

UNIVERSITY OF WITWATERSRAND

RESEARCH REPORT

Predicting Particle Fineness in a Cement Mill

Author:

Rowan LANGE

Supervisors:

Prof. Anton VAN WYK,

Dr. Terence VAN ZYL

WITS
UNIVERSITY

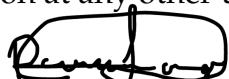


May 28, 2020

Declaration of Authorship

I, Rowan LANGE, declare that this research report titled, "Predicting Particle Finess in a Cement Mill" is my own, unaided, work except where otherwise acknowledged. It is being submitted fro the degree of Master of Science at the University of Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university.

Signed:



Date: 28/05/2020

UNIVERSITY OF WITWATERSRAND

Abstract

Faculty of Science
School of Computer Science and Applied Mathematics

Msc

Predicting Particle Fineness in a Cement Mill

by Rowan LANGE

Cement production is a multi-billion dollar industry, of which one of the main subprocesses, cement milling, is complex and non-linear. There is a need to model the fineness of particles exiting the milling circuit in order to better control the cement plant. This paper explores the relationship between the particle size of cement produced and various sensor readings from the cement mill circuit. The aim of this paper is to provide a model for predicting the fineness of particles exiting the milling circuit using data on the current and past states of the plant. A comprehensive literature review of the problem as well as a discussion of potential modelling solutions are provided. Blaine (particle fineness) is modelled using many different linear and non-linear models on 5 months of data from a large cement plant. On a holdout test set a multi-layered perceptron achieved a MAE of 8.799 and a linear regression achieved an R^2 of 0.481. Discussion of the significance of various features for predicting Blaine is also presented. The results show some success from non-linear data-driven models and highlight the unique difficulties in modelling the cement mill. Finally, recommendations are presented for future research.

Contents

Declaration of Authorship	i
Abstract	ii
1 Introduction	1
1.1 The importance of a soft-sensor for a cement mill	1
1.2 Methodology	2
1.3 Research aims and objectives	3
1.4 Limitations and assumptions of research	3
1.5 Description of process	4
1.6 Outline of paper	5
2 Literature Review	6
2.1 The importance of predictive models for control systems	6
2.2 First principle models of a cement mill circuit	7
Summary	10
2.3 Data Science solutions for a variety of problems	10
2.4 Soft-sensors for grinding circuits	12
2.4.1 Soft-sensors for mills in control literature	12
2.4.2 The evolution of soft-sensors for mills	13
2.4.3 Research aim	15
2.5 Data Science models	15
2.5.1 Linear models	16
2.5.2 Kernel SVR	16
2.5.3 Neural network type models	17
Multilayer perceptron (MLP)	17
A long short-term memory (LSTM)	18
Adaptive neuro-fuzzy inference engine (ANFIS)	19
2.5.4 Regularization	19
2.5.5 Optimisation algorithm	21
2.5.6 Optimisation for kernel SVR	23

3 Methodology	24
3.1 Data and preprocessing	24
3.1.1 Description of plant	24
3.2 Data preprocessing	28
3.2.1 Filtering out periods of non-operation	28
Filtering out noise	29
3.2.2 Including past plant conditions	31
3.2.3 Training, validation and test split	33
3.3 Experiment description	34
3.3.1 Grid search	35
3.3.2 Feature selection	35
3.3.3 Measures of accuracy	36
3.3.4 Tested models	36
Persistence model (1)	37
Linear models (2-6)	37
SVR (8-9)	37
MLP (9-10)	38
LSTM (11)	38
ANFIS (12)	38
4 Results and discussion	40
4.1 Modelling results	40
4.2 Plant information used by models	42
4.2.1 Feature importance	46
4.2.2 Training a model without using past fineness measurements	48
4.3 Summary	48
5 Conclusions and recommendations	51
References	53

