PCR can be seen visually in Figure 5.9. The improvements through incorporating LSFA are due to extracting more meaningful information in the full trends, with less noise also.

Table 5.2 – Results of pour rate modelling using linear regression-based techniques

Technique	**Training**		**Testing**		**Validation**		**Hyperparameters**
	R^2	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	
PCR	0.8410	0.3969	0.8260	0.4690	0.7207	0.5821	k = 55
LSFA-PCR	0.8508	0.3845	0.8653	0.4126	0.8199	0.4675	M = 3, k = 57

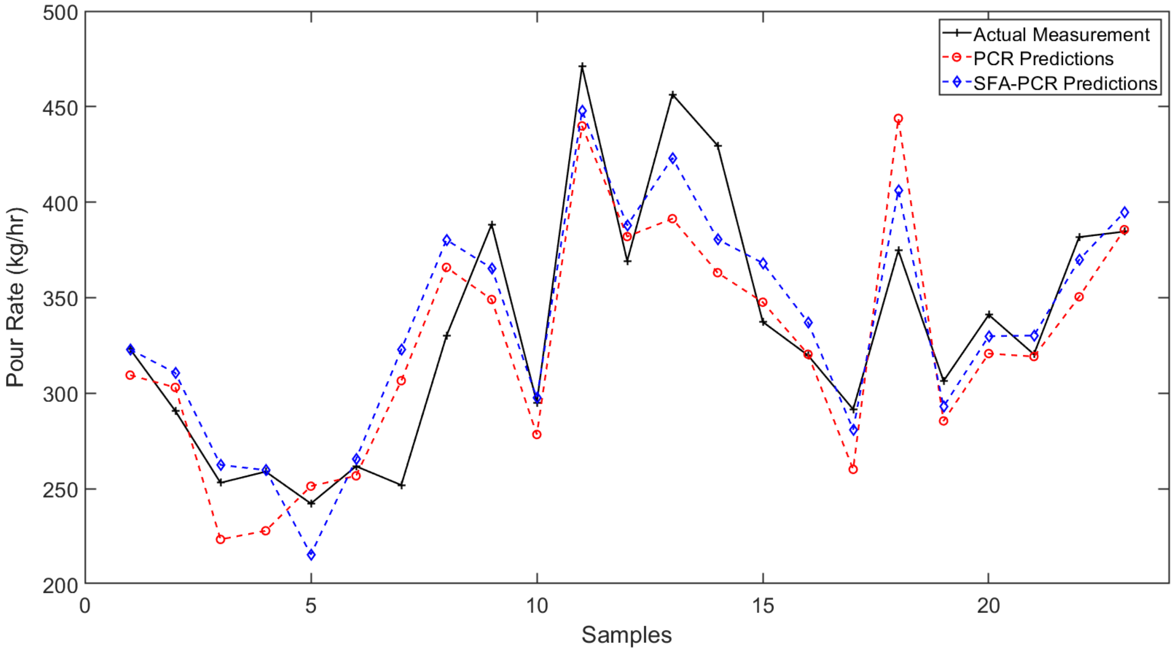

Figure 5.9 – Pour rate predictions on validation data for PCR and LSFA-PCR

5.3.2 Bootstrap aggregated linear models

From the results based on PCR and LSFA-PCR, further improvements could be made and so to improve model reliability and generalisation, bootstrap aggregation (bagging) was utilised. This involves creating multiple diverse models through random bootstrap resamples of the training plus testing input data combined, so that each model is trained on a slightly different subset of data. Then the prediction of each of these *O* models is averaged to create the final prediction. The top plot of Figure 5.10 illustrates how the prediction errors vary greatly between the individual bootstrapped models, indicating that a single model can be unreliable. Whereas the bottom plot of the figure shows the bagged model when the predictions are aggregated; the error levels off, indicating a more reliable prediction. The number of models *O* is selected when the testing RMSE begins to level off.

The impact of bagging on reliability is particularly evident for LSFA-PCR, where the validation RMSE was the lowest for all the techniques for the single model, but for BA-LSFA-PCR, the validation RMSE increased by 43.1%, despite an improvement in performance on testing data for the bagged model. In this case there was a reduction of generalisation when bagging, however it led to a more reliable model and gave a truer display of the model's generalisation performance.

Table 5.3 – Results of pour rate modelling using bagged linear regression-based techniques

Technique	Training		Testing		Validation		Hyperparameters
	R^2	RMSE	R^2	RMSE	R^2	RMSE	
BA-PCR	0.7153	0.5310	0.8344	0.4574	0.7019	0.6014	k = 28, O = 10
BA-LSFA-PCR	0.6588	0.5813	0.8993	0.3567	0.6311	0.6690	M = 3, k = 40, O = 10

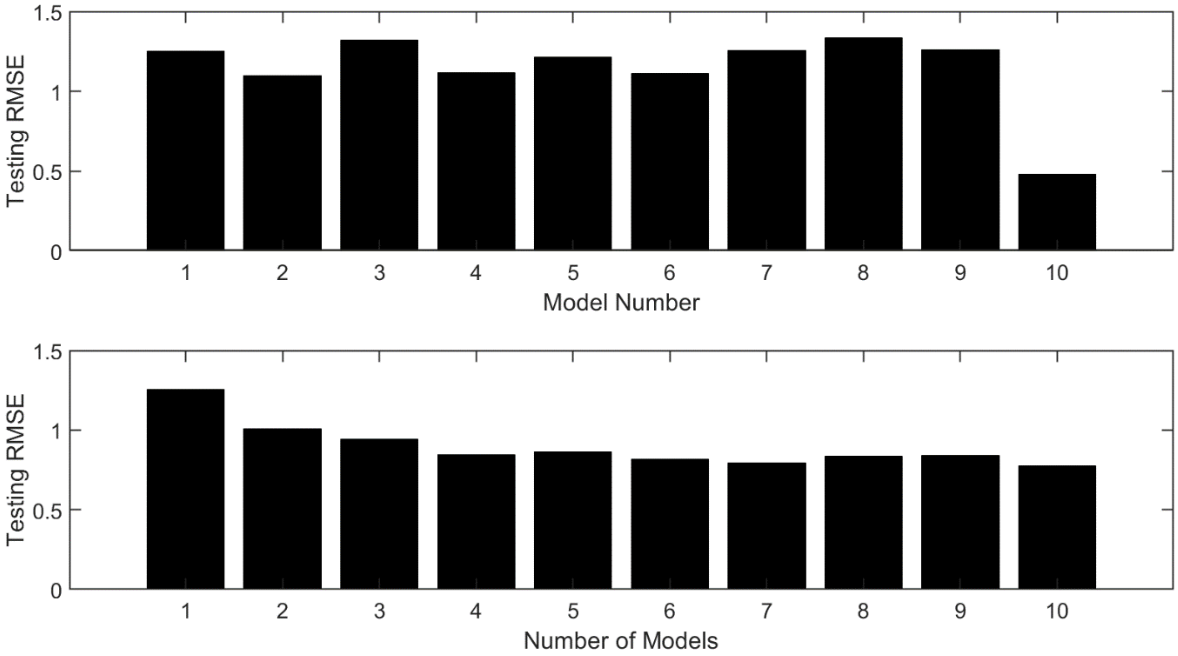

Figure 5.10 – RMSE on the testing data for each individual model (top) and for a bagged network for the number of models aggregated (bottom)

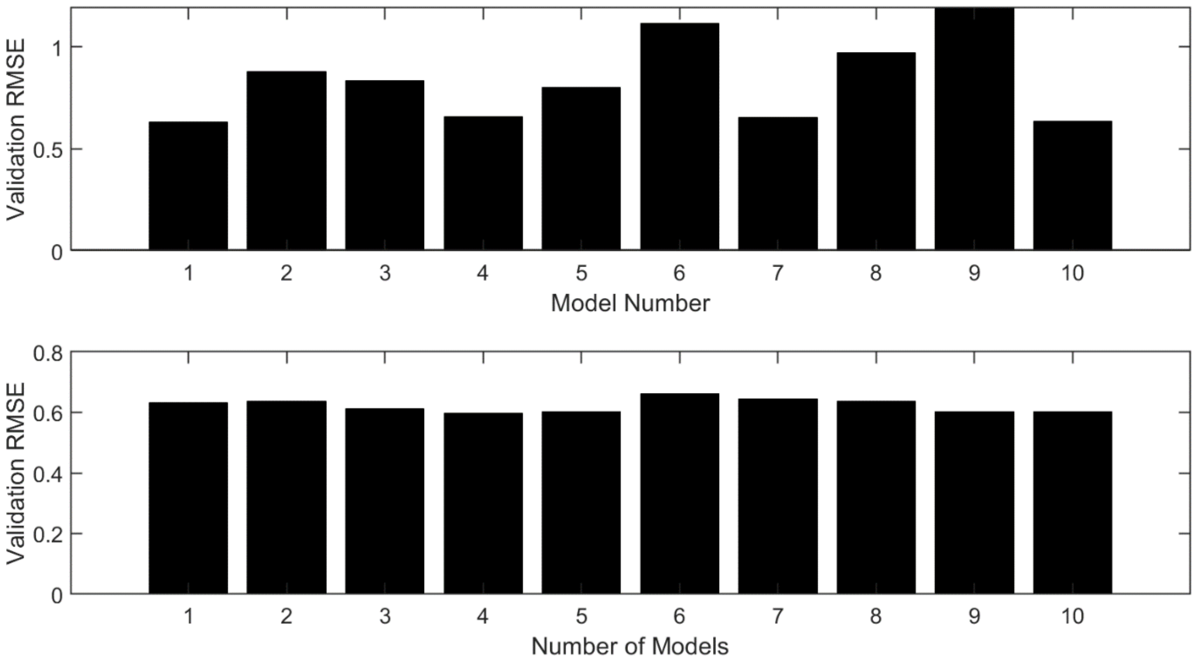

Figure 5.11 – RMSE on the validation data for each individual model (top) and for a bagged network for the number of models aggregated (bottom)

5.4 Neural Network-based Modelling

From the initial bagged LR-based results, there was still scope for improvement in model accuracy, potentially due to nonlinear relationships being present. Therefore, a single hidden layer feedforward neural network replaced LR as the regression technique, to capture any nonlinear relationships between the inputs and output to improve generalisation performance. The Levenberg-Marquart training algorithm was employed with a regularisation parameter of 0.5 to aid in improving generalisation performance. The number of hidden neurons N_H was determined by the lowest RMSE on testing data for a range of 3 to 40. Bagging was used again for these neural network-based models. The same pre-processing techniques of LSFA and PCR were used in the same way as in Section 5.3, with LR being replaced with the neural network (NN).

The results of all neural network-based techniques are shown in Table 5.4. The bagged NN-based models showed improvements in generalisation performance over the bagged LR-based methods, with BA-LSFA-NN having a 1.97% decrease in validation RMSE compared with BA-NN. This small improvement in generalisation performance can be attributed to extra driving force information extracted by LSFA for the full trend TS variables. However, due to the full trend variables making up a relatively small part of the total inputs, only this 1.97% improvement was observed. If the full trend variables were the only inputs, the improvement would likely be much larger. This is addressed in Section 5.5. Due to the large number of inputs not being an issue for NN as it does with LR (with the ill-conditioned matrix as described in Section 5.3), PCA following LSFA is not necessary and in fact produces worse model as BA-SFA-NN has a validation RMSE 8.0% better.

Table 5.4 – Results of pour rate modelling using bagged neural network-based techniques

Technique	**Training**		**Testing**		**Validation**		**Hyperparameters**
	R^2	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	
BA-NN	0.8897	0.3306	0.9054	0.3457	0.7327	0.5695	O = 20
BA-PCA-NN	0.8782	0.3473	0.8823	0.3857	0.6844	0.6188	k = 28, O = 20
BA-LSFA-NN	**0.8805**	**0.3440**	**0.9266**	**0.3045**	**0.7431**	**0.5583**	***M* = 3, *O*=20**
BA-LSFA-PCA-NN	0.8919	0.3271	0.9227	0.3126	0.7007	0.6027	M=3, k=40, O=20

5.5 Optimising Modelling Results Using Different Input Selection Methods

To assess whether the variable selection method in Section 5.2.1 is optimal, models were created using the same techniques as in Section 5.4, but with different variable selection. The main alterations would be whether the TS variables or OPB variables are more important to the model, or whether the extraction of the different trends in the TS data is necessary. Table 5.5 shows the results for different input selections for the three best techniques. "Different TS Trends + OPB" represents the variable selection described in Section 5.2.1. "Full Trend TS Variables Only" uses all the TS variables as full trends, with a 200-sample window pre-batch and a 10 down sampling rate, as was done for the full trend variables in Section 5.2.1. "Full Trend TS Variables + OPB" is the same as "Full Trend TS Variables Only" but with the addition of the OPB inputs. "OPB Only" uses the OPB inputs only. The BA-LSFA-NN results for "OPB Only" are not included because LSFA extracts time varying trends and the OPB inputs are not time series data so there is no theoretical justification for the use of LSFA in this case. Figure 5.12 illustrates the validation RMSE results from Table 5.5 to help show the magnitude of the differences in the RMSE.

These results demonstrate that the best model comes from using all the raw TS variables from pre-batch, where LSFA can extract underlying trends that improve model performance, when these SFs are used as the inputs to a bagged neural network model. The addition of the

OPB inputs decreases the model performance for all 3 techniques, showing that these variables have a detrimental impact on pour rate predictions. This is confirmed by the “OPB Only” RMSEs being the largest for all the input selection methods. This is likely due to the selected OPB variables not having a correlation to the pour rate.

If only BA-NN was considered, the TS input selection from Section 5.2.1 would produce the best model due to reducing the effects of noise, reducing model complexity, and capturing only the necessary information through the partial trends. However, for both BA-LSFA-NN and BA-PCA-NN, using the full trend for all TS variables produced the best model. For BA-PCA-NN, this is likely due to PCA providing dimensionality reduction and reducing multicollinearity more so when exposed to the large amount of full TS data. BA-LSFA-NN improves model generalisation performance by 9.89% for “Full Trend TS Variables Only” when compared with BA-PCA-NN. This improvement is due to LSFA extracting the underlying trends from each of the TS variables, providing more meaningful information (with less noise) to the bagged neural network, thus improving prediction accuracy. Additionally, the BA-LSFA-NN model had the best RMSE on testing data, which is how the optimal model is selected as the validation data must remain unseen.

When looking at the “Various TS Trends Only”, BA-LSFA-NN has a validation RMSE of 3.03% lower than the next best technique, BA-NN. This smaller improvement in performance than for “Full Trend TS Variables Only” is because there are only 4 TS variables with the full trend for “Various TS Trends Only”, and so the improvements provided by LSFA were only applied a small number of the total inputs (4 versus 21 full trend inputs).

Table 5.5 – Comparison of input selection methods on model performance for the top 3 techniques

Input Selection	**Training RMSE**			**Testing RMSE**			**Validation RMSE**		
	BA-LSFA-NN	**BA-PCA-NN**	**BA-NN**	**BA-LSFA-NN**	**BA-PCA-NN**	**BA-NN**	**BA-LSFA-NN**	**BA-PCA-NN**	**BA-NN**
Different TS Trends + OPB	0.3440	0.3473	0.3306	0.3822	0.3857	0.3457	0.5583	0.6188	0.5695
Different TS Trends Only	0.3636	0.3690	0.3597	0.3448	0.4243	0.3456	0.5092	0.5672	0.5251
Full Trend TS Variables Only	0.3188	0.3689	0.3875	**0.3045**	0.4427	0.4481	**0.4892**	0.5429	0.5826
Full Trend TS Variables + OPB	0.3475	0.3873	0.4219	0.3634	0.4719	0.4748	0.5432	0.5761	0.6729
OPB Only	N/A	0.5685	0.5466	N/A	0.7180	0.7148	N/A	0.7372	0.7342

The validation data predictions for the BA-LSFA-NN model using Full Trend TS Only input selection (best model) are presented in Figure 5.13. For a complex, real case such as this, these predictions appear to be good. Despite some clear errors, in general, the higher pour rates are predicted as higher, and vice versa for lower pour rates. Since the aim of the model is to give an indication of whether the pour rate is in certain ranges (i.e. relatively high or low), so that operators can get an early indication of the pour rate to prevent container overfilling/underfilling, this model largely works well for this.

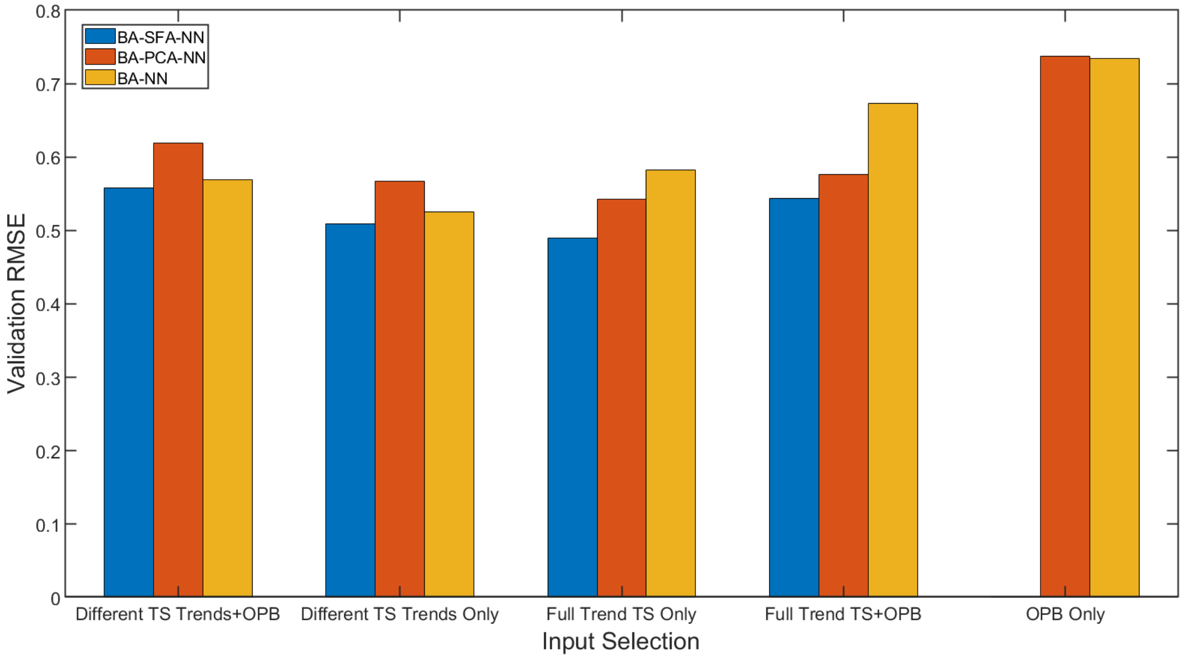

Figure 5.12 –Validation RMSE for the different input selection methods, for the top 3 modelling techniques

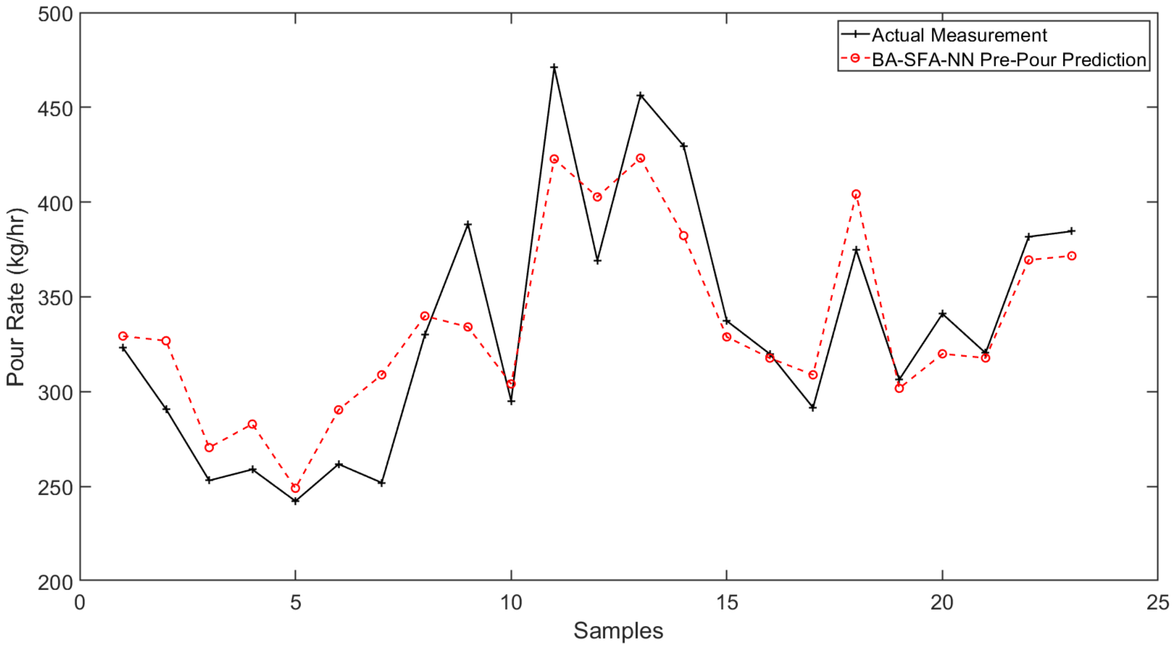

Figure 5.13 – Predictions of pour rate at the start of the pour on unseen validation data compared with actual values, for the best model using BA-LSFA-NN

5.6 Conclusions

Data from a section of the Sellafield Waste Vitrification Plant, that consists of the pouring of high-level waste from the melter to container, was gathered. This data consisted of process variables (in a time series form) and other measurements that were made once-per-pour. The aim is to create a predictive model of the average pour rate, a key quality variable, for the start of the pour because it is only known after the pour and this knowledge at the start of the pour can improve container filling efficiency.

The data was pre-processed to deal with outliers and missing data, using both visual inspection and Hotelling T^2. To make best use of the data and incorporate the most information into the model, input selection methods were developed to incorporate the time series data with the oncer-per-batch measurements by identifying trends in the pre-batch time series data, and then combining them with the once-per-batch data through batch-wise unfolding.

To create the most accurate data-driven model, techniques such as bagging, neural networks, LSFA and PCA were utilised to create a model with the best possible generalisation capability. The use of neural network-based techniques improved the model over less complex, linear regression-based models. Using LSFA as a precursor to a bagged neural network model (BA-LSFA-NN) improved prediction accuracy on unseen validation data by 9.9% over the next best technique, BA-PCA-NN, for the optimal input selection technique. Using different combinations of the time series trends and once-per-batch inputs was investigated to assess which input selection produced the model with the best generalisation. This optimal input selection, from the ones evaluated, was full trend time series variables only – that is, using the full trend of all the time series variables from a 200-sample window pre-batch, without any once-per-batch variables. This was only the best input selection method when using BA-LSFA-NN, which is due to LSFA being able to extract underlying trends from all the time series variables (not limited to 4 variables as in “different time series trends only”) that provided more meaningful information (with reduced noise and dimensionality) to the bagged neural network model; leading to improved generalisation performance.

To further improve model performance, additional techniques such as kernel SFA can be employed next, although it is likely more data is required to improve the model significantly from the current results.

Chapter 6. Early Detection of Dust Scrubber Blockages in the Nuclear Waste Vitrification Process Using Multivariate Statistical Process Control

6.1 Introduction

The dust scrubber (DS) is a vital section of the vitrification process because it recovers solids that have been carried in gas from the calciner to the primary off-gas system. It recycles these particles back into the calciner, increasing efficiency and preventing radioactive elements from getting into the environment. The DS involves a nitric acid flow down a column, with the off-gas flow from the calciner rising to the top through an input at the middle of the column. Plates in the middle of the column slow the flows to increase the exchange of particles from the gas to the nitric acid. The gas from the top of the column is then fed to a condenser.

Ustinov *et al.* (2019) discussed the types of off-gas clean-up systems and proposed an efficient and compact approach based on the characteristics of the off-gas. Morris *et al.* (1983) summarised the off-gas behaviour through experiments on 3 different types of waste using a full-scale pilot plant. The need for a filter with high efficiency and a Ru vapour absorber were emphasised from the results.

Due to narrow pipes in the DS, accumulation of particles often leads to blockages. Currently, the blockages are not detected as early as the plant would like, and they can lead to lengthy and, therefore, costly plant outages to clear. Blockages are typically detected by an increase in the differential pressure in the section of the DS where blockages occur. Using data from other process variables to develop a process monitoring approach to detect blockages early means that preventive measures could be taken and so costs associated with plant downtime and maintenance could be reduced. Additionally, an investigation into which variables could be causing these blockages would be very beneficial as these variables could be more closely monitored and controlled better to reduce the frequency of potential blockages.

Statistical process control (SPC) uses statistical techniques and process data to monitor process operation in order to detect process issues, such as process malfunctions and significant disturbances. SPC aims to provide feedback to plant operators as to whether the process is within its standard operating limits, it does not control process variables to optimise the process like standard process control does.

Univariate SPC looks at deviations in individual variables independently, typically through Shewart, CUSUM and EWMA control charts (Macgregor and Kourti, 1995). Examining each variable like this assumes all variables are independent of each other, when this is typically not the case in real, complex processes. This means that univariate SPC can miss many important events by not considering how all the variables effect one another.

Multivariate statistical process control (MSPC) can more effectively detect these process abnormalities by considering the co-variance of all process variables. It does this through statistical projection techniques, such as PCA, PLS and SFA.

This chapter looks at using an MSPC approach for early detection of the DS blockages, as well as investigating which process variables could be leading to the blockages via residual contribution plots.

6.2 Variable Selection and Data Inspection

To create a process monitoring model, process data from past periods of both normal operation and operation leading up to, and including, blockages was collected. The data provided by Sellafield Ltd contained measurements of 23 process variables for the waste vitrification process, such as temperatures, levels, flow rates and pressures. The measurements provided were for every minute for a period of 15 days, producing a total of 21600 data points for each variable. During this period 7 blockages in the DS pipe work were detected by operators and then resolved. The timings that each of the blockages occurred at and were resolved at were provided.

After initial discussion with a process expert, two of the variables corresponding to temperature measurements in the melter were removed because they were back up sensors and so provided no real measurements. Initially, all the other process variables were included because it was desirable to see if any of these variables could give early warning or be the possible cause of the blockages.

Figure 6.1 shows the time series plot of the 21 process variables (variable names omitted for confidentiality) with the red regions indicating DS blockages. Due to industrial confidentiality, the units and descriptions of the variables are omitted. Initial inspection shows that the variables 18 and 20 vary significantly when a blockage occurs (red regions), which is one of the main factors in detecting a blockage by plant operators currently. The data also shows two potential blockages occurring at around points 17004 and 20160, which may not have been recorded, or that resolved themselves.

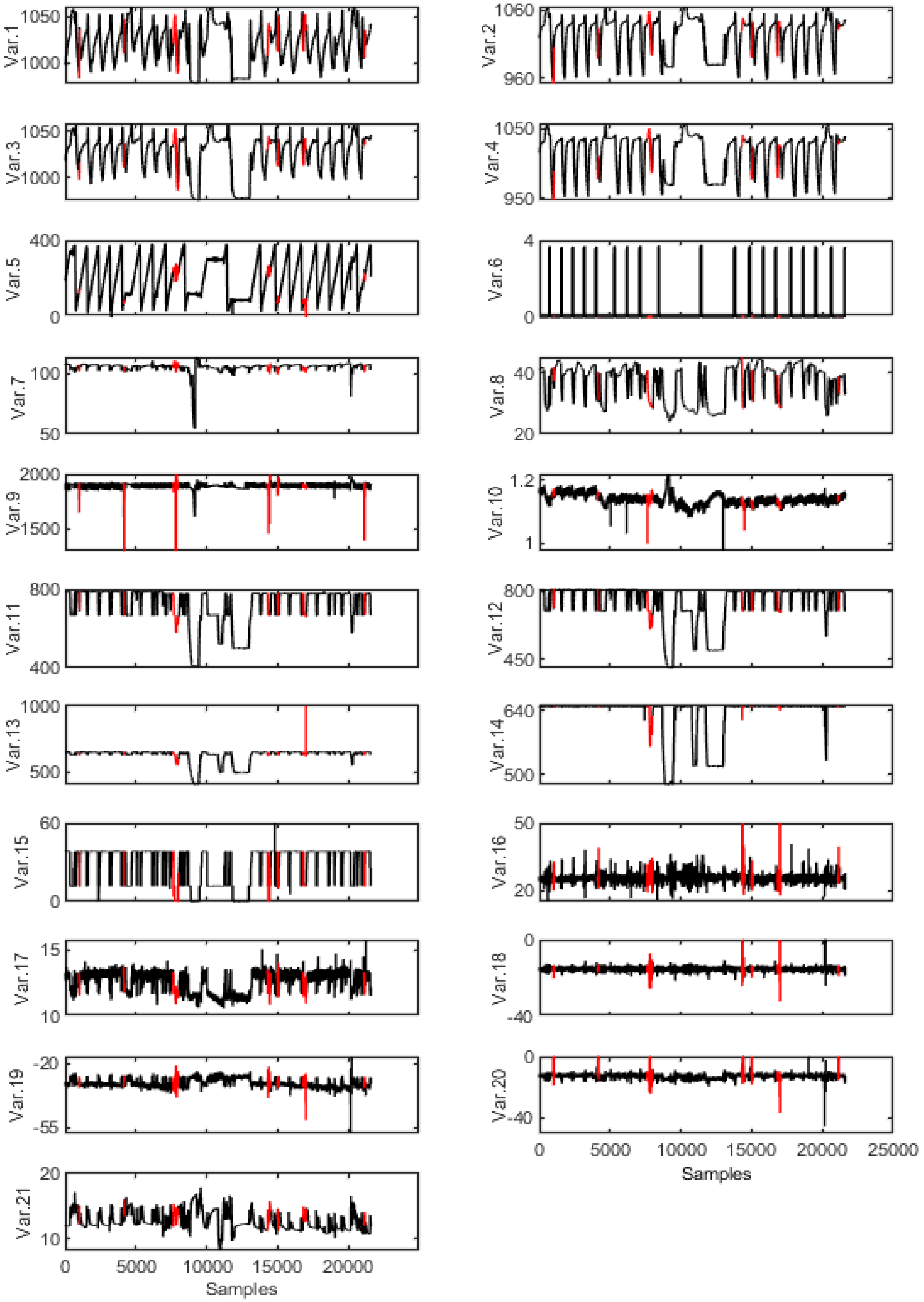

Figure 6.1 – Full data for all the selected 21 process variables, red data highlights when a recorded blockages occurred

Periods where variable 5 remained at a constant value indicated a disruption to the normal operation period, such as when a DS blockage was being cleared, as well as in some other periods of the data, likely due to other faults or maintenance.

6.3 Data Pre-Processing

6.3.1 Missing data

Missing data was occasionally present for the DS pressures after a blockage had occurred, while the process was offline for maintenance. These values were replaced with zeros to indicate the offline sensor.

6.3.2 Outlier detection and replacement

It is very difficult to apply fully automated outlier detection and removal strategies because there may be the removal of key process data and therefore, there should always be manual inspection, to either validate the results of automated procedures or to assist in outlier detection itself.

For this process data, outliers were initially detected by manual inspection because there were a few data points that were clearly outliers, possibly due to sensor malfunction.

Due to the batch-to-batch variation from the calciner section of the process, and the non-linear nature of some of the data, it would not be appropriate to replace the outliers with the mean of the data for some of the variables, particularly in the density data. Additionally, simply removing the sample would lead to problems with reducing the amount of samples. Replacement by interpolation of the nearest non-outlier samples was used for these outliers. For example, there was a significant outlier in variable 10 (density), as shown in Figure 6.2, which was replaced with interpolation of nearest non-outlier samples.

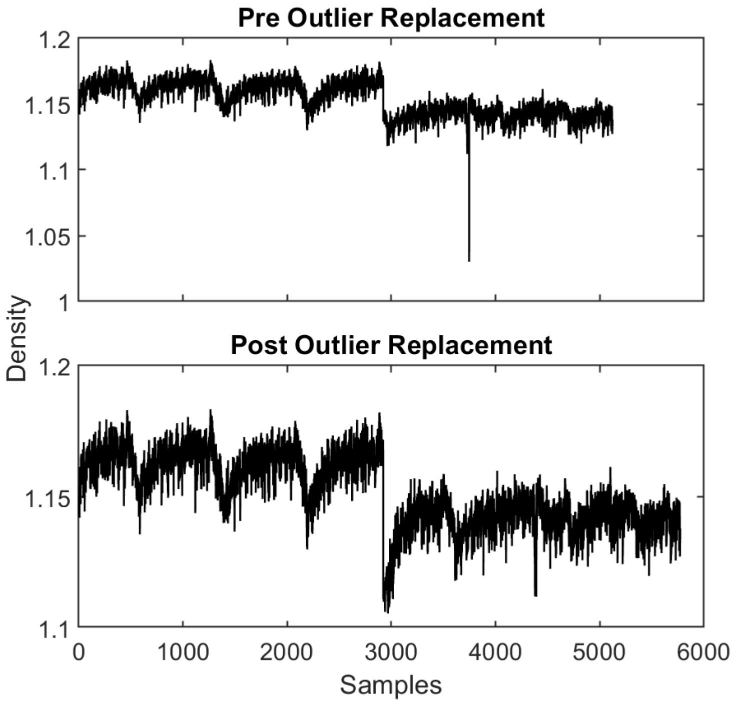

Figure 6.2 – Normal operation data for density, before and after outlier replacement

6.3.3 Data selection

Since expert process knowledge revealed that disruptions to the typical pattern of variable 5 indicated disruptions to process operation, normal operation data was taken as areas when the variable 5 pattern was consistent. This pattern is a linear increase followed by a sharp decrease, as displayed in the variable 5 time series plot in Figure 6.1. The selected normal operation data, with outliers removed, is shown in Figure 6.3, which produced a total of 5122 samples.

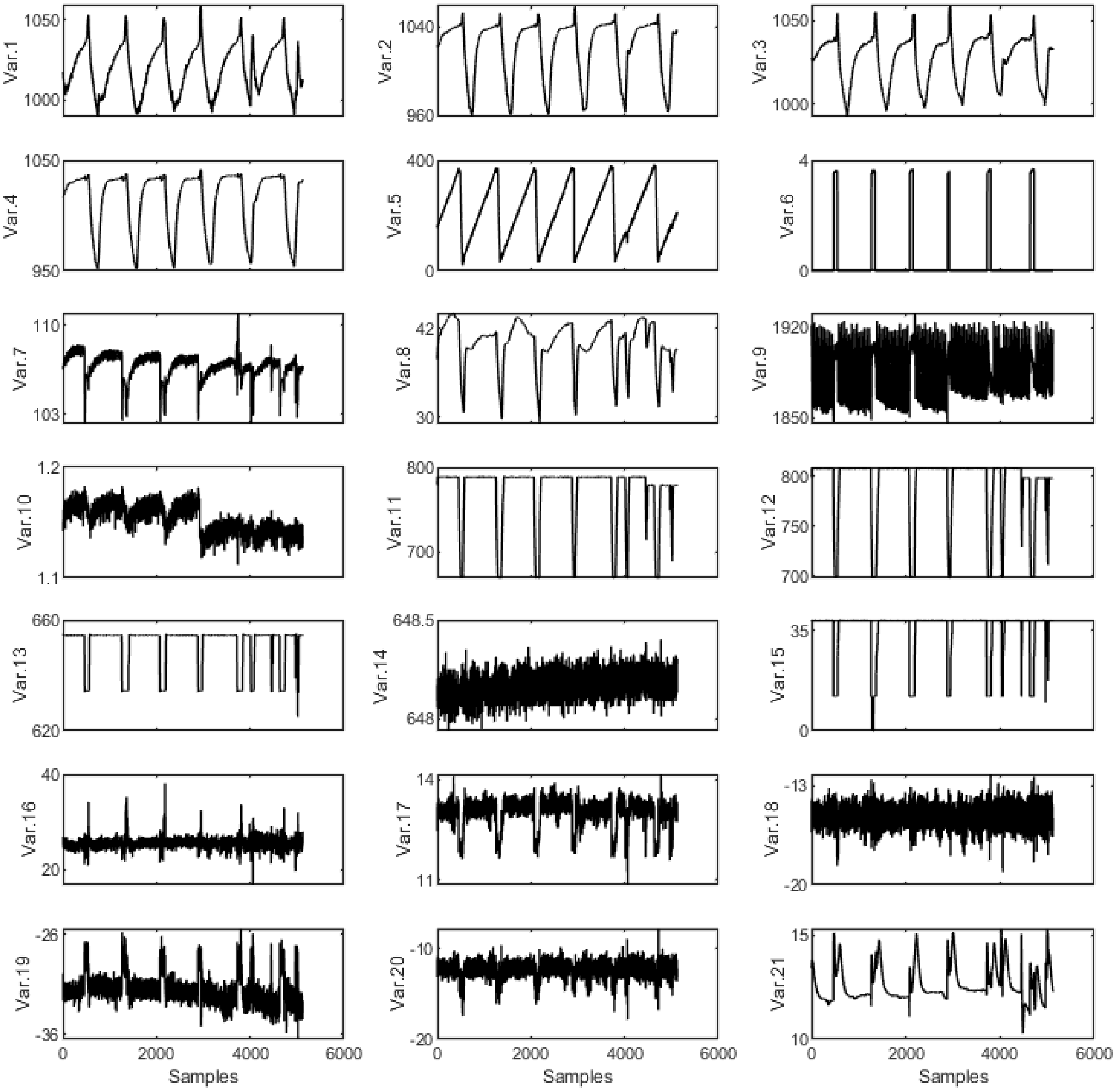

Figure 6.3 – Time series plot of the normal operation data for all the selected variables

The blockage data was taken as the period between blockage detection and resolution plus the prior 30 minutes of process operation so that any potential early detection can be discovered. Figure 6.4 illustrates the blockage data for each process variable, with each line, separated by a gap, representing one of the 7 blockages. The variation in this blockage data, compared with the consistency of normal operation data in Figure 6.3, shows that the blockages significantly disrupt normal plant operation.

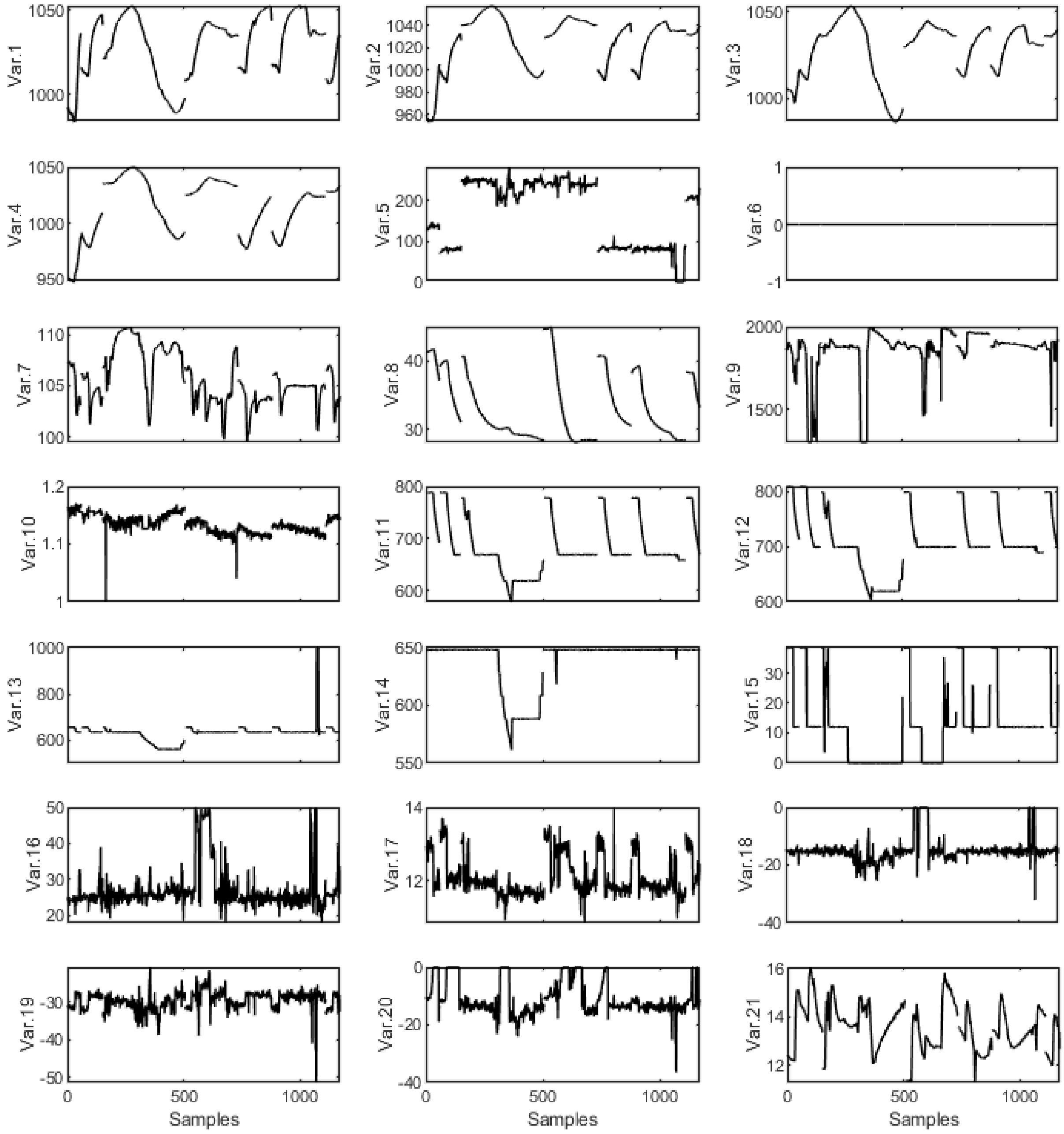

Figure 6.4 – Process variables when blockage occurred, with each subsequent line in each subplot representing the next blockage

6.3.4 Data normalisation

The data was scaled to zero mean and unit variance to avoid issues with the different magnitudes of the different process variables, as well as being a requirement for pre-processing and statistical process control techniques like PCA and SFA.

6.4 Principal Component Analysis

To attempt to provide early detection of the blockages and discover any variables that contribute to the blockages, PCA was applied to the data. PCA was used to reduce the number of variables to a smaller number of less-correlated principal components (PCs). PCA decomposes the data matrix ***X*** into a score matrix ***T*** and a loading matrix P_{PCA}, as shown below. This decomposition is typically carried out by singular value decomposition (SVD) (Qin, 2003). PCA is described in more depth in Section 2.4.1.

$$X = TP_{PCA}{}^{T} + E\,, \tag{6.1}$$

where *E* is the residual matrix.

From PCA, the cumulative explained variance for the first 15 principal components is displayed in Figure 6.5.

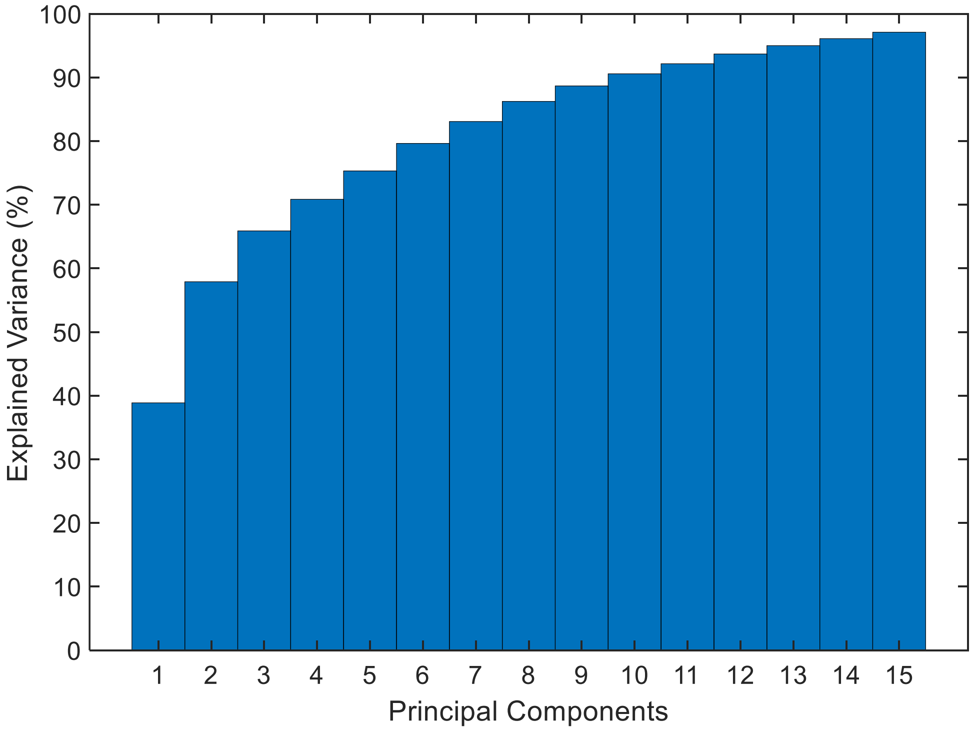

Figure 6.5 – Cumulative explained variance for the first 15 principal components

The number of principal components that were used for analysis was taken as over 90% of explained variance, which was 10 PCs.

6.4.1 Principal component analysis-based multivariate statistical process control

MSPC is a method of process monitoring that monitors process variables to ensure they remain within certain control limits. MSPC allows the information from many process variables to be included in a single chart, as opposed to monitoring deviations in individual process variables. Two monitoring charts based on Hotelling's T^2 statistic and the squared prediction error (SPE) are typically used, based on the PCA model (Qin, 2003).

Hotelling's T^2 statistic indicates variations in the principal components and is calculated from the following:

$$T^2 = x^T P_{PCA} \Lambda^{-1} {P_{PCA}}^T x = T\Lambda^{-1}T^T \,, \tag{6.2}$$

where *Λ* is a diagonal matrix of the eigenvalues of the covariance matrix of *x*, where *x* is input matrix.

The control limits for the T^2 statistic can be related to the F distribution, given the conditions that the process is normal and the data follows a multivariate normal distribution (Qin, 2003). Therefore, the process is considered normal if the T^2 is less than or equal to the following control limit of a significance level α:

$$T_\alpha^2 = \frac{k(N^2-1)}{N(N-k)} F_{k,N-k;\alpha} \,, \tag{6.3}$$

where *k* is the number of retained principal components, *N* is the number of samples and $F_{k,N-k;\alpha}$ is an *F* distribution with *k* and *N-k* degrees of freedom.

SPE provides a measure of the residuals between the data and the PCA model, as follows:

$$SPE = EE^T = x\left(I - P_{PCA}{P_{PCA}}^T\right)x^T. \tag{6.4}$$

Jackson and Mudholkar (1979) originally provided an expression for the control limit for SPE, Q_α:

$$Q_\alpha = \theta_1 \left(\frac{C_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0} \,, \tag{6.5}$$

where m is the number of variables, and

$$\theta_i = \sum_{j=k+1}^{m} \lambda_j^i \qquad i = 1,2,3 \tag{6.6}$$

and

$$h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}. \tag{6.7}$$

The T^2 statistic measures variation in the scores and a breach in the control limit typically indicates a process change with a large variance that does not necessarily violate the model. The SPE statistic measures variation that breaks the normal variable correlation, indicating an abnormal event; in this case a blockage (Qin, 2003). Therefore, if the T^2 breaches the control limits but the SPE does not, then it may not be a fault but a shift in process operation, which can still be a useful indicator to alert operators to a potential issue.

6.4.2 Blockage detection using MSPC control charts

To assess the detection of the blockages with the T^2 and SPE, the blockage data had to be projected onto the principal component (PC) space so that the T^2 and SPE could be calculated and directly compared with the normal operation data.

Firstly, the blockage data had to be normalised with the same mean and variance that the normal operation data was normalised with. This normalised data was then projected to the PC space by creating scores through multiplying the blockage data ($\boldsymbol{x_{blockage}}$) by the loading matrix:

$$\boldsymbol{T}_{blockage} = x_{blockage}P_{PCA}\,. \tag{6.8}$$

The T^2 statistic and SPE were calculated for the blockage operation data in the same way as the normal operation data, with the blockage scores replacing the normal scores.

The control charts for the normal operation data and the first blockage, with 90% and 99% control limits, are displayed in Figure 6.6.

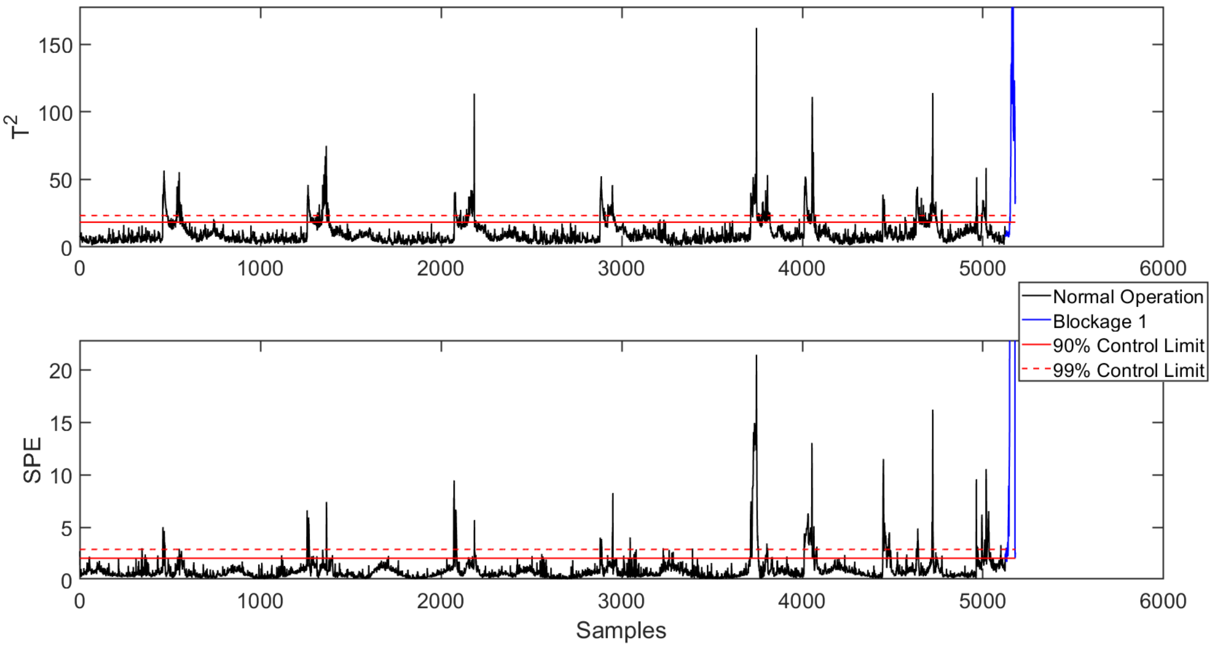

Figure 6.6 – T^2 and SPE control charts with 95% and 99% control limits for normal operation and blockage 1 operation

For both the T^2 and SPE charts in Figure 6.6, many sections of the normal data breach the 99% control limits, demonstrating that the PCA model or control limits are not adequate. As previously mentioned, breaching the T^2 may just indicate a process change that is still within acceptable conditions, and so breaches in the T^2 could be used a warning of changes in conditions but not outright detection of a blockage, therefore, some breaches during normal operation on the T^2 chart is acceptable if not associated with false alarms on the SPE chart. The SPE chart should be used to detect a blockage when accompanied with a breach in the T^2. There are some false alarms with the 99% control limit for the SPE chart, however, the blockage operating data is above the control limit for many samples and with a much higher magnitude. A higher SPE control limit, to the point where normal operation does not exceed the limit, would work for blockage detection due to the magnitude the blockage data reaches on the SPE chart. Figure 6.7 shows the control charts with a 99.99% control limit. In this case, there are no breaches of the SPE control limit during normal operation data and hence no false alarms. Therefore, the 99.99% was used. A blockage was defined as when the SPE and T^2 charts have a breach of the control limit with two consecutive samples.

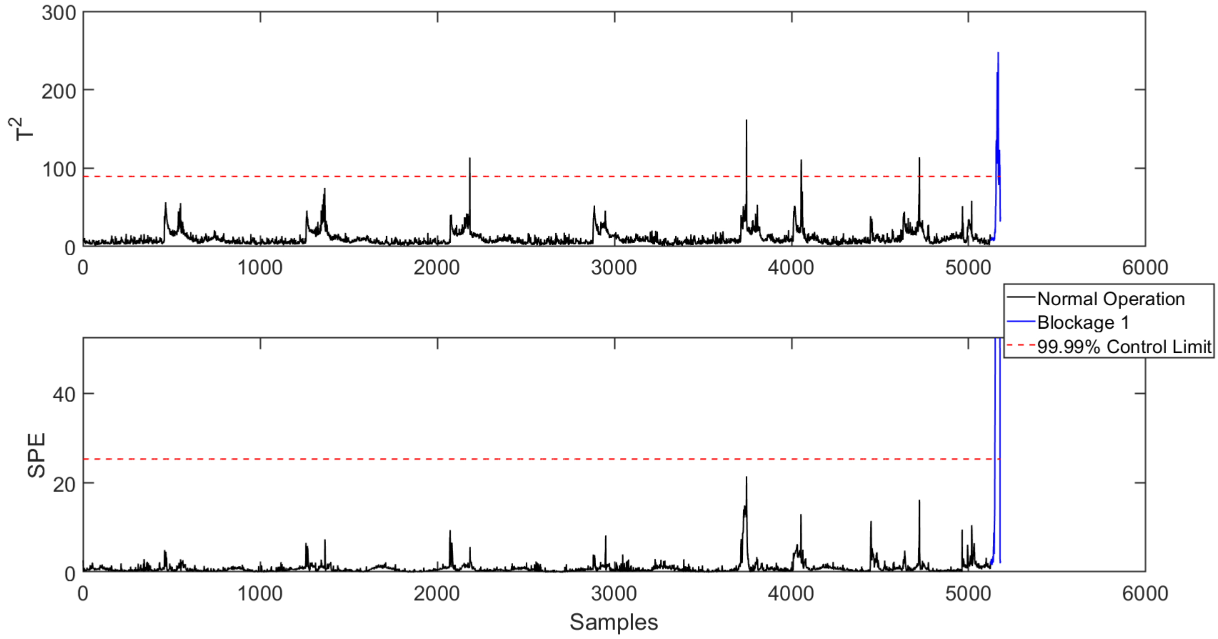

Figure 6.7 – T^2 (top) and SPE (bottom) control charts for the PCA model for blockage 1

The number of PCs were varied to assess the impact on the control charts but there was no significant difference in the charts, therefore retaining PCs above 99% variance is acceptable.