List of Figures

1.1	Milling circuit. Black arrows represent product flow and blue arrows represents airflow.	4
2.1	An illustration of a the LSTM neural network architecture proposed by Hocheiter & Schnidhuber [32]	18
2.2	An illustration of a the ANFIS neural network architecture proposed by Jang. [35]	20
3.1	Milling circuit. Black arrows represent product flow and blue arrows represents airflow. Orange boxes represent controlled variables, blue boxes represent measured variables.	26
3.2	A linear correlation matrix heat map for process variables in the plant	27
3.3	Plot of mill amps, Feed and Blaine values over the course of data capture	29
3.4	Plot showing the relative frequency of different observed values for mill current draw (MAMPS)	29
3.5	Plot showing effect of exponential smoothing applied to the variable EAMPS for random extract of data.	31
3.6	Plot showing cross correlation between RPEAMPS and EAMPS, with the maximum correlation at a lag of -29 time steps highlighted in red	31
3.7	Plot showing average temporal relationship between process variables as would be suggested by cross correlation calculations	32
3.8	Illustration of how data was split for the standard and online simulated models	36
4.1	Various plots showing model performance on the train, validation and test sets for the optimal deep MLP model.	43
4.2	Repeat of test set plot given in Figure 4.1 with periods of non-operation removed	44
4.3	Plot of Residual vs Blaine for all observations, the observation number is directly related to time	45
4.4	Various plots showing model performance on the train, validation and test sets for a deep MLP model.	49

List of Tables

2.1	Table showing various loss functions for different regression algorithms	16
3.1	Description of captured information in cement circuit given by Figure 3.1	25
4.1	Performance for various models	40
4.2	Accuracies of lasso regression variables as final few features are eliminated	44
4.3	Average importance of features across predictive models, 1 is most important	47
4.4	Accuracies of models trained on data set with no past values of Blaine or Residual	48

List of Abbreviations

MPC	Model based Predictive Control
PPS	Product Particle Size
PSD	Particle Size Distribution
RBF	Radial Basis Function
SVR	Support Vector Regression
ANN	Artificial Neural Network
MLP	MultiLayered Perceptron
RBFNN	Radial Basis Function Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory recurrent neural network
ANFIS	Adaptive Neuro-Fuzzy Inference Engine

Chapter 1

Introduction

1.1 The importance of a soft-sensor for a cement mill

As economies grow, so too does the construction of apartments, highways, offices and, as a direct result, the demand for cement. The cement production industry was estimated to be worth \$450 billion in 2015 [1]. The large footprint of the industry extends beyond the economic domain as cement production contributes about 7% of global anthropogenic CO₂ emissions [2].

At a high level, a cement plant would be considered to be running well if the cement is produced to meet quality standards, whilst process costs, such as electricity consumption, is kept to a minimum. By virtue of there being fixed costs, the cost per kilogram of cement decreases if plant downtime is minimised. There are, however, also secondary goals for plant operators such as minimised emissions as well as the flexibility to change product standards. For example, the plant manager may need to produce a quicker setting cement, or a stronger cement. Ultimately, all of these considerations show how there are large financial benefits in controlling the plant in an efficient and safe manner.

Cement production involves, three steps. Firstly, raw materials such as limestone and sand (or volcanic ash) are put through a raw mill and are blended, secondly the ground raw materials pass through a high-temperature rotary kiln to produce the intermediate product: clinker [3]. Finally, the clinker, with the addition of some other ingredients like gypsum and fly-ash, passes through the cement mill (also called Finish Mill), creating the final product: cement [3].

This research focuses on predicting the final product particle size (PPS) for the cement mill. There is a well researched relationship between the fineness of cement and various characteristics such as its strength [4]. As a result a cement plant would want to ensure the cement produced consistently meets fineness standards.