6.4.3 Early blockage detection

Proper detection of blockages without false alarms is important, however, the main aim is to detect the blockages earlier than previously detected on plant so that the impact of the blockages can be minimised, saving costs due to less plant down time for maintenance. For each of the 7 blockages, the time when the SPE control chart signalled the blockage was calculated and compared with the actual, on site detection time for the blockages. Table 6.1 shows the relative detection time (defined by Equation (6.9)) for each blockage and the average of all blockages. A negative relative detection time indicates that the control chart detected the blockage earlier than it was detected by on site. These results show that, on average, a blockage was detected 4.43 minutes earlier with the control chart when compared with on site. Two blockages were detected later than on site, but only by 1 and 2 minutes.

$$\text{Relative Detection Time} = \text{Control Chart Detection Time} - \text{On Site Detection Time} \qquad (6.9)$$

Table 6.1 – Relative detection times for each blockage using a PCA-based control chart.

	Relative detection time (min)
Blockage 1	-7
Blockage 2	-3
Blockage 3	1
Blockage 4	-7
Blockage 5	-13
Blockage 6	2
Blockage 7	-4
Mean	-4.43

6.5 Dynamic Principal Component Analysis-based MSPC

To try and better capture the variation in the data to improve early blockage detection, time lagged correlations were incorporated into the model. Dynamic PCA (DPCA) includes time lagged variables, up to a time lag *d*, into the data matrix to create an input matrix *X(t),* as described in Section 4.2.2.

The value of *d* was tested from 1 to 4 to assess its impact on the results of the T^2 and SPE control charts. For all the number of time lags, the effect on the control charts was minimal for normal operation data when compared to regular PCA. However, for early detection of the blockages, there is an improvement over PCA, as shown in Table 6.2. The optimal *d* was selected based on the best mean relative detection time for the first 3 blockages. Only the first 3 blockages were used so that the last 4 blockages were a true validation test. Increasing *d* led to an increase in the magnitude of average relative detection time up to a *d* of 4, where the improvements plateaued. Therefore, the optimal *d* for DPCA of 3 was selected. This was also the optimal *d* across all 7 blockages, so the selection was correct.

These DPCA results show an improvement of 59.7% in average relative detection time, as well as an improvement in all blockages except blockage 4, where the relative detection times for PCA and DPCA are the same. This improvement in DPCA can be attributed to the dynamics being included, providing information on temporal variations and time lagged relationships.

Table 6.2 – Relative detection times for each blockage using a DPCA-based control chart (*d* = 3)

	Relative detection time (min)
Blockage 1	-9
Blockage 2	-7
Blockage 3	-23
Blockage 4	-7
Blockage 5	-16
Blockage 6	-6
Blockage 7	-9
Mean	-11.00

Figure 6.8 shows the control chart for the DPCA model for blockage 1. Several points breached the control limit for SPE, however, these were only single points and also not accompanied by a breach in the T^2 chart. Therefore, there were no false alarms for this DPCA model.

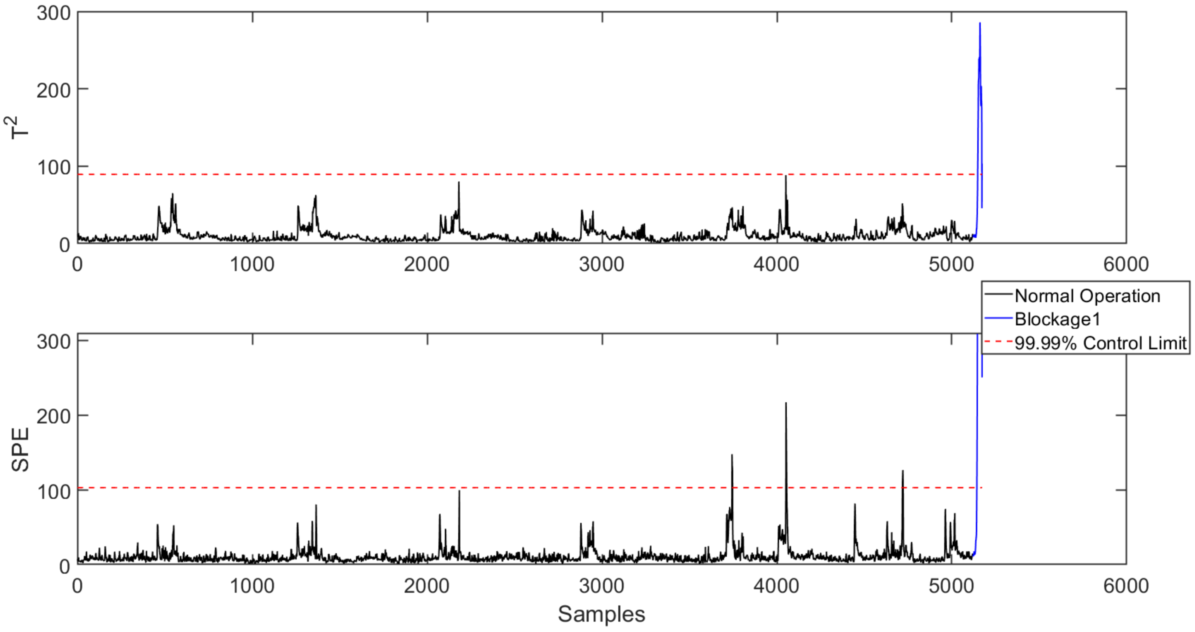

Figure 6.8 – T^2 (top) and SPE (bottom) control charts for the DPCA model (*d* = 3) for blockage 1

6.6 Slow Feature Analysis-based MSPC

Due to the large variation over time for many of the process variables, as well as the noise in some of the variables, it will be beneficial to extract slow varying features to better capture these temporal variations, which can further aid in early blockage detection. SFA extracts slow varying, underlying trends from the data, with noise reduction via retaining only the slower features. A description of linear SFA is presented in Section 3.2.

Relating the SFA theory to MSPC, T^2 and SPE statistics related to the SFs can be developed similarly to as in PCA. These statistics have been described and applied for SFA-based process monitoring in previous works (Shang *et al.*, 2016a; Gao and Shardt, 2021; Xu and Ding, 2021). Both statistics are described below:

$$\mathrm{T}^2 = s_M s_M^{\mathrm{T}}\,, \tag{6.10}$$

$$SPE = EE^T = z\big(I - P_{SFA}{P_{SFA}}^T\big)z^T, \tag{6.11}$$

where *M* represents dominant SFs, analogous to the retained number of principal components in PCA, P_{SFA} is the derived eigenvector from SFA, *z* is the sphered input signal, and *s* is the extracted SFs (See Section 3.2 for more detail on the derivations P_{SFA}, *z* and *s*).

The dominant SFs represent the slow varying trends, with the residual SFs mostly containing noise. As discussed in Section 3.3**,** the selection of *M* can be easily carried out using the scree plot, which is simply a plot of the eigenvalues from the derivation of the SFs. Figure 6.9 shows a clear elbow in the scree plot at 15 SFs and so this was selected as the value of *M*.

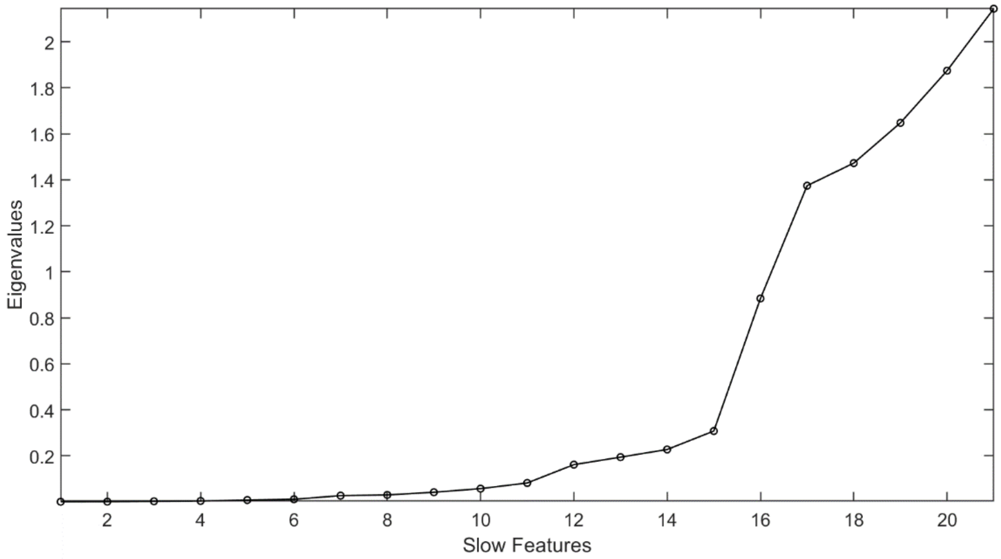

Figure 6.9 – Scree plot for SFA on normal operation data, used for dominant slow feature selection.

Figure 6.10 shows the T^2 and SPE charts for linear SFA applied to the normal operating data and blockage 1 data. There are a few periods of normal operation where the SPE breaches the control limit, albeit for only single data points and not accompanied by a T^2 either, and so it would not class as a blockage detected.

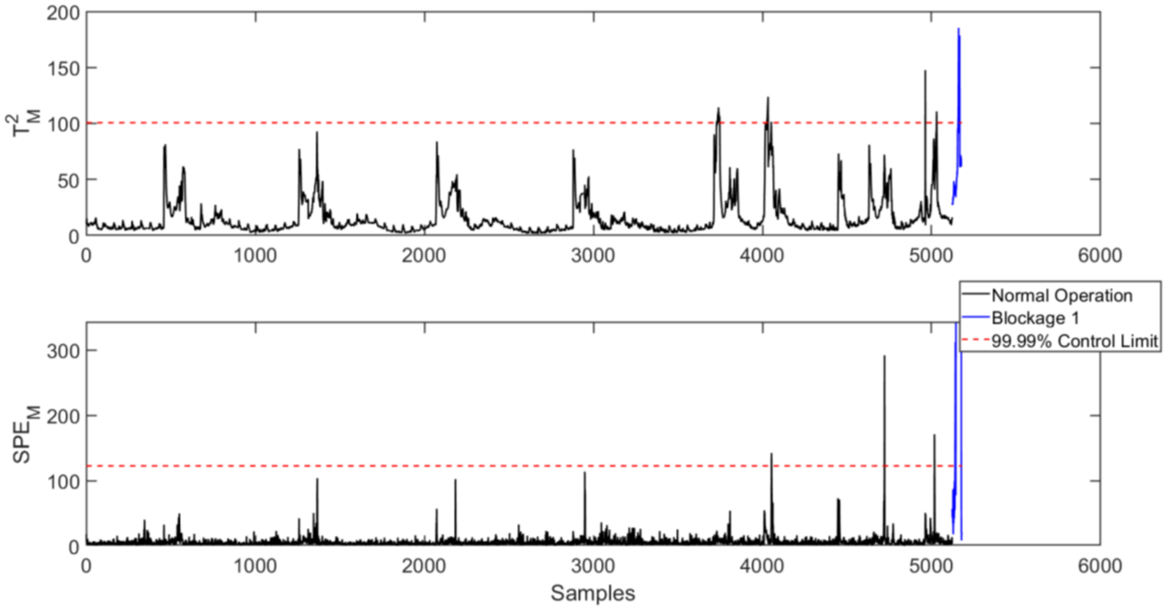

Figure 6.10 – T^2 (top) and SPE (bottom) control charts for the SFA model for blockage 1

Looking at the relative detection times, as displayed in Table 6.3, every blockage was detected earlier on the SPE chart than on site. Additionally, the average relative time was 58.1% better than PCA, and slightly better (2.6%) than DPCA. This improvement in detection time is due to the underlying trends extracted in the dominant SFs and the lower amounts of noise associated with them. Deviations from the slow, time-varying trends in typical, normal operation are more apparent than the deviations from the static variance captured in the principal components, which also contain more noise than the SFs. Reducing the effect of noise makes the control chart more sensitive to blockages, improving the relative detection time.

Table 6.3 – Relative detection times for each blockage using an SFA-based control chart

	Relative detection time (min)
Blockage 1	-13
Blockage 2	-5
Blockage 3	-18
Blockage 4	-11
Blockage 5	-20
Blockage 6	-4
Blockage 7	-8
Mean	-11.29

6.7 Dynamic Slow Feature Analysis-based MSPC

As with DPCA, dynamic SFA (DSFA) can capture temporal correlations and provide further information in addition to extracting the underlying trends and reducing noise. DSFA creates the input matrix with time lagged inputs in the same way as Equation (4.12). This input matrix is then input into SFA, and the monitoring statistics are calculated as in Section 6.7. The value of *d* was selected in the same way as DPCA, which was found to be 3.

The control chart for the first blockage using DSFA is displayed in Figure 6.11. A few 2+ sample breaches of the control limits occurred across both charts, however, no breaches occurred on both charts simultaneously and so there was no false alarms for DSFA.

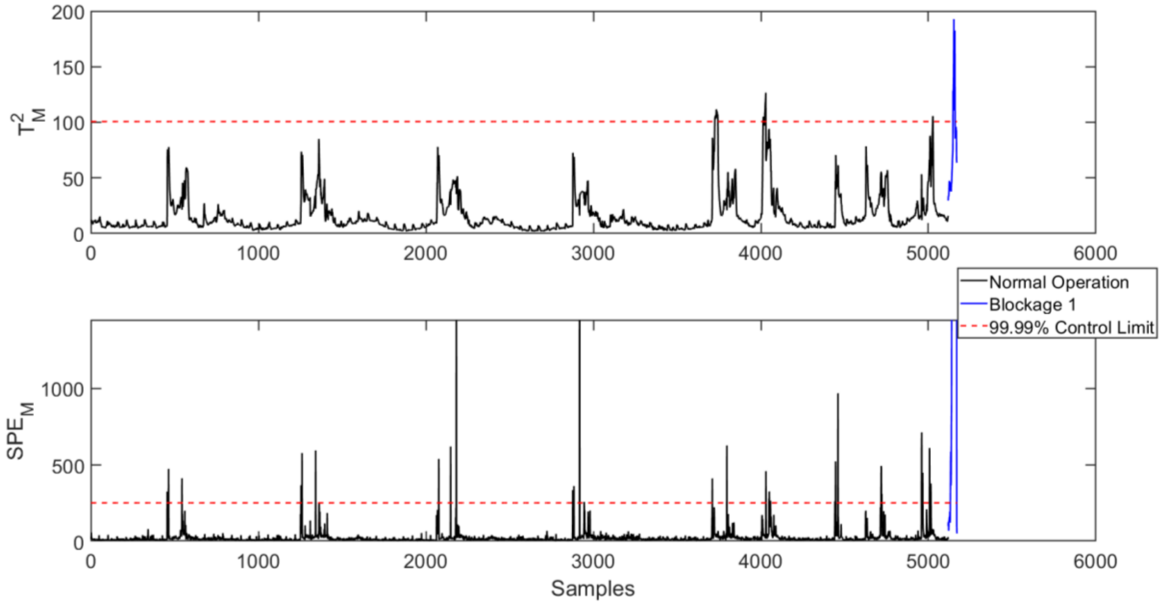

Figure 6.11 – T^2 (top) and SPE (bottom) control charts for the DSFA model (d = 3) for blockage 1

The average relative detection time was 29.1% better than SFA, demonstrating that the extra dynamic information further improved early blockage detection. The average relative detection time for each method is summarised in Table 6.5.

Table 6.4 – Relative detection times for each blockage using an DSFA-based control chart (d = 3)

	Relative detection time (min)
Blockage 1	-21
Blockage 2	-13
Blockage 3	-27
Blockage 4	-4
Blockage 5	-12
Blockage 6	-20
Blockage 7	-13
Mean	-14.57

Table 6.5 – Mean relative detection times for each method

Method	Mean Relative Detection Time (minutes)
PCA	-4.43
Dynamic PCA	-11
Linear SFA	-11.29
Dynamic SFA	-14.57

6.8 Contribution Plots

The contributions of variables to the blockages can indicate whether there are any variables which could be causing the blockages. This could give important insight for process operations to help reduce the chances of blockages in the future.

A residual contribution plot, based on the PCA model, showing 3 samples in a short time leading up to the blockage and 1 sample after the blockage was produced to assess whether any early warning from a contributing process variable could be found. The residual contribution plot assesses the contribution of the process variables to the SPE statistic. This is done by plotting the residual of each variable, for each of the 4 samples described above. The residual E is calculated as part of the SPE, as described in Equation (6.4). Figure 6.12 illustrates these residual contributions and shows that variables 18 and 20 contribute the most. This is expected because these variables directly correlate to blockages in the DS and are used as an indicator of a blockage already.

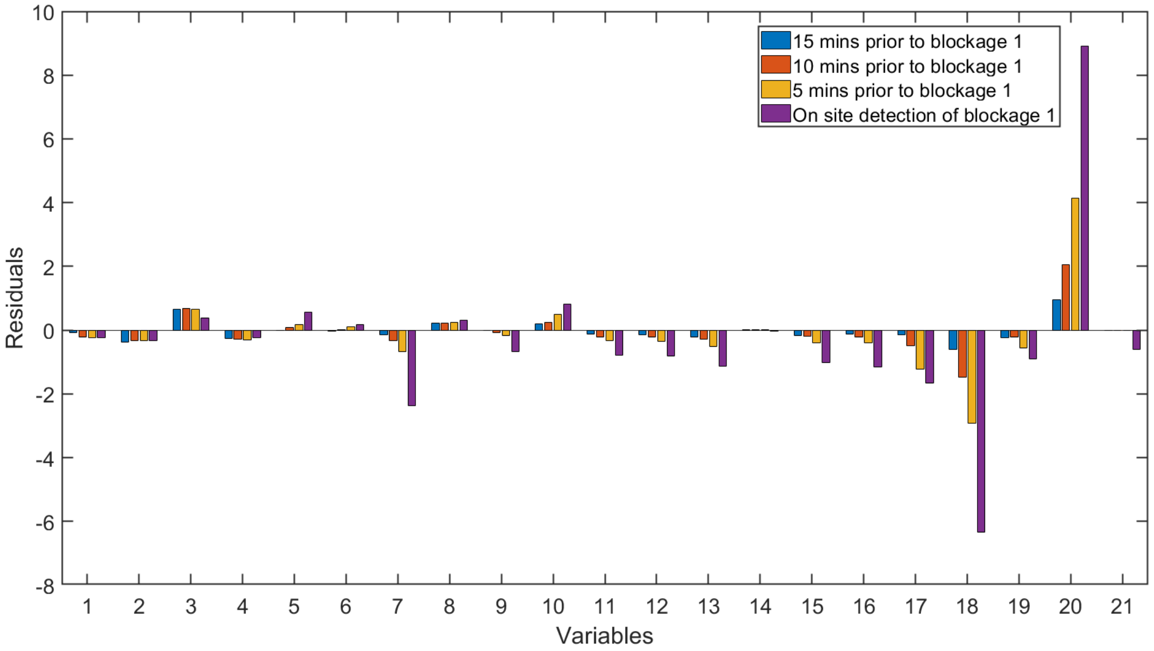

Figure 6.12 – Residual Contribution plot, using PCA, for blockage 1 for blockage 1 at certain time points leading up to blockage detection

Inspecting the DPCA, SFA and DSFA residual contribution plots (Figures 6.13 to 6.15) may reveal more information since these models provided better early blockage detection. The DPCA contribution plot shows a slightly bigger contribution from variable 7 prior to, and at detection of, the blockage. Since the blockage occurs in the DS, the DS temperature may have an impact on the blockage build up. The SFA and DSFA contribution plots confirm the contributions from variables 18 and 20.

The high residuals in DSFA demonstrate that this model is perhaps too sensitive to small deviations in the process, as also shown by the breaches in the T^2 and SPE charts on normal operation data.

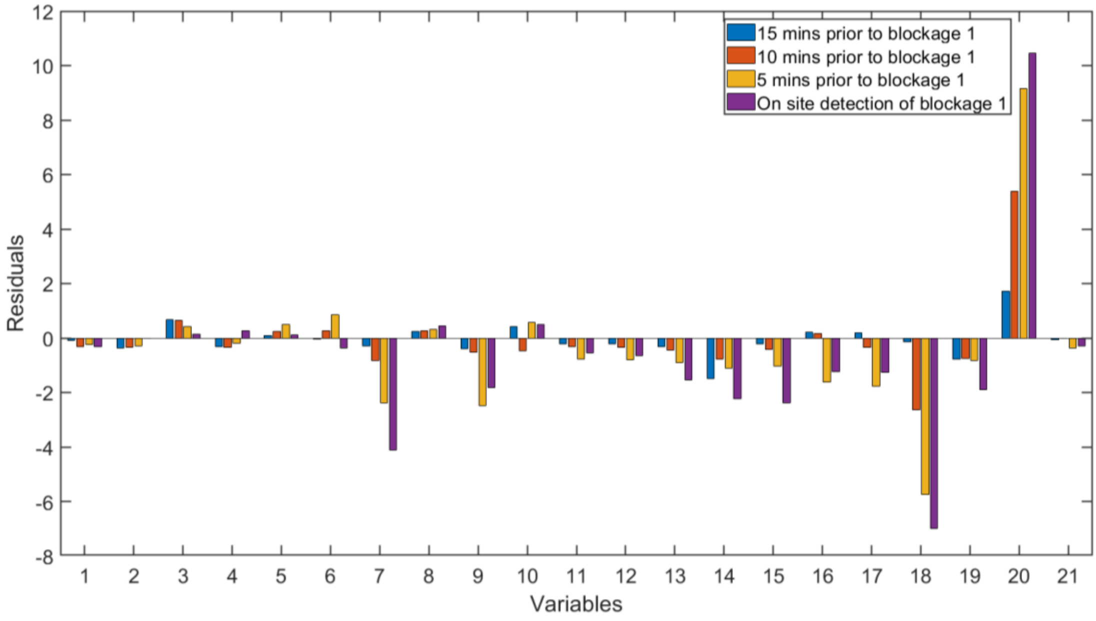

Figure 6.13 – Residual Contribution plot, using DPCA, for blockage 1 at certain time points leading up to blockage detection

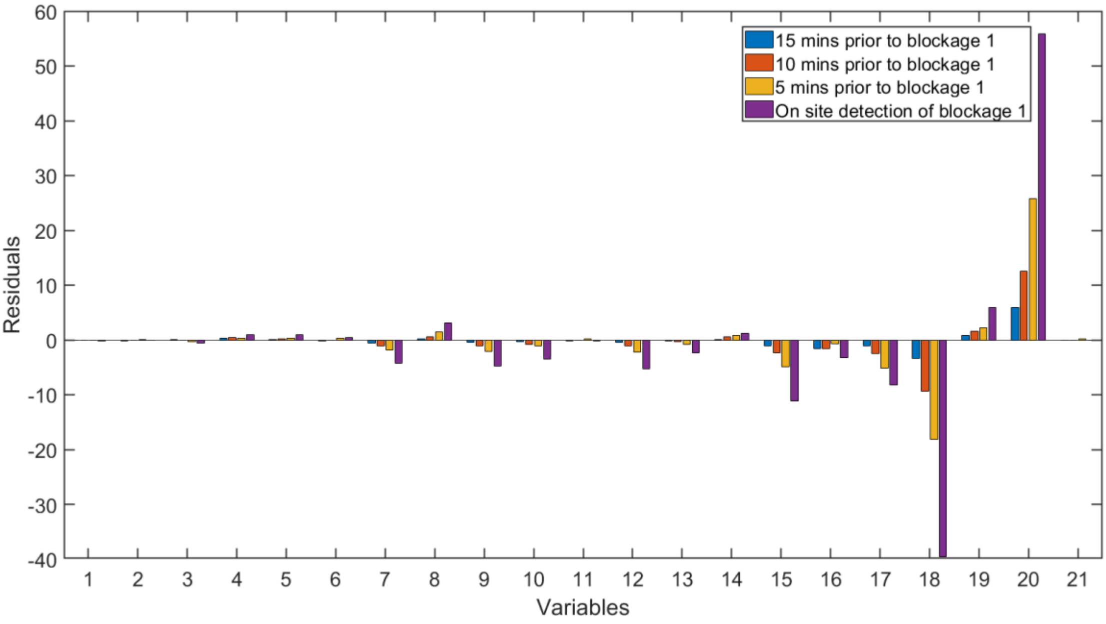

Figure 6.14 – Residual Contribution plot, using SFA, for blockage 1 at certain time points leading up to blockage detection

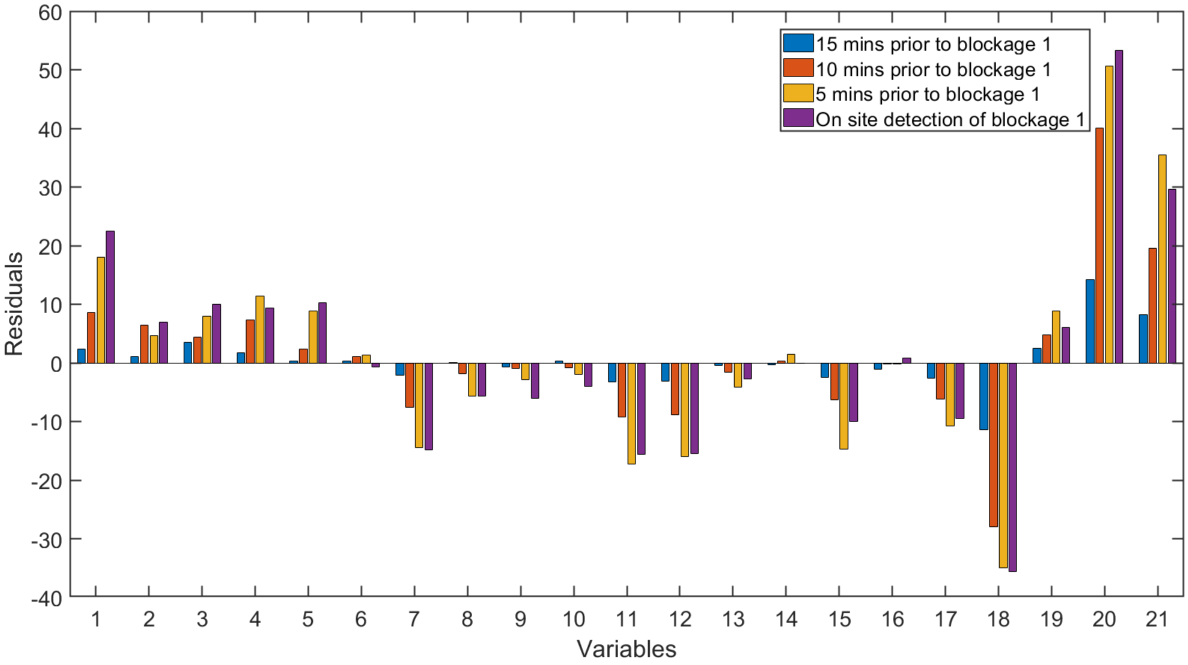

Figure 6.15 – Residual Contribution plot, using DSFA, for blockage 1 at certain time points leading up to blockage detection

Appendix A shows the residual contribution plots for the other blockages using the DSFA model (since it was selected as the best model). The contributions are similar for the other blockages, except for blockages 3 and 6, where there are no clear, large contributions for any variable that are maintained for all 4 time points. Variables 15 and 17 also show some higher contributions across most of the blockages and across all 4 time points. Through discussions with Sellafield Ltd, these variables are understood to affect a blockage but are not indicative of one.

6.9 Conclusions

Principal component analysis (PCA) and slow feature analysis (SFA) were used on the process data from Sellafield Ltd to investigate if their implementation within a multivariate statistical process control approach could lead to early warning of the blockages in the dust scrubber system, as well as detecting changes in process variables that are causing these blockages.

PCA yielded a small amount of early warning of the blockages. Dynamic PCA was tested to assess if any variables were correlated by a time lag and that if including these time lagged variables would improve the model. In terms of early detection, dynamic PCA with a time lag of 3 for all the variables improved the average detection time of the blockages by 59.7%

compared with PCA. Additionally, there were no false alarms with this dynamic PCA control chart.

Slow feature analysis was utilised because it can capture the slow varying features in the data, which are advantageous for process monitoring over PCA due to the noise and variation in many of the process variables, since SFA reduces the impact of noise through retaining the slowest of the total SFs. SFA and DSFA significantly improve the early detection of blockages over PCA and DPCA, with DSFA improving average detection time by 29.1% when compared with the next best, SFA. With no false alarms and the best average detection times, DSFA was the best technique.

Residual contribution plots were created to determine whether any process variables are the causes of the blockages, however, the main contributions came from the dust scrubber pressures, which is already known to be the main indicator of a blockage, but not a cause. Other significant contributions were likely to be consequences of the blockages and not causes.

Chapter 7. Nonlinear Data-Driven Modelling of Chemical Durability of High-Level Nuclear Waste Glass

7.1 Introduction

The latter stages of the nuclear waste vitrification process at Sellafield Ltd relate to high-level glass waste (HLW) being poured into containers that are sealed, decontaminated, and then eventually stored in a deep geological repository.

One of the key properties of the HLW glass is the chemical durability. Chemical durability is defined as the resistance of the waste to release elements into aqueous solutions, such as groundwater. Understanding this durability behaviour is important because it determines the necessary engineered barriers to prevent radionuclide release into the environment when storing the HLW in a deep geological repository. The glass is expected to remain radioactive for many thousands of years and so the radioactive containers may eventually erode, and the glass come into contact with groundwater. The chemical durability is quantified by the leach rate, typically in the form of normalised mass loss, of specific elements. The modelling of chemical durability can lead to optimal compositions of glass that can minimise the normalised mass loss of the radioactive elements.

Modelling of the chemical durability of the glass is therefore significant for nuclear waste management. Attempts have been made to model the chemical durability from first principles (Frugier *et al.*, 2009), however, inaccuracies have been shown when applied to experimental data.

Data-driven modelling can provide improvements over mechanistic modelling without the need for extensive knowledge and potentially in a less time and effort demanding way. Nonlinear data-driven techniques, such as neural networks (Bishop, 1995) and extreme learning machine (Huang *et al.*, 2006), can model complex nonlinear relationships that linear techniques may be inadequate for. Using machine learning to predict static and dynamic glass leaching behaviour has been done previously (Lillington *et al.*, 2020).

Experiments to understand the durability of HLW have been carried out to assess the effect of waste loading (Brookes *et al.*, 2010) and composition of glass (Harrison and Scales, 2008; Harrison, 2014; Harrison and Brown, 2018) on the chemical durability.

This chapter uses bootstrap aggregated extreme learning machine to model the chemical durability of glass waste based on experimental data of accelerated static leach rate. A sensitivity analysis is then carried out to determine if the chemical durability can be optimised by varying the 4 main glass forming additives.

7.2 Overview of Extreme Learning Machine

Extreme learning machine (ELM) is essentially a single hidden-layer feedforward neural network, as illustrated in Figure 7.1. The difference from a traditional neural network arises from the determination of the connection weights. In a neural network, the input and output weights are typically optimised by a backpropagation algorithm, however, for ELM the input weights are randomised and so the output weights can be computed more simply.

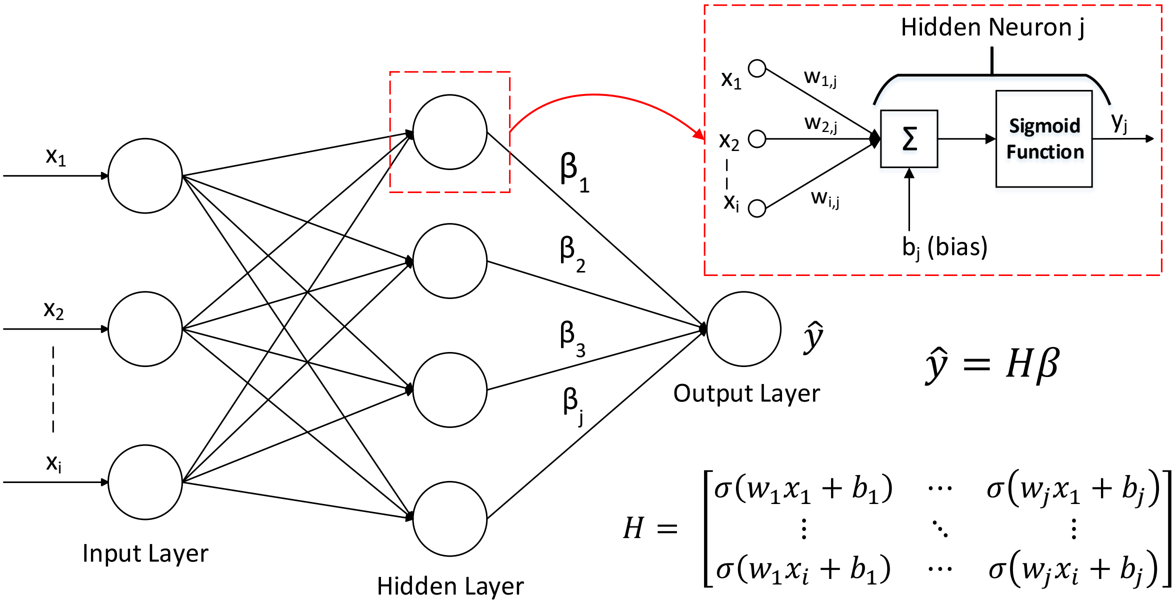

Figure 7.1 – A diagram of extreme learning machine

Given a training set of *N* different samples, where the inputs are defined by Equation (7.1) and the target outputs by Equation (7.2):

$$X = [x_1, x_2, \dots x_n]^T\ , \tag{7.1}$$

$$Y = [y_1, y_2, \dots y_m]^T\ , \tag{7.2}$$

the output of a SLFFN, *y*, can be defined as:

$$y_j = \sum_{i=1}^{NH} \beta_i g\left(w_i x_j + b_i\right) \qquad j = 1\ to\ N\ , \tag{7.3}$$

where N_H is the number of hidden neurons, w_i is the weight vector connecting the input neurons to the *i*th hidden neuron, b_i is the bias of the *i*th hidden neuron, g is the hidden layer activation function, and β_i is the output weight vector connecting the *i*th hidden neuron to the output neurons.

In standard SLFFN, the error function is defined, and gradient descent methods are used to iteratively determine the optimal input and output weights to minimise the error function. This error function, e, is typically the mean squared error (MSE),

$$e = \frac{1}{N} \sum_{j=1}^{N} \left(y_j - t_j\right)^2 . \tag{7.4}$$

In ELM, the input weights and bias are randomised and so, for compactness, the output of the hidden neurons can be written as a matrix *H*, such that $\hat{y} = H\beta$ (Huang *et al.*, 2006), where

$$H = \begin{bmatrix} g(w_1 x_1 + b_1) & \cdots & g(w_L x_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(w_1 x_N + b_1) & \cdots & g(w_L x_N + b_L) \end{bmatrix} , \tag{7.5}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix} . \tag{7.6}$$

In order to find the optimal β matrix that minimises the error function, the error function is differentiated with respect β and set equal to zero. Firstly, the error function is written in matrix form as:

$$e = \frac{1}{2} \|Y - T\|^2 . \tag{7.7}$$

Using the result $Y = H\beta$, Equation (7.7) can be re-written as

$$e = \frac{1}{2}\|H\beta - T\|^2 = \frac{1}{2}(H\beta - T)^T(H\beta - T) \tag{7.8}$$

and then differentiated with respect to β to give

$$\frac{\partial e}{\partial \beta} = H^T H\beta - H^T T\ . \tag{7.9}$$

Setting Equation (7.9) equal to zero and solving for β gives the solution:

$$\beta = H^{\dagger} T\ , \tag{7.10}$$

where $H^{\dagger}$ is the Moore-Penrose generalised inverse of matrix H. This gives the minimum norm least squares solution of the linear system $H\beta = T$ (Huang *et al.*, 2006).

7.3 Bootstrap Aggregated Extreme Learning Machine

Bootstrap aggregating, also known as bagging, is an ensemble method that was first proposed by Breiman (1996) as a way to improve generalisation performance through the combination of multiple predictors. Bootstrap aggregated neural networks have been investigated previously for process modelling and have been shown to improve model robustness and reliability (Zhang, 1999; Zhang *et al.*, 2006; Murkherjee and Zhang, 2008). Bootstrap aggregated models have also been described in Chapters 2 and 5.

In this work, bootstrapped aggregated extreme learning machine (BA-ELM) is used to increase model generalisation performance and reliability over single ELM models.

The idea behind BA-ELM is to create multiple models which then have their individual predictions combined to produce the final model output, as illustrated in Figure 7.2.

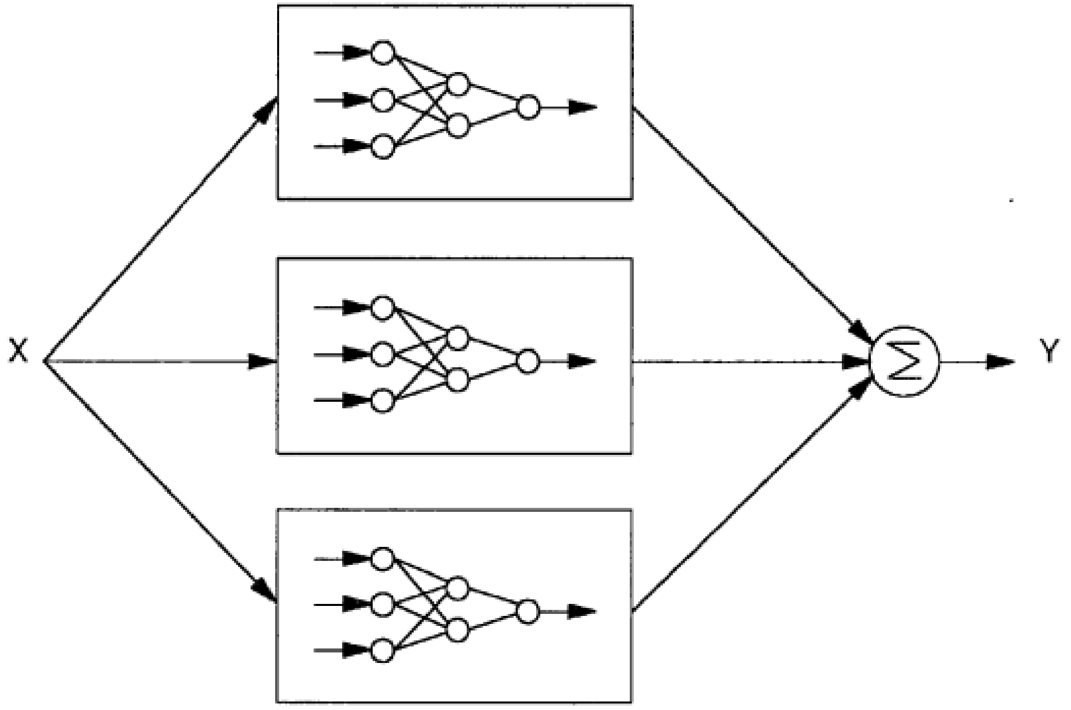

Figure 7.2 – A simple diagram of a bootstrap aggregated extreme learning machine (Zhang, 1999).

To improve accuracy through model combination, the individual models must be diverse. This diversity is achieved through bootstrap resampling of the training data, which is essentially sampling with replacement (Bland and Altman, 2015). The simplest way to combine the predictions from the individual models is through averaging, which was the method adopted in this work. Methods using weighted averaged, such as PCR, have also been investigated (Zhang, 1999).

7.4 Modelling of Chemical Durability

Static leach rate accelerated experiments have been carried out by the National Nuclear Laboratory (NNL) to produce data of the normalised mass loss against time of 8 key elements for different feed compositions (Harrison, 2014). The experiments involve a powdered glass submerged in a test solution, typically deionised water or test groundwater, at a fixed temperature to investigate glass dissolution up to saturation. The powdered glass provides a high surface area to volume ratio so that saturation conditions can be reached quicker. Investigating normalised mass loss to saturation provides information on the long term corrosion behaviour of the glass. Since experiments measuring the leach rate for hundreds or thousands of years are not possible, these accelerated experiments, and creating models with them, are the best way to understand the long-term leach rate currently.

This experimental data from the static leach rate experiments covered 26 different glass compositions that varied based on the type of waste, blend ratio of waste, and amount of glass forming additives. The glass compositions comprised of 23 different oxides, including the 4 main glass forming additives of boron oxide, lithium oxide, sodium oxide and silicon dioxide. For each composition, the data included the normalised mass loss of 8 key elements for 9 time points between 7 and 182 days. An example of the raw data is displayed in Figure 7.3. Although the chemical durability can be affected by other factors, the focus of this data and this work is on the effect of different glass compositions on the chemical durability.

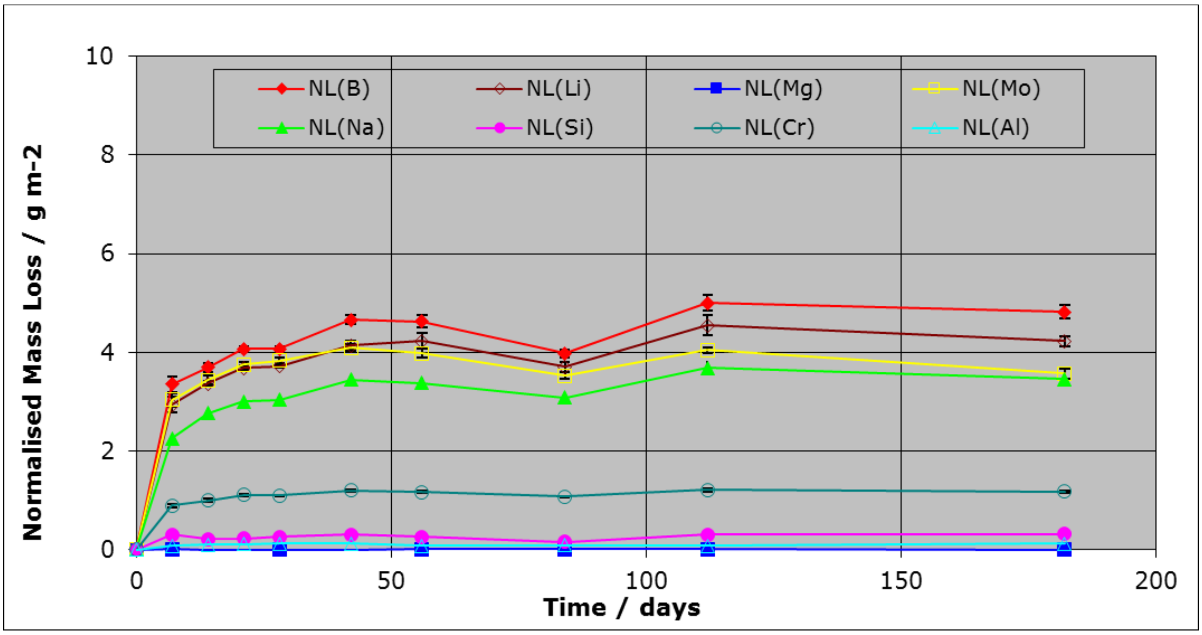

Figure 7.3 – Example of the data obtained from one static leach rate experiment given a certain glass composition.

7.4.1 Data pre-processing

Outliers were detected by visual inspection and also with the “3σ” rule (as described in Section 2.1.1). For example, the normalised mass loss for Chromium with 3σ limits is displayed in Figure 7.4, showing that one outlier is detected with this method. For this data, the outliers were removed due to the batch nature of the data meaning that replacement is difficult. Additionally, the outliers were few and so did not impact the amount of data very much. Replacement is a higher priority if a higher proportion of the data is outliers and so removal would lead to a reduction in data for model building.

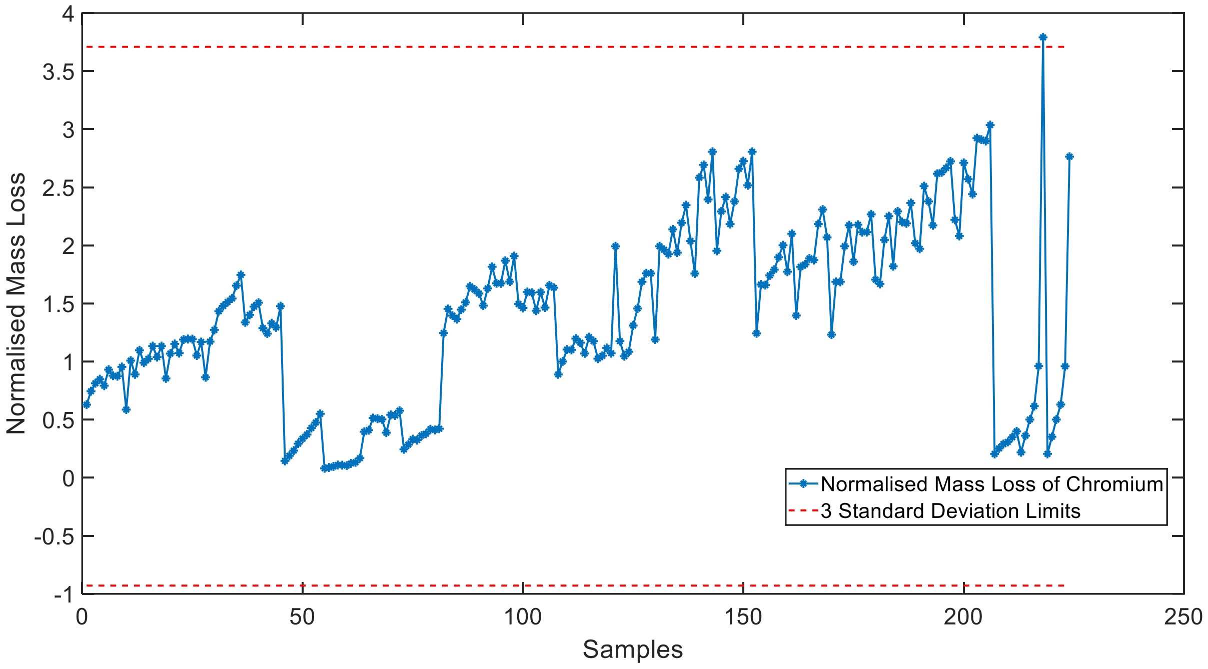

Figure 7.4 – Outlier detection using the 3σ rule for the normalised mass loss of Chromium

Although the end time points are of most importance, because this is the region of saturation, all the data is utilised in the model building because the amount of data points is so small to begin with. This also allows for predictions to be made for different stages of leaching over time. This means that there is a mixture of time varying data and once-per-experiment data. To include each time point into the model, the time was used as a model input along with the 23 glass compositions. This means that the composition of each glass was repeated for all the time points associated with the normalised mass losses.

This gave an input matrix of the following structure:

$$\begin{bmatrix} time_{1,1} & Glass\ Composition_1 \\ \vdots & \vdots \\ time_{1,T} & Glass\ Composition_1 \\ time_{2,1} & Glass\ Composition_2 \\ \vdots & \vdots \\ time_{j,T} & Glass\ Composition_j \end{bmatrix} \tag{7.11}$$

where T is the number of time samples from an experiment, j is the number of glass compositions, and "$Glass\ Composition_j$" represents a horizontal vector of the 23 different oxide compositions for the j^{th} glass composition.

The elements making up the compositions of each glass type are listed in Table 7.1.

Table 7.1 – List of the 23 elements present in each glass

Elements in Glass
Al_2O_3
B_2O_3
BaO
CeO_2
Cr_2O_3
Cs_2O
Fe_2O_3
Gd_2O_3
La_2O_3
Li_2O
MgO
MoO_3
Na_2O
Nd_2O_3
NiO
Pr_2O_3
RuO_2
SiO_2
Sm_2O_3
SrO
TeO_2
Y_2O_3
ZrO_2

The output matrix is simply the normalised mass loss (NML) of the 8 target elements, for each timepoint, for each glass type, as described below:

$$\begin{bmatrix} {NML_1}^1 \\ \vdots \\ {NML_T}^1 \\ \vdots \\ {NML_1}^J \\ \vdots \\ {NML_T}^J \end{bmatrix} \tag{7.12}$$

7.4.2 Modelling using bootstrap aggregated extreme learning machine

A separate model was built for each of the 8 elements, using BA-ELM, so that the highest prediction accuracy can be obtained for each individual model. This is because creating a single multiple-output model leads to a compromise in the performance of each output. Creating individual models allows each model to maximise performance for that specific element.

The data was split into training (60%), testing (20%) and unseen validation (20%) data sets overall. However, initially the data was split in a training plus testing set (TT) and a validation set so that bootstrapping could be performed on the whole TT data set to increase diversity in the individual models. Then the TT is split into training and testing following the bootstrapping.

The number of hidden neurons was determined by cross validation with the testing data RMSE for each model in the ensemble individually. To determine the number of ELM networks, a plot of root mean squared error (RMSE) on testing data against the cumulative number of networks in the model was plotted. An example of this plot is illustrated in Figure 7.5. The error levels off as more networks are included into the BA-ELM model and the number of networks was selected by when the variation in testing RMSE from adding in additional networks becomes insignificant (Zhang, 2004; Murkherjee and Zhang, 2008; Mohammed and Zhang, 2013).

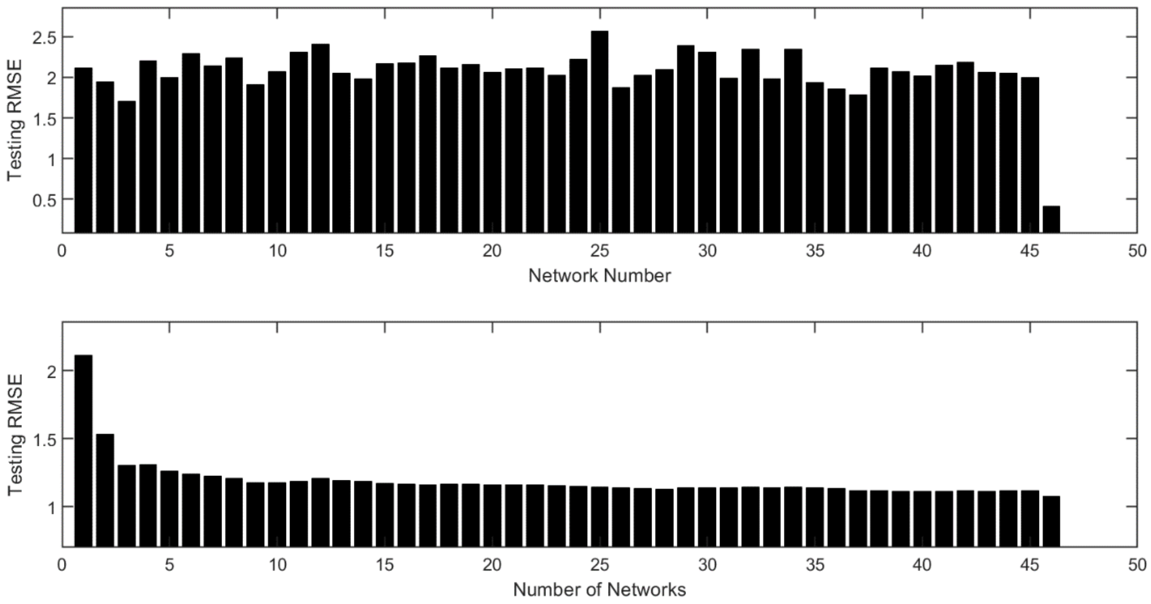

Figure 7.5 – Testing RMSE for each individual ELM network for the boron model (top). Testing RMSE for BA-ELM models for different numbers of aggregated networks for the boron model (bottom).

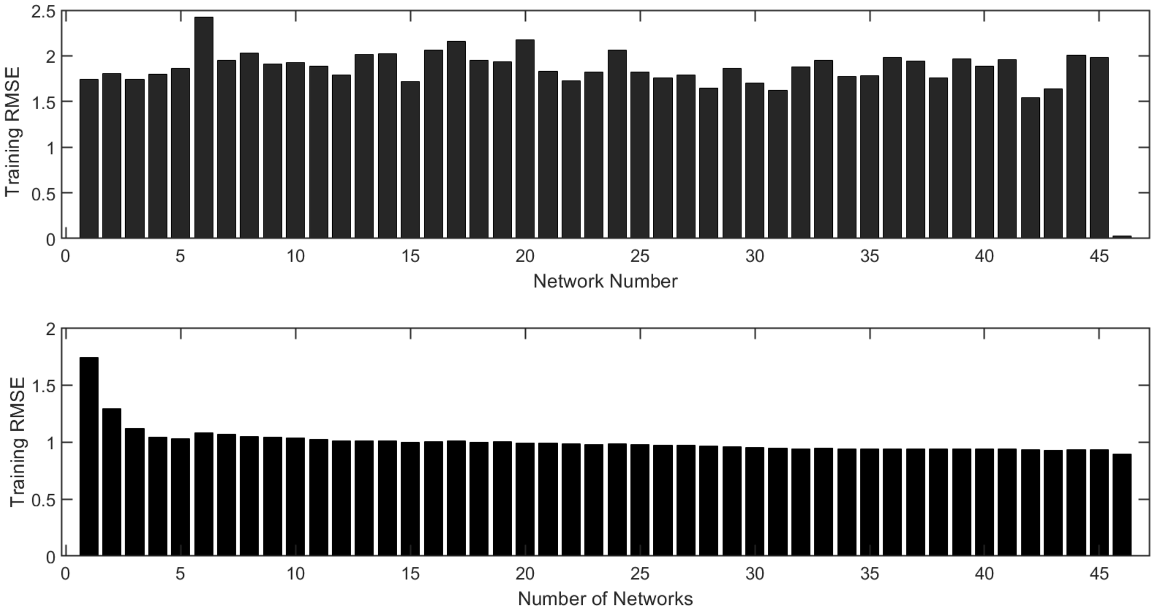

Figure 7.6 – Training RMSE for each individual ELM network for the boron model (top). Training RMSE for BA-ELM models for different numbers of aggregated networks, for the boron model (bottom).

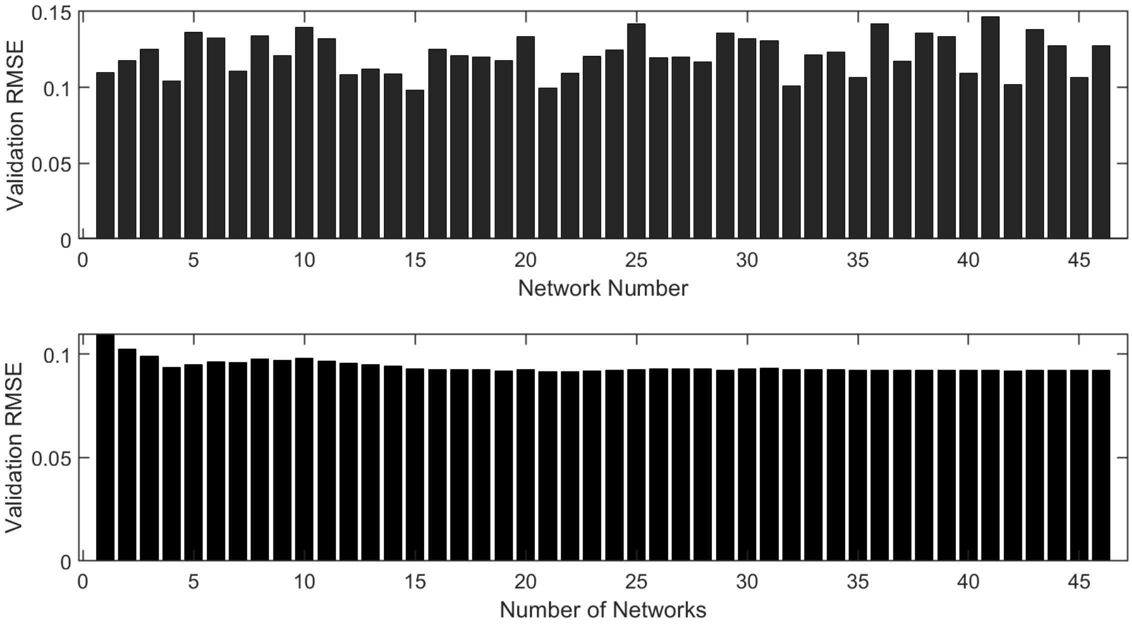

Figure 7.7 – Validation RMSE for each individual ELM network for the boron model (top). Validation RMSE for BA-ELM models for different numbers of aggregated networks, for the boron model (bottom).

Comparing the training RMSE and testing RMSE for the individual networks highlights the inconsistency in single ELM models due to some networks performing well on training data but poorly on testing data. This is also confirmed with the validation data, as shown in Figure 7.7. On the other hand, the consistency in RMSE for BA-ELM models shows improved reliability when compared to single ELM.

Table 7.2 – Single ELM performance for each data set for each element's model.

	Training		**Testing**		**Validation**	
Element	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**
Aluminium	0.9495	0.2247	0.9459	0.2321	0.9587	0.2129
Boron	0.9439	0.2334	0.9132	0.2980	0.8872	0.3314
Chromium	0.9737	0.1640	0.9458	0.2297	0.9540	0.2133
Lithium	0.9485	0.2284	0.9268	0.2683	0.9028	0.3133
Magnesium	0.7004	0.4385	0.6817	0.6547	0.5976	0.6490
Molybdenum	0.9696	0.1555	0.9613	0.2154	0.9251	0.2111
Sodium	0.9416	0.2378	0.9188	0.2886	0.8907	0.3262
Silicon	0.8320	0.4046	0.7493	0.5060	0.7028	0.5461

Table 7.2 and Table 7.3 show the performance of single ELM and BA-ELM models respectively, showing both R^2 and RMSE performance metrics. The elements with validation RMSE values highlighted in bold indicate where the generalisation performance was best for that element model when comparing the two methods. It can be seen that all eight difference models had better validation performance for BA-ELM than single ELM. The magnesium model performance is poor due to quality of data not being as good and so more data, and potentially better pre-processing, would be required to improve this model.

Table 7.4 provides further detail on this comparison by showing the percentage difference for each model for both testing RMSE and validation RMSE, including the average for each element model. A negative percentage indicates the RMSE is lower for BA-ELM, i.e. BA-ELM is better by that negative percentage. The averages for both testing and validation RMSE demonstrate that the improvement in generalisation performance for BA-ELM, compared with Single ELM, is significant. There is also then the added benefit of the reliability improvements provided by creating a diverse ensemble, as described earlier.

Table 7.3 – BA-ELM performance for each data set for each element's model.

	Training		**Testing**		**Validation**		
Element	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**#Nets**
Aluminium	0.9473	0.2295	0.9591	0.2019	0.9596	**0.2103**	40
Boron	0.9500	0.2205	0.9449	0.2374	0.9055	**0.3038**	50
Chromium	0.9663	0.1857	0.9797	0.1405	0.9627	**0.1919**	45
Lithium	0.9532	0.2176	0.9567	0.2062	0.9198	**0.2845**	46
Magnesium	0.6432	0.4785	0.7750	0.5504	0.6707	**0.5870**	50
Molybdenum	0.9617	0.1745	0.9802	0.1542	0.9332	**0.1994**	40
Sodium	0.9512	0.2948	0.9470	0.2333	0.9108	**0.2948**	45
Silicon	0.8388	0.3963	0.8101	0.4404	0.7365	**0.5142**	46

Table 7.4 – Percentage difference between BA-ELM and Single ELM models

	Percentage Difference (%)	
Element	**Testing RMSE**	**Validation RMSE**
Aluminium	-13.0	-1.2
Boron	-20.3	-8.3
Chromium	-38.8	-10.0
Lithium	-23.1	-9.2
Magnesium	-15.9	-9.6
Molybdenum	-28.4	-5.5
Sodium	-19.2	-9.6
Silicon	-13.0	-5.8
Average	-21.5	-7.4

7.4.3 Principal component analysis-based method results

To further improve on the BA-ELM results, PCA was added as a pre-processing step to reduce collinearity and dimensionality. The use of PCA in data-driven modelling has been detailed in Sections 2.4.1, 5.3.1 and 6.4.

The retained principal components *k* was determined by the principal components that captured at least 99% of the variance of the data. The *k* principal components were used as the inputs to the BA-ELM models. As shown by Figure 7.8, 7 principal components captured over 99% of the variance in the data and so this was the value of *k*. Since the input data is the same for each element model, *k* was 7 for all models.

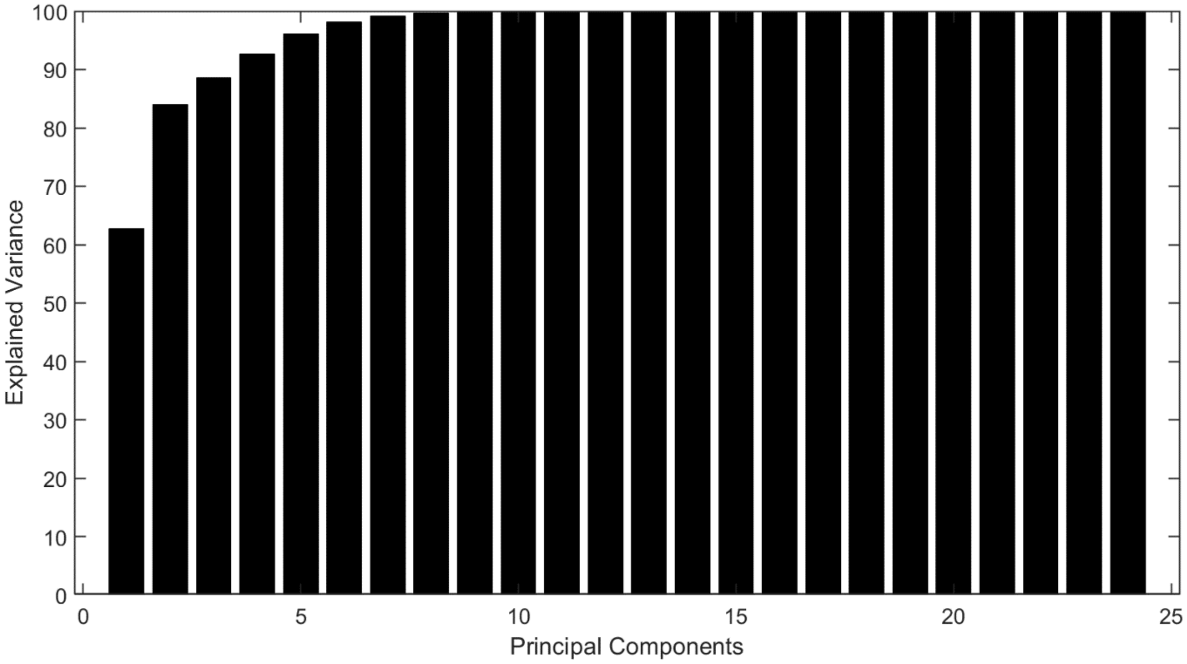

Figure 7.8 – Cumulative explained variance for PCA applied to the input data

The results of incorporating PCA into BA-ELM (PCA-BA-ELM) are shown in Table 7.5. Table 7.6 shows the percentage difference between PCA-BA-ELM and BA-ELM for each element. These results show a significant performance increase of 10.4% on average across all element models when comparing PCA-BA-ELM with BA-ELM. This improvement can be attributed to the ability of PCA to capture most of the variance in less features (leading to fewer inputs that have little meaningful information or even misleading information) and to reduce co-linearity, to ensure the data is information rich.

Table 7.5 – PCA-BA-ELM performance for each data set for each element's model

	Training		Testing		Validation		
Element	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**#Nets**
Aluminium	0.9540	0.2143	0.9665	0.1827	0.9648	0.1965	50
Boron	0.9566	0.2053	0.9554	0.2134	0.9181	0.2824	50
Chromium	0.9685	0.1795	0.9825	0.1306	0.9646	0.1872	40
Lithium	0.9576	0.2073	0.9635	0.1894	0.9282	0.2692	45
Magnesium	0.7251	0.4200	0.7146	0.5046	0.7146	0.5466	40
Molybdenum	0.9627	0.1722	0.9769	0.1664	0.9364	0.1945	40
Sodium	0.9590	0.1991	0.9593	0.2044	0.9262	0.2681	35
Silicon	0.9388	0.2557	0.9250	0.2636	0.8434	0.3668	40

Table 7.6 – Percentage difference between PCA-BA-ELM and BA-ELM models

	Percentage Difference (%)	
Element	**Testing RMSE**	**Validation RMSE**
Aluminium	-10.5	-7.0
Boron	-11.2	-7.6
Chromium	-7.6	-2.5
Lithium	-8.9	-5.7
Magnesium	-9.1	-7.4
Molybdenum	7.3	-2.5
Sodium	-14.1	-10.0
Silicon	-67.1	-40.2
Average	-15.1	-10.4

To assess the impact of using ELM over the very similar SLFNN, models were created using PCA as the precursor to a bagged neural network model (PCA-BA-NN). The neural networks were trained using the Levenberg-Marquart algorithm and 0.001 regularisation (same regularisation as for ELM for a better comparison). The results using PCA-BA-NN are displayed in Table 7.7 and, again, the percentage difference between PCA-BA-NN and PCA-BA-ELM is shown in Table 7.8. For every element on both testing and validation data, the PCA-BA-ELM model was best, as shown by the positive percentages in Table 7.8. On average, PCA-BA-ELM had a generalisation performance 11.6% better than PCA-BA-NN, which is significant. These results demonstrate that ELM improves generalisation performance over a neural network while also reducing computational effort by simply computing the output weights through the input weights being randomised.

Table 7.7 – PCA-BA-NN performance for each data set for each element's model

	Training		**Testing**		**Validation**		
Element	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**#Nets**
Aluminium	0.9459	0.2325	0.9584	0.2036	0.9593	0.2111	45
Boron	0.9480	0.2247	0.9438	0.2396	0.9026	0.3080	50
Chromium	0.9627	0.1952	0.9782	0.1456	0.9610	0.1964	35
Lithium	0.9518	0.2208	0.9548	0.2108	0.9197	0.2848	40
Magnesium	0.6425	0.4790	0.7757	0.5496	0.6554	0.6006	40
Molybdenum	0.9588	0.1809	0.9743	0.1754	0.9298	0.2043	40
Sodium	0.9490	0.2222	0.9455	0.2364	0.9111	0.2943	35
Silicon	0.8368	0.3988	0.8112	0.4391	0.7352	0.5155	50

Table 7.8 – Percentage difference between PCA-BA-NN and PCA-BA-ELM models

	Percentage Difference (%)	
Element	**Testing RMSE**	**Validation RMSE**
Aluminium	11.4	7.4
Boron	12.3	9.1
Chromium	11.5	4.9
Lithium	11.3	5.8
Magnesium	8.9	9.9
Molybdenum	5.4	5.0
Sodium	15.7	9.8
Silicon	66.6	40.5
Average	17.9	11.6

Figures 8 to 7 show the predictions of each element for the PCA-BA-ELM models. These figures confirm the good performance on unseen data demonstrated through the performance metrics. The magnesium model is much worse than the others, however, as seen by the plot of magnesium predictions (Figure 7.13), the scale of the magnesium NML is much lower than the other elements. This means that the contribution of magnesium is very low compared to the total and so the poor model performance does not have a significant impact on the total normalised mass loss, which is typically of the most interest because this determines how much radioactivity is leaching into the environment overall.

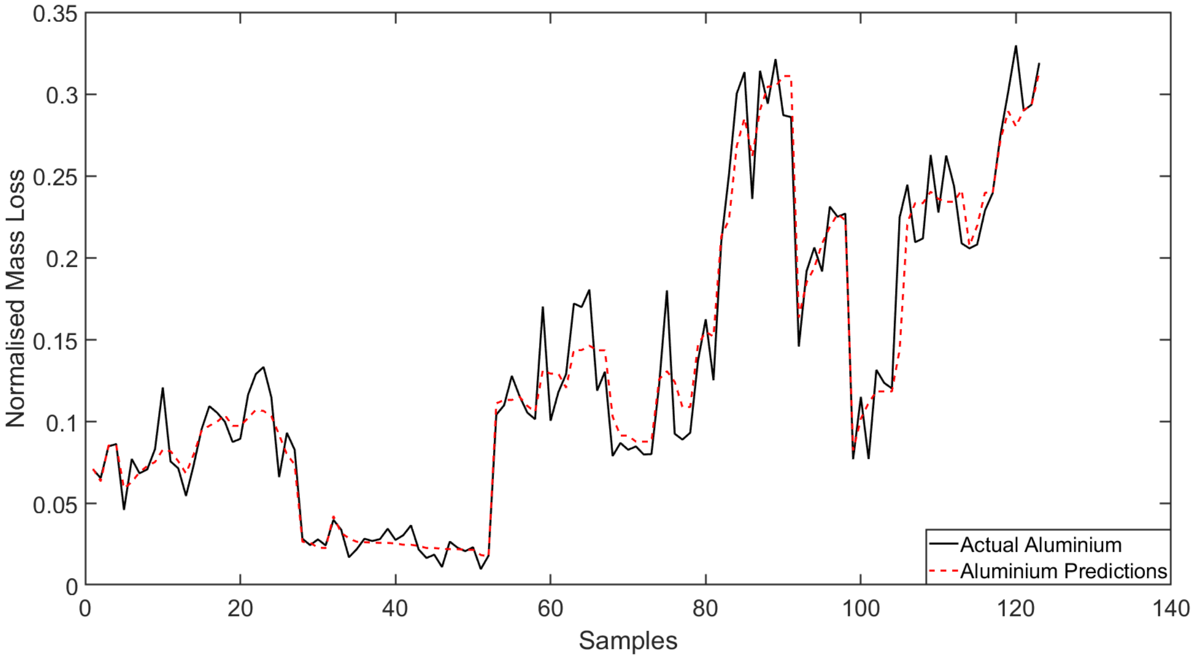

Figure 7.9 – Normalised mass loss predictions for aluminium on validation data, using the PCA-BA-ELM model

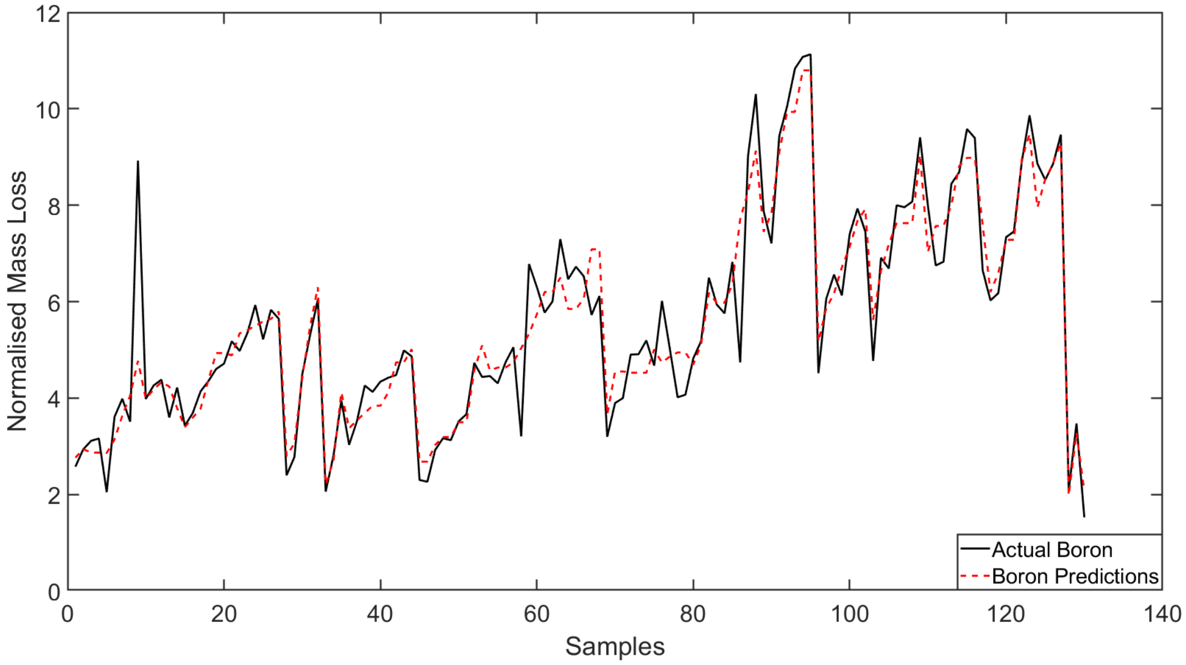

Figure 7.10 – Normalised mass loss predictions for boron on validation data, using the PCA-BA-ELM model

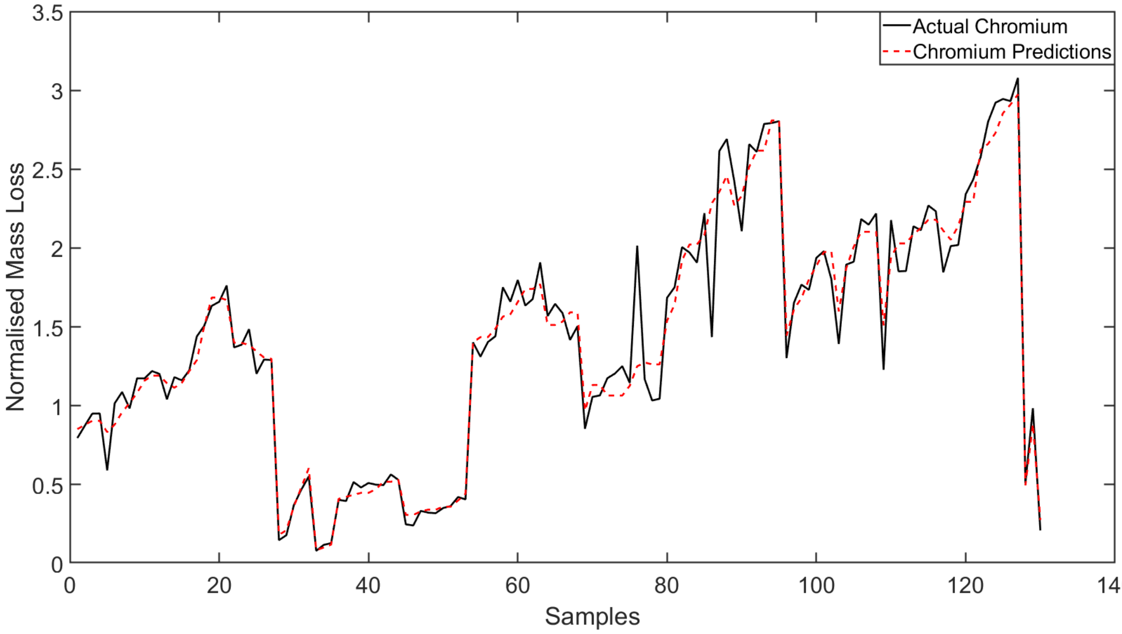

Figure 7.11 – Normalised mass loss predictions for chromium on validation data, using the PCA-BA-ELM model

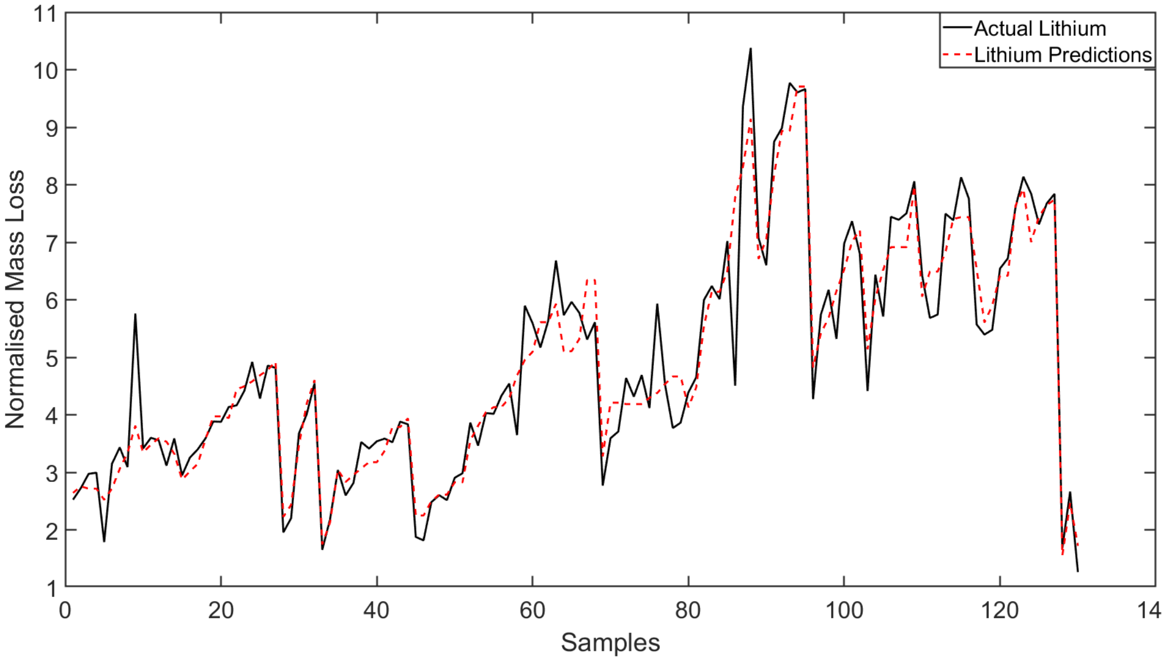

Figure 7.12 – Normalised mass loss predictions for lithium on validation data, using the PCA-BA-ELM model

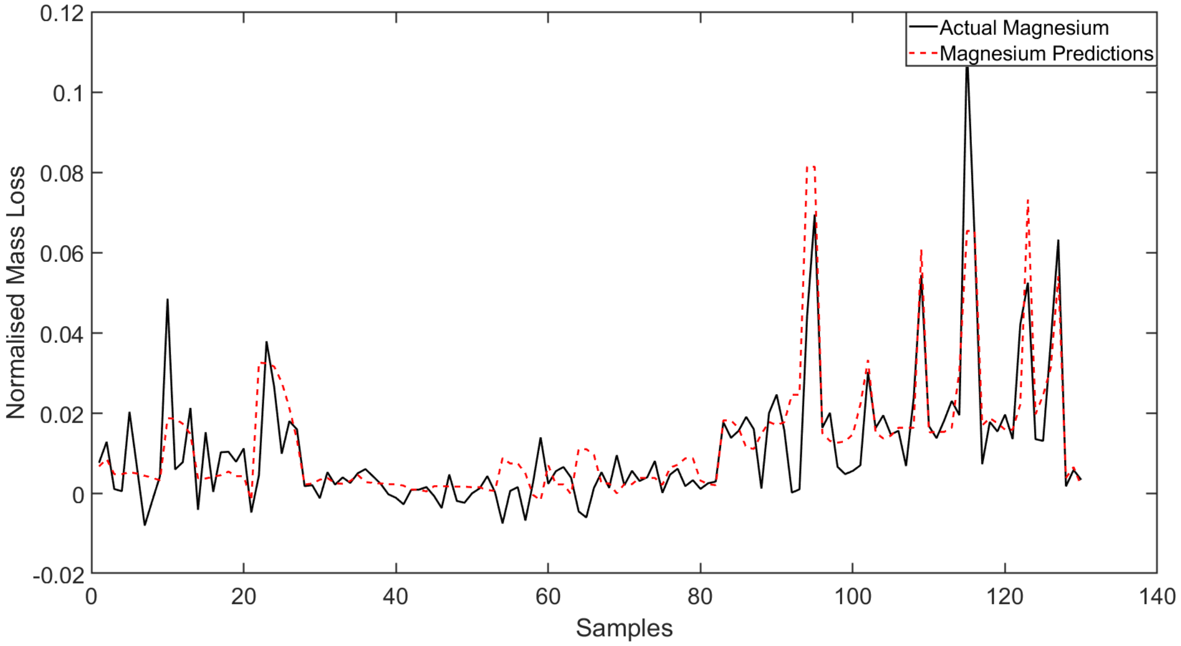

Figure 7.13 – Normalised mass loss predictions for magnesium on validation data, using the PCA-BA-ELM model

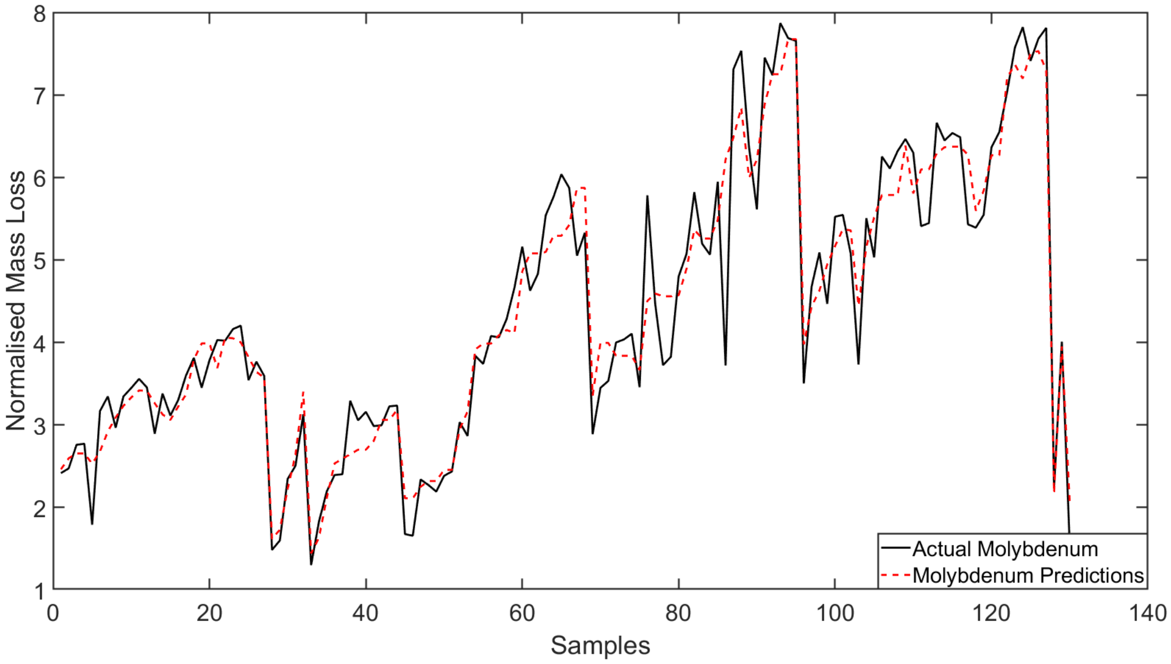

Figure 7.14 – Normalised mass loss predictions for molybdenum on validation data, using the PCA-BA-ELM model

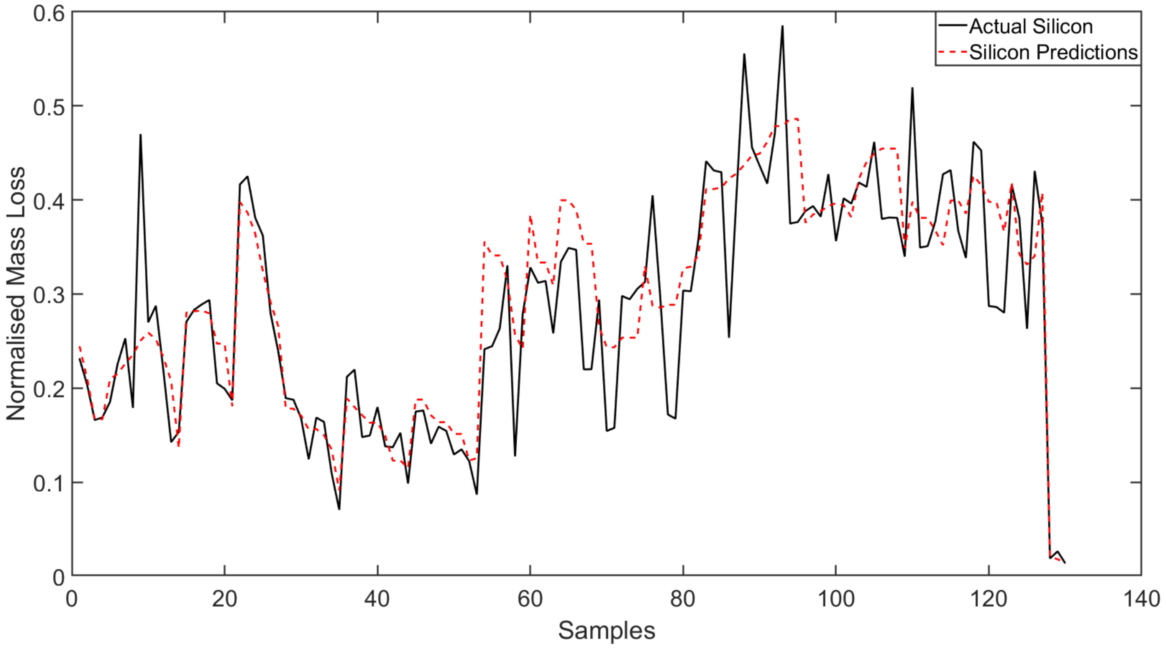

Figure 7.15 – Normalised mass loss predictions for silicon on validation data, using the PCA-BA-ELM model

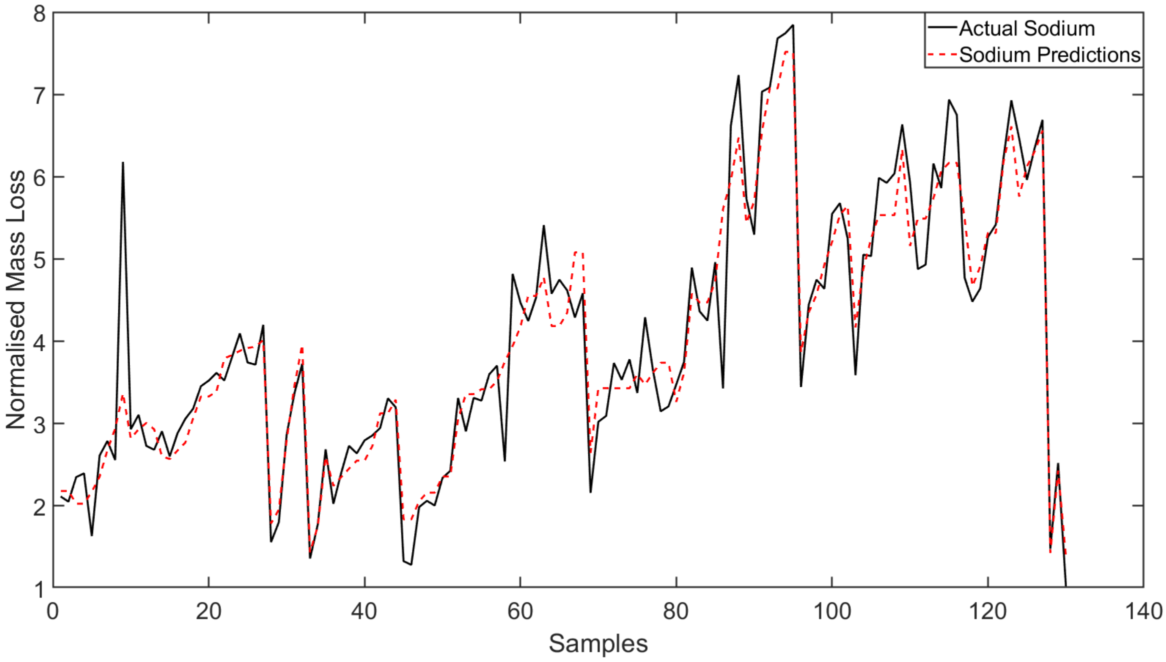

Figure 7.16 – Normalised mass loss predictions for sodium on validation data, using the PCA-BA-ELM model

7.5 Sensitivity Analysis

A model-based sensitivity analysis was carried out to assess how the four main glass forming additives affect the normalised mass loss of each element. The profile method (Gevrey *et al.*, 2003) was used for this sensitivity analysis. The inputs of interest were the four glass forming additives as these can be most easily manipulated in the feed. For the profile method, the input of interest is varied over its full range and each point over the range is used as an input to the model to create a prediction. 12 points within the range of each input was used. All other inputs are held at a constant level. Typically, this constant level is a quartile and then the process is repeated for all quartiles. However, for this data, some combinations of inputs created from these constant levels were outside the scope of the model because they were too different from anything seen in the data due to such a small amount of data available. These combinations lead to infeasible predictions. Therefore, it was decided to make the constant levels equivalent to the glass compositions from the experimental data. The predictions for each constant level were then averaged to give the final sensitivity output.

For example, for the sensitivity analysis for the boron oxide additive:

- Boron oxide is varied 12 times between the minimum and maximum values that are observed for it over all 26 glass compositions provided in the data (26 “constant levels”).
- For all 12 boron oxide values, the other inputs are kept constant at their values corresponding to glass composition 1.
- These 12 input vectors are used to create 12 model predictions of normalised mass loss based on the PCA-BA-ELM model for a chosen element.
- This process is then repeated for the other 25 glass compositions, where the same 12 boron oxide values are used, and the other inputs are kept at the value corresponding to the glass compositions.
- This leads to a 12x26 matrix of normalised mass loss predictions. The 26 predictions for each boron oxide value (relating to each “constant level”) are then averaged, giving 12 predictions of normalised mass loss for range of boron oxide values.

Figure 7.17 shows the sensitivity analysis for the boron PCA-BA-ELM model at the end time of 182 days, because it is one of the elements that contributes to the total normalised mass loss the most. The final time point of 182 days is focused on because this is when the glass dissolution is at saturation and then normalised mass loss is at its greatest. If the saturation normalised mass loss can be reduced then the prior time points should also be reduced. Figure 7.17 shows that there is a potential optimum amount of boron oxide, while the other three additives all show an almost linear relationship between the boron normalised mass loss and the composition of the additive. Although this suggests that these inputs should be minimised, some of the additives are important for thermal and chemical stability of the product. However, within the ranges used in the model, these should be minimised.

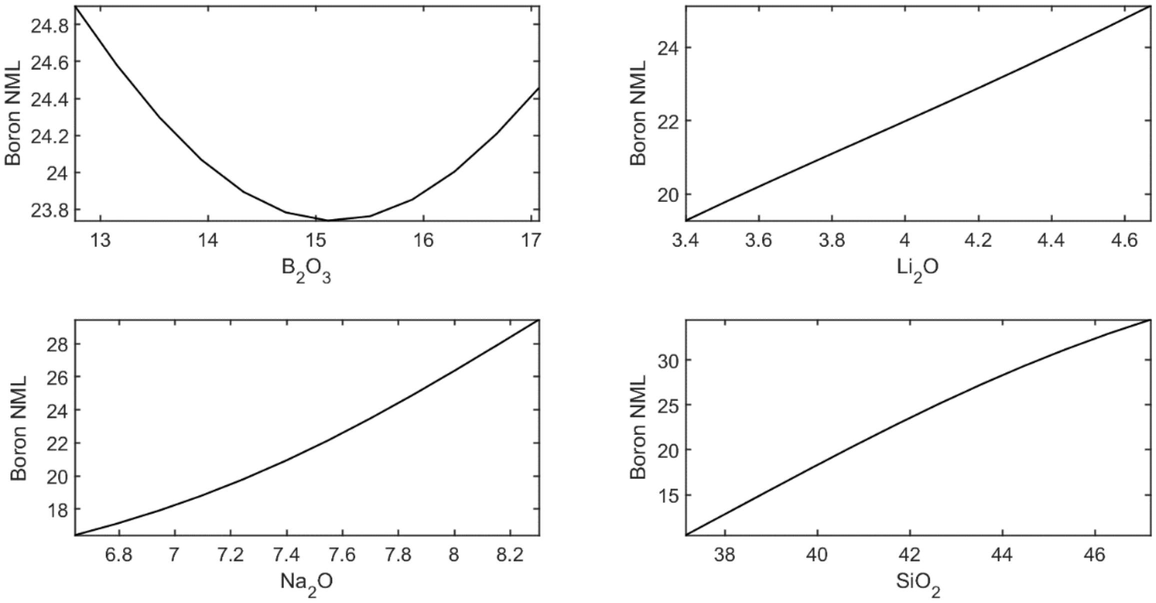

Figure 7.17 – Sensitivity analysis of boron normalised mass loss for a time of 182 days.

Looking at a sensitivity analysis of the total normalised mass loss is of most interest because it is the total that matters the most when it comes to total radioactivity leaching into the environment. The same sensitivity analysis method was applied except the normalised mass loss from each of the eight individual models was summed to give a total.

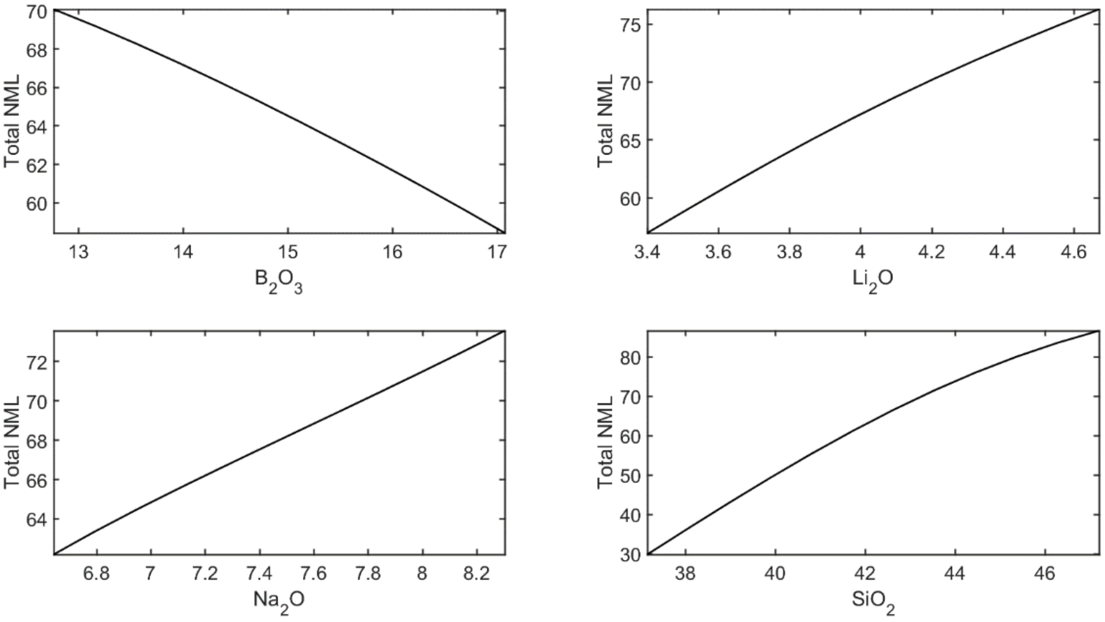

Figure 7.18 – Sensitivity analysis of the total normalised mass loss for a time of 182 days

Figure 7.18 shows the sensitivity analysis for the total normalised mass loss at 182 days. The trends for lithium oxide, sodium oxide and silicon oxide are very similar, however, the boron oxide trend is different. As the composition of boron oxide increases the total normalised mass loss decreases linearly. This shows that using less lithium, sodium and silicon oxides, but more boron oxides (within the ranges specified by the experimental data) maximise the chemical durability of the glass waste (within the glass types provided by in the experimental data).

7.6 Conclusions

Nuclear energy is still an important fuel source and dealing with nuclear waste is one of the key issues associated with it. High-level nuclear waste can be converted to a more stable, glass form through a process known as vitrification. If the vitrified waste is to be stored in deep geological repositories, then eventually the engineered barriers will corrode, and the radioactive waste may come into contact with groundwater. The chemical durability of the waste is the amount of elements that leach into the ground water and so being able to model this for different glass compositions can lead to minimising the leach rate. Bootstrap aggregated extreme learning machine integrated with principal component analysis was used to model the normalised mass loss of eight elements using experimental data. It was shown that robust and reliable models of the normalised mass losses were produced using this technique when compared with similar techniques, such as single ELM, and bagged neural network. Combining multiple models increased the generalisation performance compared to single ELM models and also improved the model reliability. Including principal component analysis further improve model generalisation performance by reducing collinearity and dimensionality. The developed models were used as a part of a sensitivity analysis to assess how the four main glass forming additives affect the normalised mass loss, which could be used to adjust the feed composition to minimise the normalised mass loss. Future work could look at a model-based optimisation to determine the best feed composition of glass additives to minimise the normalised mass loss, hence maximising the chemical durability. Additionally, more data is required to further improve generalisation performance and to validate the model further, because carrying out the sensitivity analysis highlighted that the model cannot handle any combination of glass compositions due to the small amount of data the models were built on.

Chapter 8. Conclusions and Recommendations for Further Works

8.1 Summary of Thesis and Main Contributions

Data-driven process modelling has vastly improved in recent years due to the ever-increasing amount of process data being stored and hardware advances meaning more complex techniques can be used, and with less computational time and effort. However, issues with data pre-processing and selection, and developing techniques to enhance model reliability and robustness still exist.

Nuclear energy is key to tackling the withdrawal of fossil fuels as the main source of electricity in the world. One of the key issues with nuclear energy is dealing with the radioactive waste that cannot be recycled. The nuclear waste vitrification plant at Sellafield Ltd gives one option for dealing with high-level waste from nuclear fuel reprocessing. The high-level waste is converted into a glass form, which is more stable and gives a lower volume than the original waste. This is critical as the waste remains radioactive for hundreds or thousands of years and so a stable form of the waste is important. Utilising data-driven modelling and monitoring to optimise parts of the process will lead to a reduction in costs and environmental impact.

This thesis firstly focuses on developing novel, data-driven modelling techniques to enhance model performance for industrial process applications. Secondly, utilising data-driven modelling and monitoring to optimise the nuclear waste vitrification process at Sellafield Ltd.

Chapter 2 provides an overview of data-driven process modelling and monitoring. Process data pre-processing issues are highlighted and previous literature on these is reviewed. Typical data-driven techniques and their associated applications in the process industry are also described.

Slow feature analysis is a statistical technique that can extract underlying trends in process data, while also reducing the effects of noise and dimensionality. Utilising slow feature analysis as a pre-processing technique to data-driven modelling has been previously done for linear applications. However, combining slow feature analysis with a neural network was a novel contribution, as laid out in Chapter 3. Two industrial process case studies were used as benchmarks to assess the proposed methodology compared with other similar, standard data-driven modelling techniques. The results showed the improvements to model generalisation performance that slow feature analysis provided by extracting underlying trends (providing more meaningful information to the regression technique), reducing noise, and reducing dimensionality. All these attributes lead to an information rich scenario that is beneficial to model performance.

Expanding on linear slow feature analysis to extract nonlinear underlying trends in process data led to the development of kernel slow feature analysis as a precursor to a neural network, another novel contribution, described in Chapter 4. A nonlinear numerical example was developed to demonstrate that kernel slow feature analysis can extract driving forces behind the data where linear slow feature analysis cannot do so fully. An industrial batch process case study demonstrated that better nonlinear slow feature extraction leads to significant improvements in model generalisation when combined with a neural network, compared with a variety of techniques, including linear slow feature analysis and kernel principal component analysis.

The rest of the thesis involved taking the data-driven modelling techniques developed and applying them to real, novel industrial applications for Sellafield Ltd's nuclear waste vitrification process. A key stage in the vitrification process is the pouring of the waste glass into containers. Overfilling and underfilling of the containers can be very costly and so understanding the pour rate is critical. There is currently no way to measure the pour rate in real time and only the average pour rate can be calculated. Having a prediction of the pour rate just before the pour starts can optimise the pouring. Using process data to create a predictive model of the pour rate is described in Chapter 5, which has not been attempted in previous works. Slow feature analysis was incorporated into a bootstrap aggregated neural network methodology to improve model predictions when compared with principal component analysis-based bootstrap aggregated methods. Additionally, the results demonstrate that bootstrap aggregation improves the model robustness and reliability over single models.

Another challenge in the waste vitrification process is blockages developing in pipe work in the dust scrubber section of the process. These blockages accumulate quickly and can lead to very expensive maintenance to clear them. Multivariate statistical process control approaches using slow feature analysis and principal component analysis were developed to try and detect these blockages earlier than they typically are on site, as presented in Chapter 6. Earlier detection of the blockages leads to more preventive measures being applied to ensure a full blockage does not occur. The results demonstrated that early detection was improved using slow feature analysis over principal component analysis. Additionally, including external dynamics into the model improved the early detection times further. It is also of interest to understand if there are any specific process variables driving the blockages and so residual contribution plots were created. However, these plots did not provide any conclusive insight into the direct causes of the blockages besides the contributions of dust scrubber differential pressure, which was already known.

The final piece of work for the waste vitrification process involved data-driven modelling of the chemical durability of the waste glass product. The containers of glass are likely to remain radioactive for a very long time and can be stored in deep geological repositories. Eventually the containers may erode, and the glass comes into contact with groundwater, causing the leaching of radionuclides into the environment. The rate of this leaching of radionuclides (known as the chemical durability) needs to be minimised to avoid long term environmental issues. Using accelerated static leach rate experiments, the long-term chemical durability of several key radioactive elements can be measured. Data-driven modelling of these leach rates using this experimental data were created using bootstrap aggregated extreme learning machine for glasses of different compositions. Good generalisation was performed for 7 out of the 8 element models, with PCA as a pre cursor to BA- ELM giving the best generalisation when compared with just BA-ELM and with BA-NN and PCA-BA-NN. A sensitivity analysis was carried out to assess whether the glass composition could be optimised to minimise the overall leach rate. The results suggested that increasing the composition of boron oxide glass additive, while decreasing the other three additives, within the range of compositions assessed, could minimise the total leach rate.

8.2 Recommendations for Future Research

The work into linear and kernel slow feature analysis in Chapters 3 and 4 could be further expanded. Selection of optimal hyperparameters is difficult, particularly for kernel slow feature analysis with the amount of hyperparameters involved, and so further research into this could improve model performance. Further assessing the effectiveness of kernel slow feature analysis with a neural network on more case studies and against more techniques should also be done.

The pour rate models in Chapter 5 could be improved by incorporating more complex techniques such as kernel slow feature analysis. Furthermore, analysis to investigate which process variables impact the pour rate the most could lead to control of the pour rate being possible, something that would significantly improve process operations because currently the pour rate is not controlled in such a way.

The results on early detection of blockages could be further improved by incorporating kernel slow feature analysis or kernel principal component analysis, to capture nonlinear underlying trends in the data to improve the false alarm rate and relative detection time. Additionally, the residual contribution plots did not yield impactful results and so using other type of contribution plots, and other methods to try to find what variables cause the blockages, is of great importance to process operations.

In addition to the sensitivity analysis for the chemical durability modelling, model-based optimisation could be performed to find an optimal glass composition to maximise chemical durability. Furthermore, the chemical durability models can be retrained if more experimental data is available to improve the range of compositions the model performs well on.

Appendix A

Introduction

This appendix provides additional results relating to the dust scrubber blockage detection from Chapter 6.

PCA control charts for blockages 2-7 with 99.99% control limit

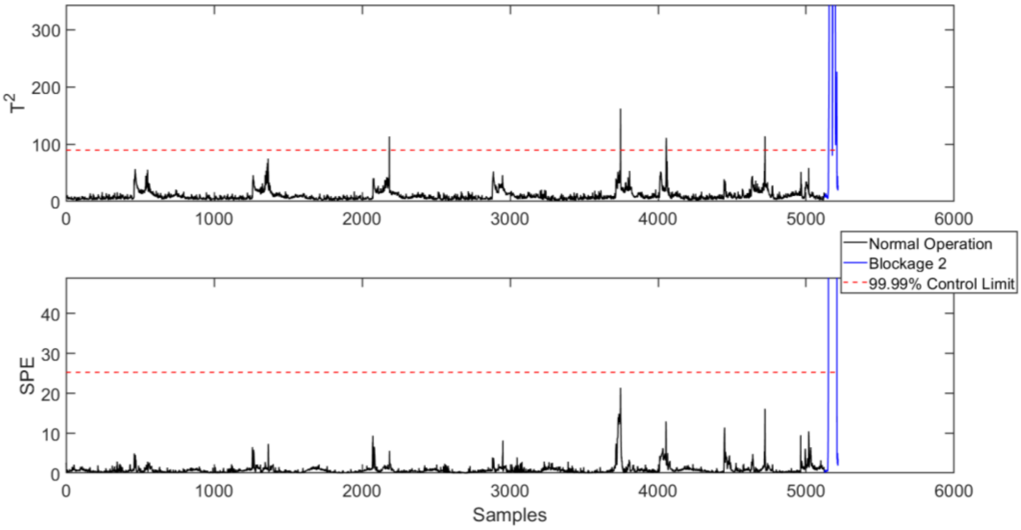

Figure A.1 – T^2 and SPE control chart for blockage 2, using a PCA model

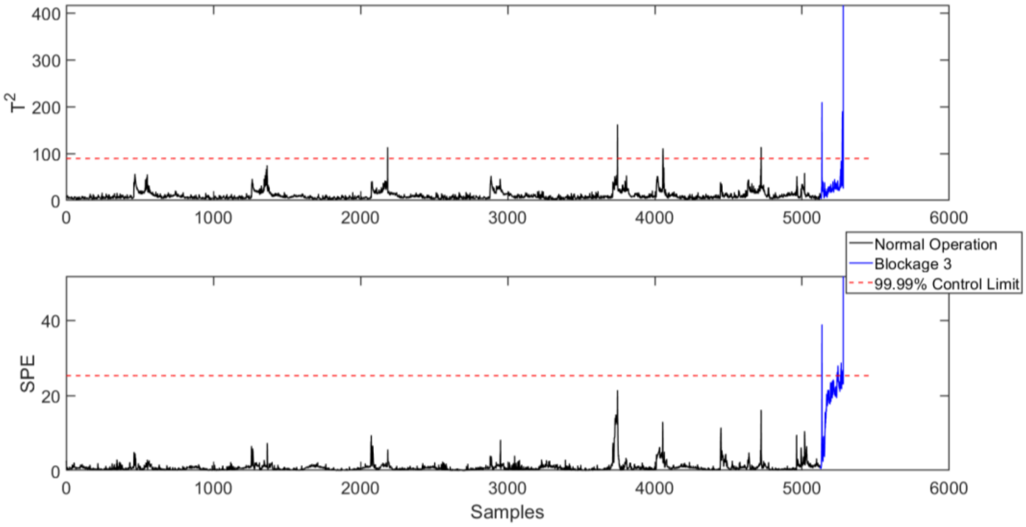

Figure A.2 – T^2 and SPE control chart for blockage 3, using a PCA model

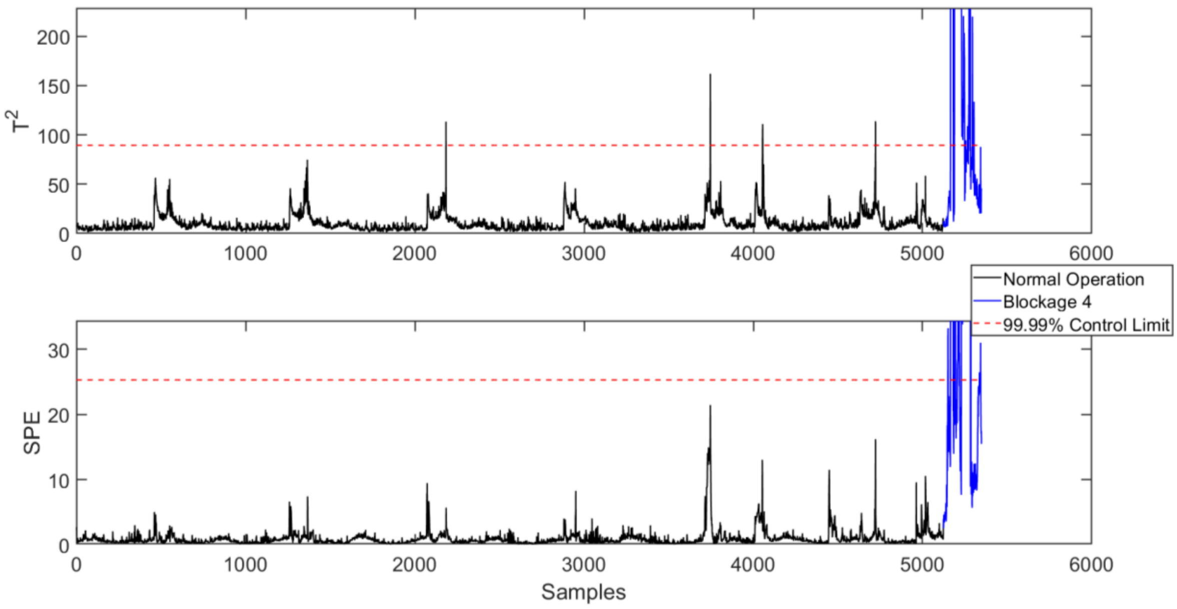

Figure A.3 – T^2 and SPE control chart for blockage 4, using a PCA model

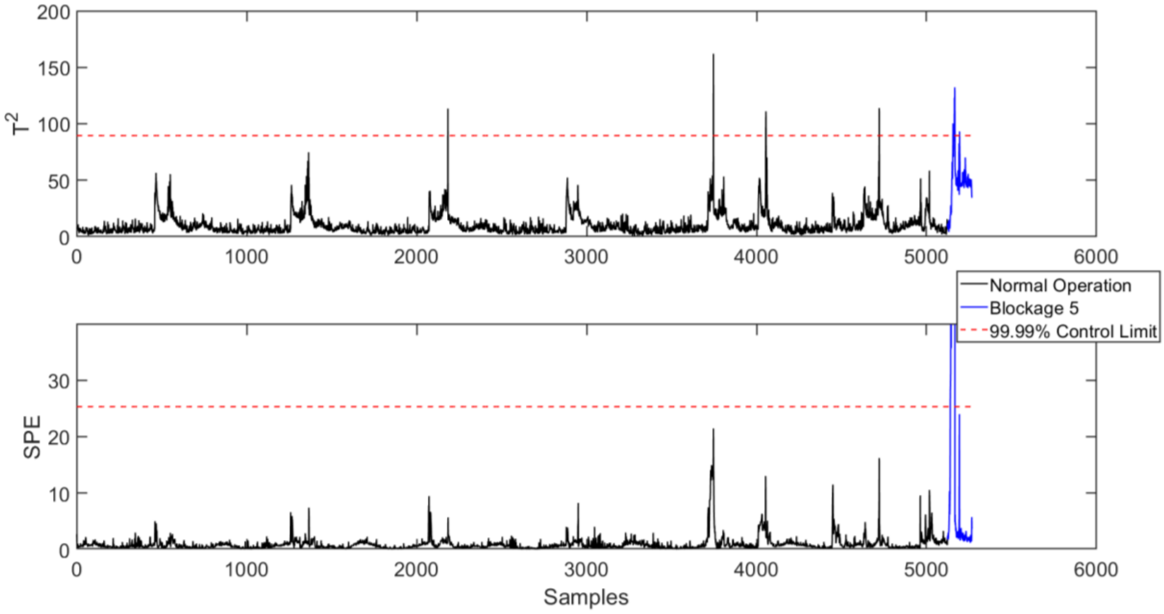

Figure A.4 – T^2 and SPE control chart for blockage 5, using a PCA model

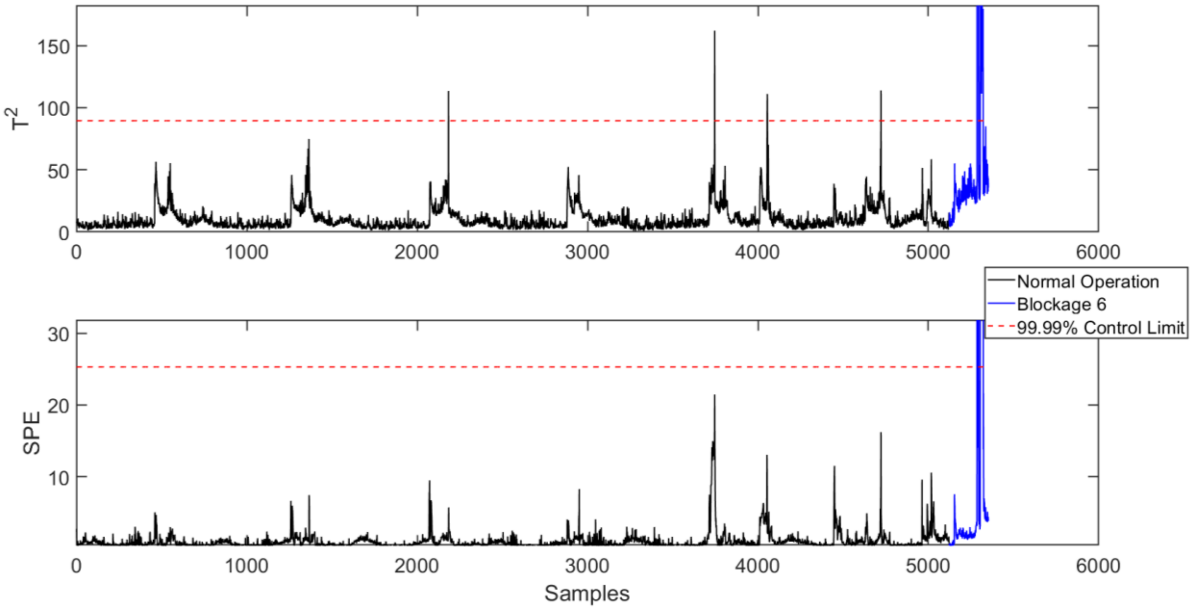

Figure A.5 – T^2 and SPE control chart for blockage 6, using a PCA model

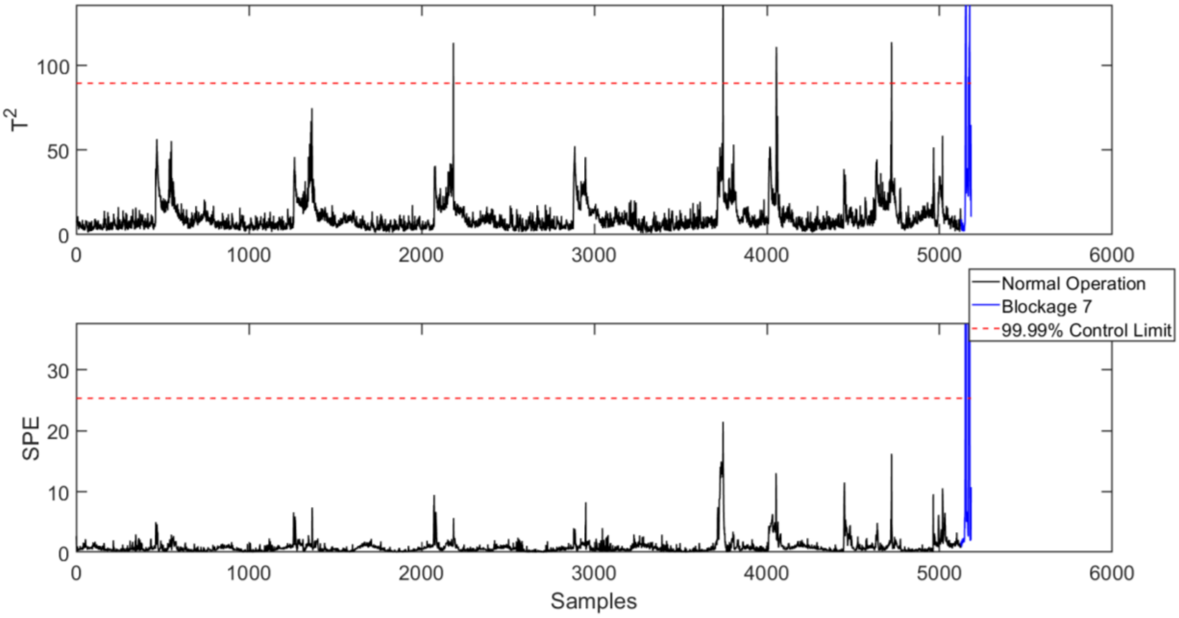

Figure A.6 – T^2 and SPE control chart for blockage 7, using a PCA model

DPCA relative detection times for d values of 1, 2 and 4.

Table A.1 – Relative detection times for each blockage using a DPCA-based control chart, for d = 1

	Relative detection time (min)
Blockage 1	-5
Blockage 2	-3
Blockage 3	-19
Blockage 4	-3
Blockage 5	-12
Blockage 6	-3
Blockage 7	-5
Mean	-7.14

Table A.2 – Relative detection times for each blockage using a DPCA-based control chart, for d = 2

	Relative detection time (min)
Blockage 1	-7
Blockage 2	-5
Blockage 3	-21
Blockage 4	-5
Blockage 5	-15
Blockage 6	-4
Blockage 7	-7
Mean	-9.14

Table A.3 – Relative detection times for each blockage using a DPCA-based control chart, for d = 4

	Relative detection time (min)
Blockage 1	-8
Blockage 2	-6
Blockage 3	-23
Blockage 4	-8
Blockage 5	-18
Blockage 6	-4
Blockage 7	-11
Mean	-11.14

Contribution plots for blockages 2-7 using the DSFA model (best model)

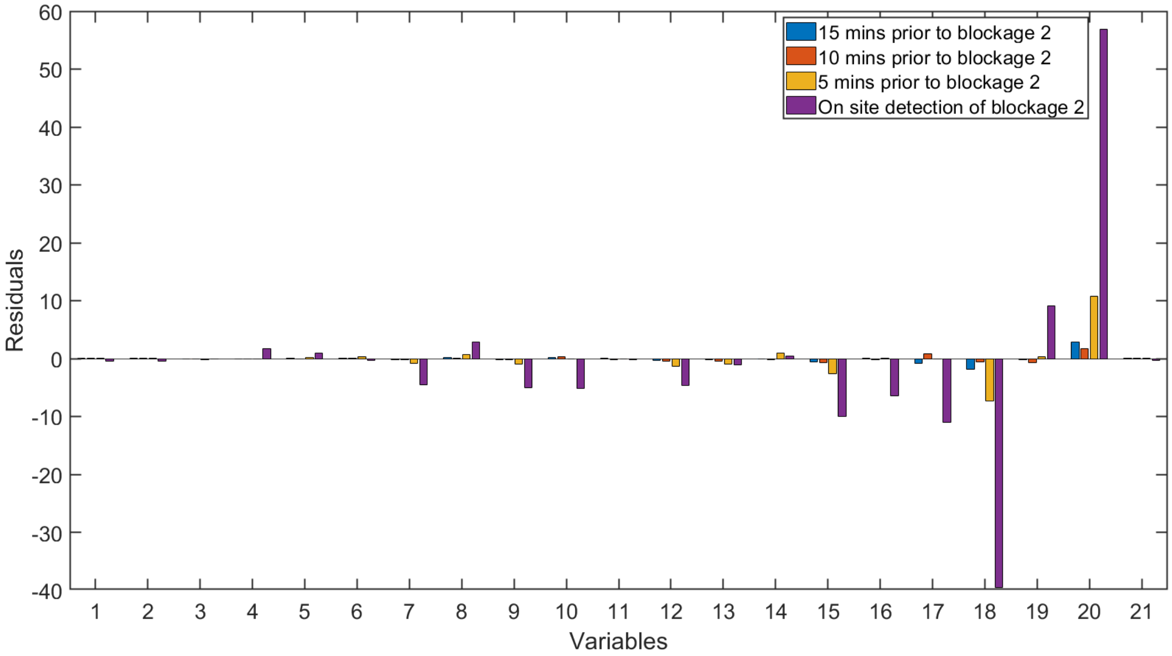

Figure A.7 – Residual contribution plot for DSFA model for blockage 2

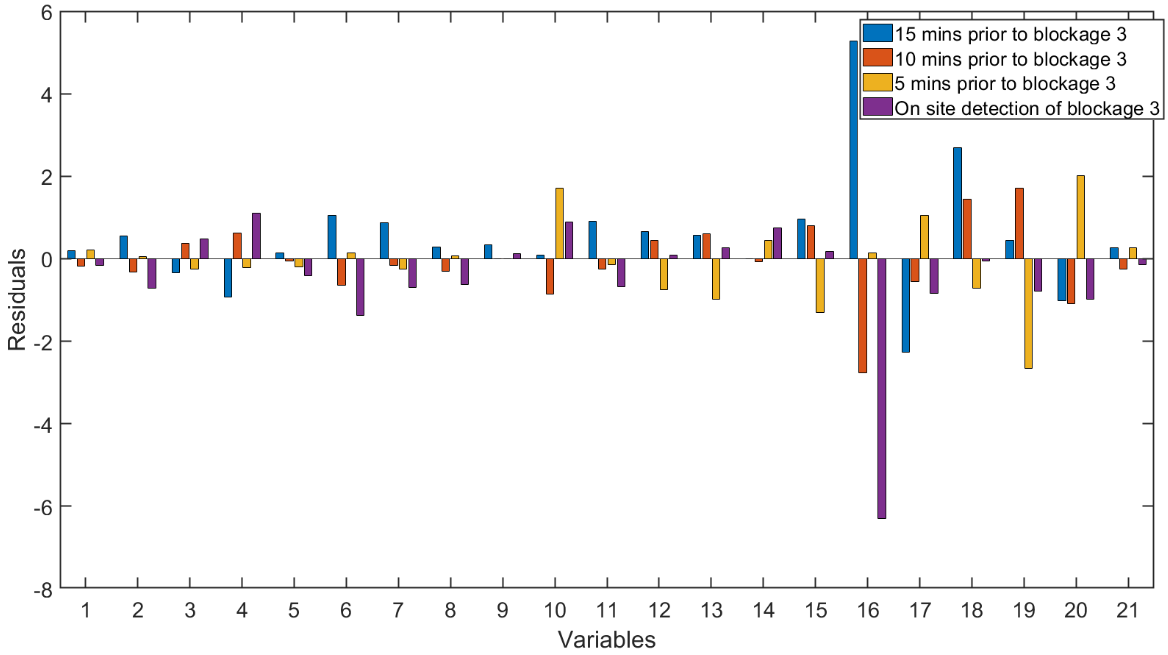

Figure A.8 – Residual contribution plot for DSFA model for blockage 3

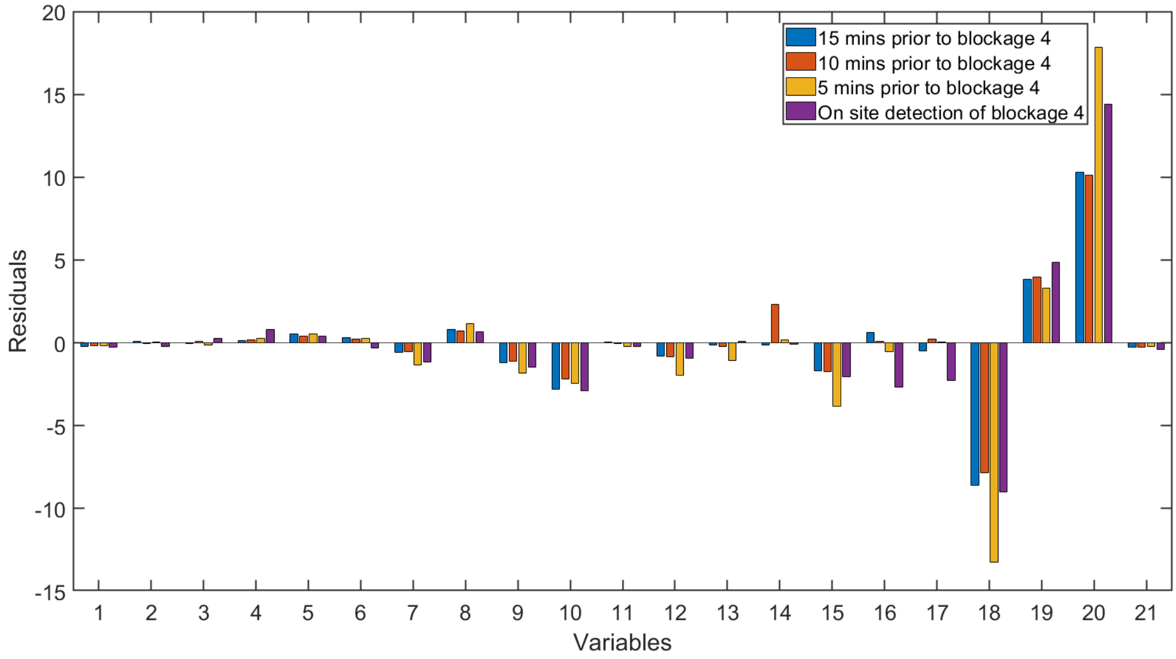

Figure A.9 – Residual contribution plot for DSFA model for blockage 4

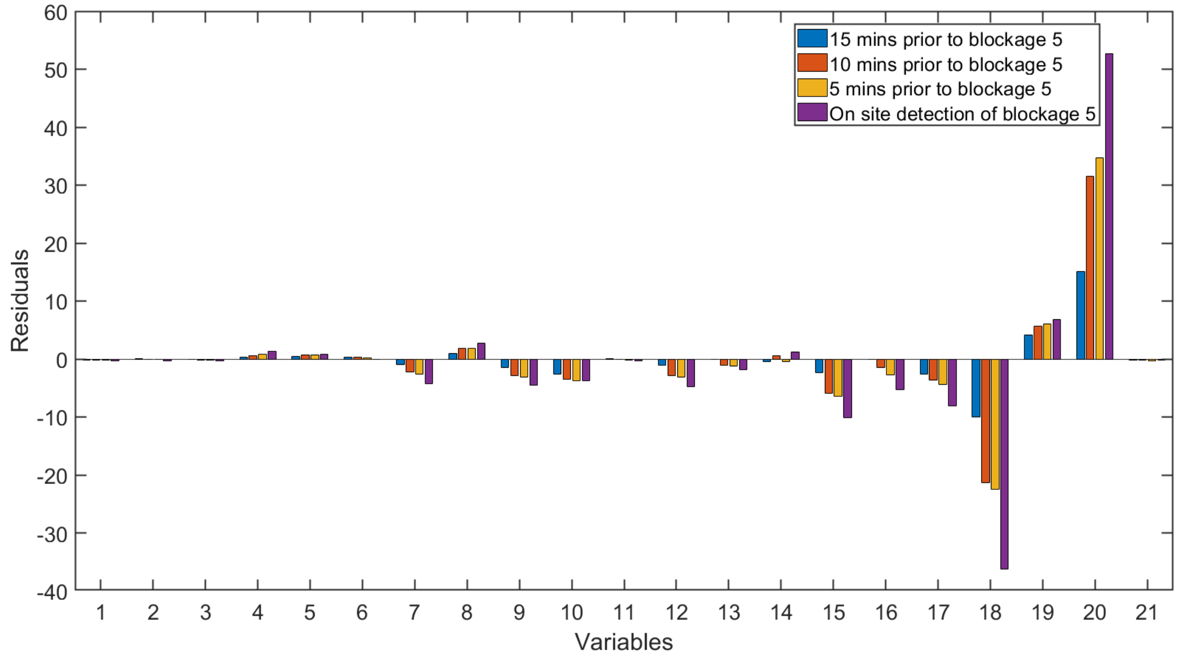

Figure A.10 – Residual contribution plot for DSFA model for blockage 5

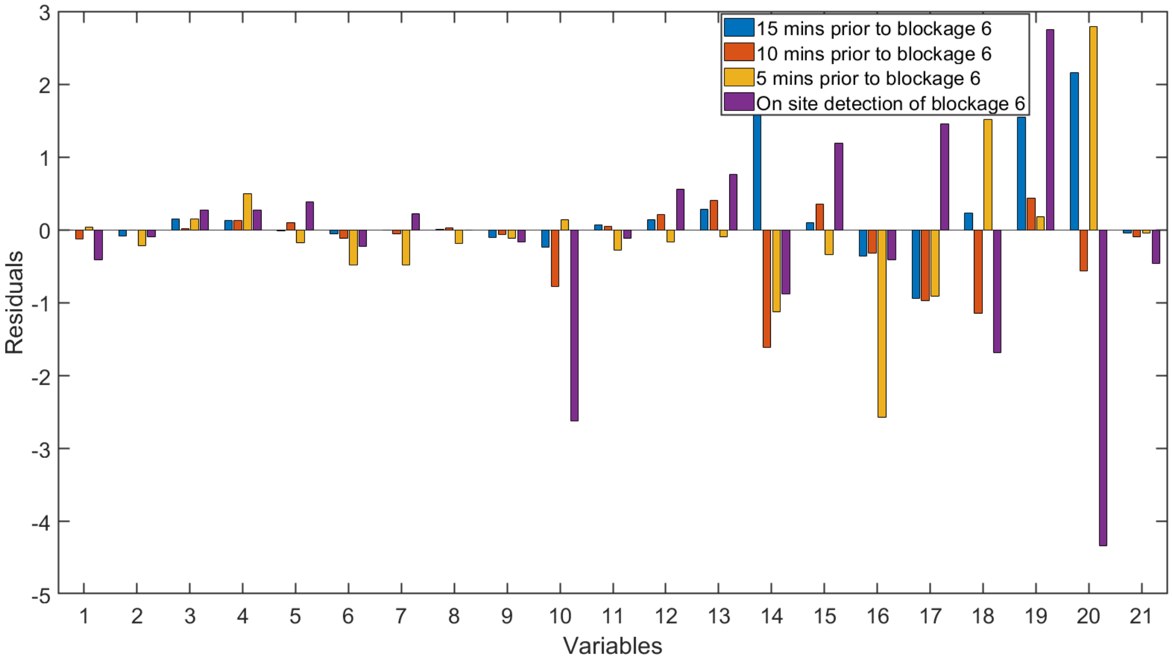

Figure A.11 – Residual contribution plot for DSFA model for blockage 6

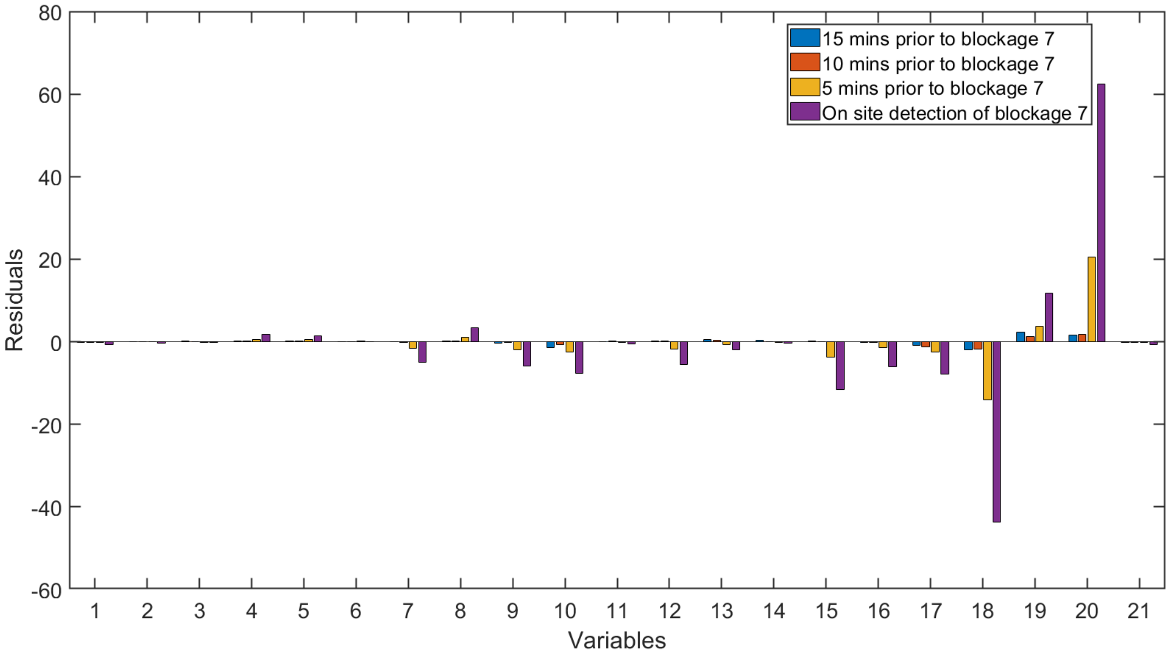

Figure A.12 – Residual contribution plot for DSFA model for blockage 7

References

Ahmad, Z., Noor, R.A.M. and Zhang, J. (2009) 'Multiple neural networks modeling techniques in process control: a review', *Asia-Pacific Journal of Chemical Engineering*, 4(4), pp. 403-419.

Ahmad, Z. and Zhang, J. (2005a) 'Bayesian selective combination of multiple neural networks for improving long-range predictions in nonlinear process modelling', *Neural Computing & Applications*, 14(1), pp. 78-87.

Ahmad, Z. and Zhang, J. (2005b) 'Combination of multiple neural networks using data fusion techniques for enhanced nonlinear process modelling', *Computers & Chemical Engineering*, 30(2), pp. 295-308.

Alcala, C.F. and Qin, S.J. (2009) 'Reconstruction-based contribution for process monitoring', *Automatica*, 45(7), pp. 1593-1600.

Alcala, C.F. and Qin, S.J. (2010) 'Reconstruction-based contribution for process monitoring with kernel principal component analysis', *Industrial & Engineering Chemistry Research*, 49(17), pp. 7849-7857.

Backhouse, D.J. (2017) *A Study of the Dissolution of Nuclear Waste Glasses in Highly-Alkaline Conditions*. PhD Thesis. The University of Sheffield.

Baraldi, A.N. and Enders, C.K. (2010) 'An introduction to modern missing data analyses', *Journal of School Psychology*, 48(1), pp. 5-37.

Barnett, V. (1994) *Outliers in Statistical Data*. 3rd ed.. edn. Chichester, New York: Wiley & Sons.

Bauer, E. and Kohavi, R. (1999) 'An empirical comparison of voting classification algorithms: Bagging, boosting, and variants', *Machine Learning*, 36(1-2), pp. 105-139.

Bella, A.D., Fortuna, L., Graziani, S., Napoli, G. and Xibilia, M.G. (2007) *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*. 1-3 May 2007.

Bersimis, S., Psarakis, S. and Panaretos, J. (2007) 'Multivariate statistical process control charts: An overview', *Quality and Reliability Engineering International*, 23(5), pp. 517-543.

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.

Birol, G., Ündey, C. and Çinar, A. (2002) 'A modular simulation package for fed-batch fermentation: penicillin production', *Computers & Chemical Engineering*, 26(11), pp. 1553-1565.

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

Bland, J.M. and Altman, D.G. (2015) 'Statistics notes: Bootstrap resampling methods', *BMJ*, h2622, p. 350.

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

Brookes, C., Harrison, M.T., Riley, A. and Steele, C. (2010) 'The effect of increased waste loading on the durability of high level waste glass', *MRS Online Proceedings Library*, 1265(1), p. 305.

Brown, G., Wyatt, J., Harris, R. and Yao, X. (2005) 'Diversity creation methods: a survey and categorisation', *Information Fusion*, 6(1), pp. 5-20.

Cattell, R.B. (1966) 'The scree test for the number of factors', *Multivariate Behavioral Research*, 1(2), pp. 245-276.

Chai, Z. and Zhao, C.H. (2020) 'Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification', *IEEE Transactions on Industrial Informatics*, 16(1), pp. 54-66.

Chang, P. and Li, Z. (2021) 'Over-complete deep recurrent neutral network based on wastewater treatment process soft sensor application', *Applied Soft Computing*, 105, p. 107227.

Chen, N., Dai, J., Yuan, X., Gui, W., Ren, W. and Koivo, H.N. (2018) 'Temperature prediction model for roller kiln by ALD-based double locally weighted kernel principal component regression', *IEEE Transactions on Instrumentation and Measurement*, 67(8), pp. 2001-2010.

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.

Corrigan, J. and Zhang, J. (2020) 'Integrating dynamic slow feature analysis with neural networks for enhancing soft sensor performance', *Computers & Chemical Engineering*, 139, p. 106842.

Cybenko, G. (1989) 'Approximation by superpositions of a sigmoidal function', *Mathematics of Control, Signals and Systems*, 2(4), pp. 303-314.

de Assis, A.J. and Maciel, R. (2000) 'Soft sensors development for on-line bioreactor state estimation', *Computers & Chemical Engineering*, 24(2-7), pp. 1099-1103.

Donald, I.W., Metcalfe, B.L. and Taylor, R.N.J. (1997) 'The immobilization of high level radioactive wastes using ceramics and glasses', *Journal of Materials Science*, 32(22), pp. 5851-5887.

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.

Dunia, R., Qin, S.J., Edgar, T.F. and McAvoy, T.J. (1996) 'Identification of faulty sensors using principal component analysis', *AIChE Journal*, 42(10), pp. 2797-2812.

Fan, L., Kodamana, H. and Huang, B.A. (2018) 'Identification of robust probabilistic slow feature regression model for process data contaminated with outliers', *Chemometrics and Intelligent Laboratory Systems*, 173, pp. 1-13.

Feng, L., Zhao, C. and Huang, B. (2019) 'A slow independent component analysis algorithm for time series feature extraction with the concurrent consideration of high-order statistic and slowness', *Journal of Process Control*, 84, pp. 1-12.

Ferguson, K. (2013) *Applications Of Multivariate Statistical Modelling In The Nuclear Industry*. MPhil Thesis, Newcastle University.

Ferguson, K., Zhang, J., Steele, C., Clarke, C. and Morris, J. (2011) ‘Modelling vitrified glass viscosity in a nuclear fuel reprocessing plant using neural networks’, *Proceedings of the International Conference on Neural Computation Theory and Applications (NTCA2011)*, 24 – 26 October 2011, Paris, France, pp. 322-325.

Fernandez de Canete, J., del Saz-Orozco, P., Gómez-de-Gabriel, J., Baratti, R., Ruano, A. and Rivas-Blanco, I. (2021) 'Control and soft sensing strategies for a wastewater treatment plant using a neuro-genetic approach', *Computers & Chemical Engineering*, 144, p. 107146.

Fortuna, L., Graziani, S., Rizzo, A. and Xibilia, M. G. (2007) *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer. Available at: https://link.springer.com/book/10.1007%2F978-1-84628-480-9.

Foschi, J., Turolla, A. and Antonelli, M. (2021) 'Soft sensor predictor of E. coli concentration based on conventional monitoring parameters for wastewater disinfection control', *Water Research*, 191, p. 116806.

Freund, Y. and Schapire, R.E. (1997) 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of Computer and System Sciences*, 55(1), pp. 119-139.

Frugier, P., Chave, T., Gin, S. and Lartigue, J.E. (2009) 'Application of the GRAAL model to leaching experiments with SON68 nuclear glass in initially pure water', *Journal of Nuclear Materials*, 392(3), pp. 552-567.

Gao, X. and Shardt, Y.A.W. (2021) 'Dynamic system modelling and process monitoring based on long-term dependency slow feature analysis', *Journal of Process Control*, 105, pp. 27-47.

Garcia-Munoz, S. and Macgregor, J.F. (2016) 'Big Data: Success Stories in the Process Industries', *Special Section*, [Online]. Available at: https://www.aiche.org/resources/publications/cep/2016/march/big-data-success-stories-process-industries.

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.

Gevrey, M., Dimopoulos, L. and Lek, S. (2003) 'Review and comparison of methods to study the contribution of variables in artificial neural network models', *Ecological Modelling*, 160(3), pp. 249-264.

Goel, A., McCloy, J.S., Pokorny, R. and Kruger, A.A. (2019) 'Challenges with vitrification of Hanford high-level waste (HLW) to borosilicate glass – An overview', *Journal of Non-Crystalline Solids: X*, 4, p. 100033.

Gonzalez, G.D. (1999) 'Soft sensors for processing plants', *Intelligent Processing and Manufacturing of Materials, 1999. IPMM '99. Proceedings of the Second International Conference on*. Honolulu, HI, USA, USA. IEEE.

Gopakumar, V., Tiwari, S. and Rahman, I. (2018) 'A deep learning based data driven soft sensor for bioprocesses', *Biochemical Engineering Journal*, 136, pp. 28-39.

Guo, Y., Zhao, Y. and Huang, B. (2014) 'Development of soft sensor by incorporating the delayed infrequent and irregular measurements', *Journal of Process Control*, 24(11), pp. 1733-1739.

Hangos, K.M. (2001) *Process Modelling and Model Analysis*. San Diego: San Diego : Academic Press.

Harrison, M.T. (2014) 'The effect of composition on short- and long-term durability of UK HLW glass', *Procedia Materials Science*, 7, pp. 186-192.

Harrison, M.T. and Brown, G.C. (2018) 'Chemical durability of UK vitrified high level waste in Si-saturated solutions', *Materials Letters*, 221, pp. 154-156.

Harrison, M.T. and Scales, C.R. (2008) 'Durability of borosilicate glass compositions for the immobilisation of the UK's separated plutonium stocks', *MRS Online Proceedings Library*, 1107(1), p. 429.

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.

Hastie, T. (2009) *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. 2nd ed.. edn. New York: New York : Springer.

He, Y.L., Geng, Z.Q. and Zhu, Q.X. (2015) 'Data driven soft sensor development for complex chemical processes using extreme learning machine', *Chemical Engineering Research & Design*, 102, pp. 1-11.

Hong, J.J., Zhang, J. and Morris, J. (2014) 'Progressive multi-block modelling for enhanced fault isolation in batch processes', *Journal of Process Control*, 24(1), pp. 13-26.

Hrma, P. (2008) 'Arrhenius model for high-temperature glass-viscosity with a constant pre-exponential factor', *Journal of Non-Crystalline Solids*, 354(18), pp. 1962-1968.

Hu, G., Mao, Z., He, D. and Yang, F. (2011) 'Hybrid modeling for the prediction of leaching rate in leaching process based on negative correlation learning bagging ensemble algorithm', *Computers & Chemical Engineering*, 35(12), pp. 2611-2617.

Huang, G., Huang, G.B., Song, S.J. and You, K.Y. (2015) 'Trends in extreme learning machines: A review', *Neural Networks*, 61, pp. 32-48.

Huang, G., Song, S.J., Gupta, J.N.D. and Wu, C. (2014) 'Semi-supervised and unsupervised extreme learning machines', *IEEE Transactions on Cybernetics*, 44(12), pp. 2405-2417.

Huang, G.B., Zhou, H.M., Ding, X.J. and Zhang, R. (2012) 'Extreme learning machine for regression and multiclass classification', *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 42(2), pp. 513-529.

Huang, G.B., Zhu, Q.Y. and Siew, C.K. (2006) 'Extreme learning machine: Theory and applications', *Neurocomputing*, 70(1-3), pp. 489-501.

Huang, J., Yang, X. and Yan, X. (2020) 'Slow feature analysis-independent component analysis based integrated monitoring approach for industrial processes incorporating dynamic and static characteristics', *Control Engineering Practice*, 102, p. 104558.

Jackson, J.E. and Mudholkar, G.S. (1979) 'Control procedures for residuals associated with principal component analysis', *Technometrics*, 21(3), pp. 341-349.

Jia, Q., Cai, J., Jiang, X. and Li, S. (2020) 'A subspace ensemble regression model based slow feature for soft sensing application', *Chinese Journal of Chemical Engineering*, 28(12), pp. 3061-3069.

Jolliffe, I.T. (2002) *Principal Component Analysis*. 2nd edn. 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.

Kadlec, P., Grbic, R. and Gabrys, B. (2011) 'Review of adaptation mechanisms for data-driven soft sensors', *Computers & Chemical Engineering*, 35(1), pp. 1-24.

Kaiser, H.F. (1960) 'The application of electronic computers to factor analysis', *Educational and Psychological Measurement*, 20, pp. 141-151.

Kalman, R.E. (1960) 'A new approach to linear filtering and prediction problems', *Journal of Basic Engineering*, 82(1), pp. 35-45.

Kaneko, H. and Funatsu, K. (2015) 'Smoothing-combined soft sensors for noise reduction and improvement of predictive ability', *Industrial & Engineering Chemistry Research*, 54(50), pp. 12630-12638.

Kaunga, D.L., Zhang, J., Ferguson, K. and Steele, C. (2013) 'Reliable modeling of chemical durability of high level waste glass using bootstrap aggregated neural networks', *2013 Ninth International Conference on Natural Computation (ICNC2013)*, pp. 178-183.

Kermani, B.G., Schiffman, S.S. and Nagle, H.T. (2005) 'Performance of the Levenberg-Marquardt neural network training method in electronic nose applications', *Sensors and Actuators B-Chemical*, 110(1), pp. 13-22.

Kresta, J.V., Macgregor, J.F. and Marlin, T.E. (1991) 'Multivariate statistical monitoring of process operating performance', *Canadian Journal of Chemical Engineering*, 69(1), pp. 35-47.

Lee, C.K., Choi, S.W. and Lee, I.B. (2004a) 'Sensor fault identification based on time-lagged PCA in dynamic processes', *Chemometrics and Intelligent Laboratory Systems*, 70(2), pp. 165-178.

Lee, J.-M., Yoo, C. and Lee, I.-B. (2004b) 'Fault detection of batch processes using multiway kernel principal component analysis', *Computers & Chemical Engineering*, 28(9), pp. 1837-1847.

Li, F., Zhang, J., Oko, E. and Wang, M. (2017a) 'Modelling of a post-combustion CO2 capture process using extreme learning machine', *International Journal of Coal Science & Technology*, 4(1), pp. 33-40.

Li, Q., Du, Q., Ba, W. and Shao, C. (2012) 'Multiple-input multiple-output soft sensors based on KPCA and MKLS-SVM for quality prediction in atmospheric distillation column', *International Journal of Innovative Computing, Information and Control*, 8(12), pp. 8215-8230.

Li, Q., Xie, M., Du, X. and Ba, W. (2017b) *2017 Chinese Automation Congress (CAC2017)*. 20-22 Oct. 2017.

Li, W., Wang, D. and Chai, T. (2015) 'Multisource data ensemble modeling for clinker free lime content estimate in rotary kiln sintering processes', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2), pp. 303-314.

Li, W., Zhuo, Y., Bao, J. and Shen, Y. (2021a) 'A data-based soft sensor approach to estimating raceway depth in ironmaking blast furnaces', *Powder Technology*, 390, pp. 529-538.

Li, W.T., Wang, D.H. and Chai, T.Y. (2013) 'Burning state recognition of rotary kiln using ELMs with heterogeneous features', *Neurocomputing*, 102, pp. 144-153.

Li, Z., Rehman, K.U., Wenhui, L. and Atique, F. (2021b) 'Soft sensor modeling method based on SPA-GWO-SVR for marine protease fermentation process', *Journal of Control Science and Engineering*, 2021, p. 6653503.

Liang, N.Y., Huang, G.B., Saratchandran, P. and Sundararajan, N. (2006) 'A fast and accurate online sequential learning algorithm for feedforward networks', *IEEE Transactions on Neural Networks*, 17(6), pp. 1411-1423.

Lillington, J.N.P., Goût, T.L., Harrison, M.T. and Farnan, I. (2020) 'Predicting radioactive waste glass dissolution with machine learning', *Journal of Non-Crystalline Solids*, 533, p. 119852.

Liu, Y. and Yao, X. (1999) 'Ensemble learning via negative correlation', *Neural Networks*, 12(10), pp. 1399-1404.

Macgregor, J.F. and Kourti, T. (1995) 'Statistical process-control of multivariate processes', *Control Engineering Practice*, 3(3), pp. 403-414.

Marquardt, D.W. (1963) 'An algorithm for least-squares estimation of nonlinear parameters', *Journal of the Society for Industrial and Applied Mathematics*, 11(2), pp. 431-441.

Mehranbod, N., Soroush, M. and Panjapornpon, C. (2005) 'A method of sensor fault detection and identification', *Journal of Process Control*, 15(3), pp. 321-339.

Mohammed, K.-J.R. and Zhang, J. (2013) 'Reliable optimisation control of a reactive polymer composite moulding process using ant colony optimisation and bootstrap aggregated neural networks', *Neural Computing and Applications*, 23(7), pp. 1891-1898.

Morris, J.B., Chidley, B.E. and Walmsley, D. (1983) 'Off-gas behavior in the Harvest pot vitrification process', *Radioact Waste Manage Nucl Fuel Cycle*, 3(3-4), pp. 347-370.

Murkherjee, A. and Zhang, J. (2008) 'A reliable multi-objective control strategy for batch processes based on bootstrap aggregated neural network models', *Journal of Process Control*, 18, pp. 720-734.

Niu, D.-p., Wang, F.-l., Zhang, L.-l., He, D.-k. and Jia, M.-x. (2011) 'Neural network ensemble modeling for nosiheptide fermentation process based on partial least squares regression', *Chemometrics and Intelligent Laboratory Systems*, 105(1), pp. 125-130.

Nomikos, P. and MacGregor, J.F. (1995) 'Multi-way partial least squares in monitoring batch processes', *Chemometrics and Intelligent Laboratory Systems*, 30(1), pp. 97-108.

Ojovan, M.I. and Lee, W.E. (2011) 'Glassy wasteforms for nuclear waste immobilization', *Metallurgical and Materials Transactions A*, 42(4), pp. 837-851.

Pascanu, R., Mikolov, T. and Bengio, Y. (2013) 'On the difficulty of training recurrent neural networks', *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. Atlanta, GA, USA. JMLR.org, pp. III–1310–III–1318.

Pearson, R.K. (2002) 'Outliers in process modeling and identification', *IEEE Transactions on Control Systems Technology*, 10(1), pp. 55-63.

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.

Qin, S.J. (2003) 'Statistical process monitoring: basics and beyond', *Journal of Chemometrics*, 17(8-9), pp. 480-502.

Qin, S.J. (2012) 'Survey on data-driven industrial process monitoring and diagnosis', *Annual Reviews in Control*, 36(2), pp. 220-234.

Qin, Y. and Zhao, C. (2019) 'Comprehensive process decomposition for closed-loop process monitoring with quality-relevant slow feature analysis', *Journal of Process Control*, 77, pp. 141-154.

Qin, Y., Zhao, C.H. and Huang, B. (2019) 'A new soft sensor algorithm with concurrent consideration of slowness and quality interpretation for dynamic chemical process', *Chemical Engineering Science*, 199, pp. 28-39.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) 'Learning representations by back-propagating errors', *Nature*, 323(6088), pp. 533-536.

Schapire, R.E. (1990) 'The strength of weak learnability', *Machine Learning*, 5(2), pp. 197-227.

Schölkopf, B., Smola, A. and Müller, K.-R. (1998) 'Nonlinear component analysis as a kernel eigenvalue problem', *Neural Computation*, 10(5), pp. 1299-1319.

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

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. (2016b) '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 (ACC2015),* 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.

Sprekeler, H., Zito, T. and Wiskott, L. (2014) 'An extension of slow feature analysis for nonlinear blind source separation', *Journal of Machine Learning Research*, 15, pp. 921-947.

Sun, W., Meng, Y., Palazoglu, A., Zhao, J., Zhang, H. and Zhang, J. (2011) 'A method for multiphase batch process monitoring based on auto phase identification', *Journal of Process Control*, 21(4), pp. 627-638.

Tang, Q., Li, D. and Xi, Y. (2018) 'A new active learning strategy for soft sensor modeling based on feature reconstruction and uncertainty evaluation', *Chemometrics and Intelligent Laboratory Systems*, 172, pp. 43-51.

Taylor, R.F. (1985) 'Chemical-engineering problems of radioactive-waste fixation by vitrification', *Chemical Engineering Science*, 40(4), pp. 541-569.

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.

Ustinov, O.A., Yakunin, S.A. and Smelova, T.V. (2019) 'Variant of off-gas cleanup in liquid radwaste vitrification', *Atomic Energy*, 127(2), pp. 105-108.

Utton, C.A., Swanton, S.W., Schofield, J., Hand, R.J., Clacher, A. and Hyatt, N.C. (2012) 'Chemical durability of vitrified wasteforms: effects of pH and solution composition', *Mineralogical Magazine*, 76(8), pp. 2919-2930.

Valentini, G. and Masulli, F. (2002) 'Ensembles of learning machines', *Proceedings of the 13th Italian Workshop on Neural Nets-Revised Papers*. Springer-Verlag, pp. 3–22.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. and Yin, K. (2003) 'A review of process fault detection and diagnosis Part III: Process history based methods', *Computers & Chemical Engineering*, 27(3), pp. 327-346.

Walczak, B. and Massart, D.L. (2001a) 'Dealing with missing data Part I', *Chemometrics and Intelligent Laboratory Systems*, 58(1), pp. 15-27.

Walczak, B. and Massart, D.L. (2001b) 'Dealing with missing data: Part II', *Chemometrics and Intelligent Laboratory Systems*, 58(1), pp. 29-42.

Wang, B., Shahzad, M., Zhu, X., Ur Rehman, K., Ashfaq, M. and Abubakar, M. (2020a) 'Soft sensor modeling for l-lysine fermentation process based on hybrid ICS-MLSSVM', *Scientific Reports*, 10(1), p. 11630.

Wang, G., Jia, Q.-S., Zhou, M., Bi, J. and Qiao, J. (2021a) 'Soft-sensing of wastewater treatment process via deep belief network with event-triggered learning', *Neurocomputing*, 436, pp. 103-113.

Wang, H., Bah, M.J. and Hammad, M. (2019) 'Progress in outlier detection techniques: A survey', *IEEE Access*, 7, pp. 107964-108000.

Wang, J. and Zhao, C. (2020) 'Variants of slow feature analysis framework for automatic detection and isolation of multiple oscillations in coupled control loops', *Computers & Chemical Engineering*, 141, p. 107029.

Wang, K., Chang, P. and Meng, F. (2021b) ‘Monitoring of wastewater treatment process based on slow feature analysis variational autoencoder’, *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS2021)*. 14-16 May 2021.

Wang, X., Kruger, U. and Irwin, G.W. (2005) 'Process monitoring approach using fast moving window PCA', *Industrial & Engineering Chemistry Research*, 44(15), pp. 5691-5702.

Wang, Z., Zheng, Y. and Wong, D.S.-H. (2020b) 'Trajectory-based operation monitoring of transition procedure in multimode process', *Journal of Process Control*, 96, pp. 67-81.

Willis, M.J., Di Massimo, C., Montague, G.A., Tham, M.T. and Morris, A.J. (1991) 'Inferential measurement via artificial neural networks', Proceedings of IFAC Symposium on Intelligent Tuning and Adaptive Control, 15-17 January 1991, Singapore, pp. 85-90.

Wiskott, L. and Sejnowski, T.J. (2002) 'Slow feature analysis: Unsupervised learning of invariances', *Neural Computation*, 14(4), pp. 715-770.

Wold, S., Ruhe, A., Wold, H. and Dunn, W.J. (1984) 'The collinearity problem in linear-regression - the partial least-squares (PLS) approach to generalized inverses', *SIAM Journal on Scientific and Statistical Computing*, 5(3), pp. 735-743.

Xibilia, M.G., Latino, M., Marinković, Z., Atanasković, A. and Donato, N. (2020) 'Soft sensors based on deep neural networks for applications in security and safety', *IEEE Transactions on Instrumentation and Measurement*, 69(10), pp. 7869-7876.

Xie, W., Wang, J.S., Xing, C., Guo, S.S., Guo, M.W. and Zhu, L.F. (2020) 'Extreme learning machine soft sensor model with different activation functions on grinding process optimized by improved black hole algorithm', *IEEE Access*, 8, pp. 25084-25110.

Xu, X. and Ding, J. (2021) 'Decentralized dynamic process monitoring based on manifold regularized slow feature analysis', *Journal of Process Control*, 98, pp. 79-91.

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.

Yoo, C.K., Lee, J.-M., Vanrolleghem, P.A. and Lee, I.-B. (2004) 'On-line monitoring of batch processes using multiway independent component analysis', *Chemometrics and Intelligent Laboratory Systems*, 71(2), pp. 151-163.

Yu, W.K. and Zhao, C.H. (2019) 'Recursive exponential slow feature analysis for fine-scale adaptive processes monitoring with comprehensive operation status identification', *IEEE Transactions on Industrial Informatics*, 15(6), pp. 3311-3323.

Yuan, X., Li, L. and Wang, Y. (2020a) 'Nonlinear dynamic soft sensor modeling with supervised long short-term memory network’, *IEEE Transactions on Industrial Informatics*, 16(5), pp. 3168-3176.

Yuan, X., Zhou, J. and Wang, Y. (2020b) 'Locally weighted slow feature regression for nonlinear dynamic soft sensor modeling and its application to an industrial hydrocracking process', *Measurement Science and Technology*, 31(5).

Yuan, X.F., Ge, Z.Q. and Song, Z.H. (2014) 'Locally weighted kernel principal pomponent regression model for soft sensing of nonlinear time-variant processes', *Industrial & Engineering Chemistry Research*, 53(35), pp. 13736-13749.

Zamprogna, E., Barolo, M. and Seborg, D.E. (2004) 'Estimating product composition profiles in batch distillation via partial least squares regression', *Control Engineering Practice*, 12(7), pp. 917-929.

Zhang, A.-H., Zhu, K.-Y., Zhuang, X.-Y., Liao, L.-X., Huang, S.-Y., Yao, C.-Y. and Fang, B.-S. (2020a) 'A robust soft sensor to monitor 1,3-propanediol fermentation process by Clostridium butyricum based on artificial neural network', *Biotechnology and Bioengineering*, 117(11), pp. 3345-3355.

Zhang, B., Han, Y., Yu, B. and Geng, Z. (2020b) 'Novel monlinear autoregression with external input integrating PCA-WD and its application to a dynamic soft sensor', *Industrial & Engineering Chemistry Research*, 59(35), pp. 15697-15706.

Zhang, H., Deng, X., Zhang, Y., Hou, C. and Li, C. (2021) 'Dynamic nonlinear batch process fault detection and identification based on two-directional dynamic kernel slow feature analysis', *The Canadian Journal of Chemical Engineering*, 99(1), pp. 306-333.

Zhang, H., Tian, X. and Cai, L. (2015a) 'Nonlinear process fault diagnosis using kernel slow feature discriminant analysis', *IFAC-PapersOnLine*, 48(21), pp. 607-612.

Zhang, H., Tian, X. and Deng, X. (2017) 'Batch process monitoring based on multiway global preserving kernel slow feature analysis', *IEEE Access*, 5, pp. 2696-2710.

Zhang, H., Tian, X., Deng, X. and Cao, Y. (2018a) 'Batch process fault detection and identification based on discriminant global preserving kernel slow feature analysis', *ISA Transactions*, 79, pp. 108-126.

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

Zhang, J. (2004) 'A reliable neural network model based optimal control strategy for a batch polymerization reactor', *Industrial & Engineering Chemistry Research*, 43(4), pp. 1030-1038.

Zhang, J. (2006) 'Offset-free inferential feedback control of distillation compositions based on PCR and PLS models', *Chemical Engineering & Technology*, 29(5), pp. 560-566.

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.

Zhang, J. and Morris, A.J. (1998) 'A sequential learning approach for single hidden layer neural networks', *Neural Networks*, 11(1), pp. 65-80.

Zhang, J. and Morris, A.J. (1999) 'Recurrent neuro-fuzzy networks for nonlinear process modeling', *IEEE Transactions on Neural Networks*, 10(2), pp. 313-326.

Zhang, J., Morris, A.J. and Martin, E.B. (1998) 'Long-term prediction models based on mixed order locally recurrent neural networks', *Computers & Chemical Engineering*, 22(7-8), pp. 1051-1063.

Zhang, N., Tian, X., Cai, L. and Deng, X. (2015b) 'Process fault detection based on dynamic kernel slow feature analysis', *Computers & Electrical Engineering*, 41, pp. 9-17.

Zhang, S., Zhao, C. and Huang, B. (2019) 'Simultaneous static and dynamic analysis for fine-scale identification of process operation statuses', *IEEE Transactions on Industrial Informatics*, 15(9), pp. 5320-5329.

Zhang, S.M. and Zhao, C.H. (2019) 'Slow-feature-analysis-based batch process monitoring with comprehensive interpretation of operation condition deviation and dynamic anomaly', *IEEE Transactions on Industrial Electronics*, 66(5), pp. 3773-3783.

Zhang, X., Zhu, Q., Jiang, Z.-Y., He, Y. and Xu, Y. (2018b) 'A novel ensemble model using PLSR integrated with multiple activation functions based ELM: Applications to soft sensor development', *Chemometrics and Intelligent Laboratory Systems*, 183, pp. 147-157.

Zhang, Y.W. and Zhang, P.C. (2011) 'Optimization of nonlinear process based on sequential extreme learning machine', *Chemical Engineering Science*, 66(20), pp. 4702-4710.

Zhao, C.H. and Huang, B. (2018) 'A full-condition monitoring method for nonstationary dynamic chemical processes with cointegration and slow feature analysis’, *AIChE Journal*, 64(5), pp. 1662-1681.

Zheng, J. and Zhao, C. (2019) 'Online monitoring of performance variations and process dynamic anomalies with performance-relevant full decomposition of slow feature analysis', *Journal of Process Control*, 80, pp. 89-102.

Zhou, C., Liu, Q., Huang, D. and Zhang, J. (2012) 'Inferential estimation of kerosene dry point in refineries with varying crudes', *Journal of Process Control*, 22(6), pp. 1122-1126.

Zhou, Z.-H., Wu, J. and Tang, W. (2002) 'Ensembling neural networks: Many could be better than all', *Artificial Intelligence*, 137(1), pp. 239-263.

Zhu, J., Ge, Z., Song, Z. and Gao, F. (2018) 'Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data', *Annual Reviews in Control*, 46, pp. 107-133.