Controlling a mill circuit can be very difficult. The process is non-linear and non-stationary and many important variables, such as incoming clinker particles sizes and hardness, are not known to operators [5]. Another major difficulty with

controlling the processes is that the quality metric (cement fineness) is not measured online. Cement is sampled from the production-line at some regular interval, e.g. hourly, and sent through to a lab with results on the product fineness coming back some time later. Therefore, during operation of the plant, cement fineness data is out of date by at least as much time as it takes to perform lab tests with additional large gaps in between taking samples. Minchala et al. [6] reported lab tests as occurring every 2 hours which is typical within the industry.

Creating a reliable model for the cement PPS based on plant conditions has several major benefits. Firstly, it provides a real-time estimate for any controller, including an engineer, on the state of the PPS. This information would provide a more comprehensive understanding of the state of the plant, the evolution of this state over time as well as a check on recently reported values which might be subject to sampling and measurement error. Secondly, this model would build towards a better understanding of the plant dynamics and the effect of control variables on PPS, which could aid in the development of an automated controller. This is particularly valuable as cement plant dynamics are not very well understood well enough for accurate predictive control to be widespread [6, 7].

For the cement circuit analysed in this research report, there appears to be no existing literature on potential soft-sensor techniques and the related accuracies of those techniques. Furthermore, there appears to be limited literature, in general, that investigates the accuracy of data-driven soft-sensors for actual industrial milling circuits.

1.2 Methodology

There are many different approaches that could be used to build a soft-sensor including first-principle models, machine learning and statistical techniques such as time series models. This research report only looks at machine learning techniques. First principle models are difficult to implement due to limitations in sensors and data capture at a plant [7]. Statistical techniques, generally are not designed to take advantage of the vast number of different features for which data is captured at the cement plant and have been shown to be less effective for particle size soft-sensors [8]. Machine learning techniques such as regression and neural networks have thus come to dominate most similar research [9–12]. Furthermore many machine learning regression techniques such as deep neural networks and Long Short-Term Memory recurrent neural networks have been achieving state of the art performance on many other problems and do not seem to have been explored in mill-circuit soft-sensor literature[13, 14].

1.3 Research aims and objectives

The aim of this research is to recommend a method for building an accurate, reliable cement fineness soft-sensor. This aim is achieved by completing the following objectives:

- Investigate literature both for grinding circuits and other regression problems in order to propose a range of methods for building the soft-sensor.
- Report on the performance of various modelling techniques.
- Investigate difficulties in designing a soft-sensor for the real-world data to gain insight into problems that might invalidate soft-sensors.

1.4 Limitations and assumptions of research

The Scope of the research is limited in the following ways:

- An underlying assumption behind the modelling approach in this paper is that given sufficient data, a predictive algorithm can learn reliable time invariant relationships in the process. However, as is shown in Chapter 4, the grinding process shows evidence of significant time-variance and the 5 months of data used for this research may not be reflective of future operating conditions. Furthermore, system dynamics can change dramatically under closed loop control, and incorporation of the soft-sensor into a control system may change plant dynamics and invalidate the soft-sensor accuracy. Models that are trained online are theoretically more adaptive to time variant systems and results in Chapter 4 offer empirical support to this conclusion as well.
- Cement circuits can differ substantially by design and the best model or modelling technique for one circuit may not be the best for another.
- An investigation was done on the temporal relationships between different process variables (see Subsection 3.2.2) which was used to inform the decision to include process data with a lag of 8 minutes in the regression. However, better regression results might be possible through a more detailed analysis for selecting different lag values for each feature.
- Lab measurements for the variable of interest are sparse in the data and no methods were used to interpolate or synthesise additional observations. Therefore, the trained models generally are predicting Blaine at one hour intervals.

Modelling results may differ if more frequent observations, or synthetic observations are used.

1.5 Description of process

Cement milling circuits can differ significantly in implementation, utilising different mills, separators, pre-crushers, etc. The circuit analysed by Minchala et al. [6] differs significantly from the circuit analysed by Pani [11] which is, in turn, very different from the circuit analysed in this research report.

A diagram of the circuit for the mill analysed in this research report is given in Figure 1.1.

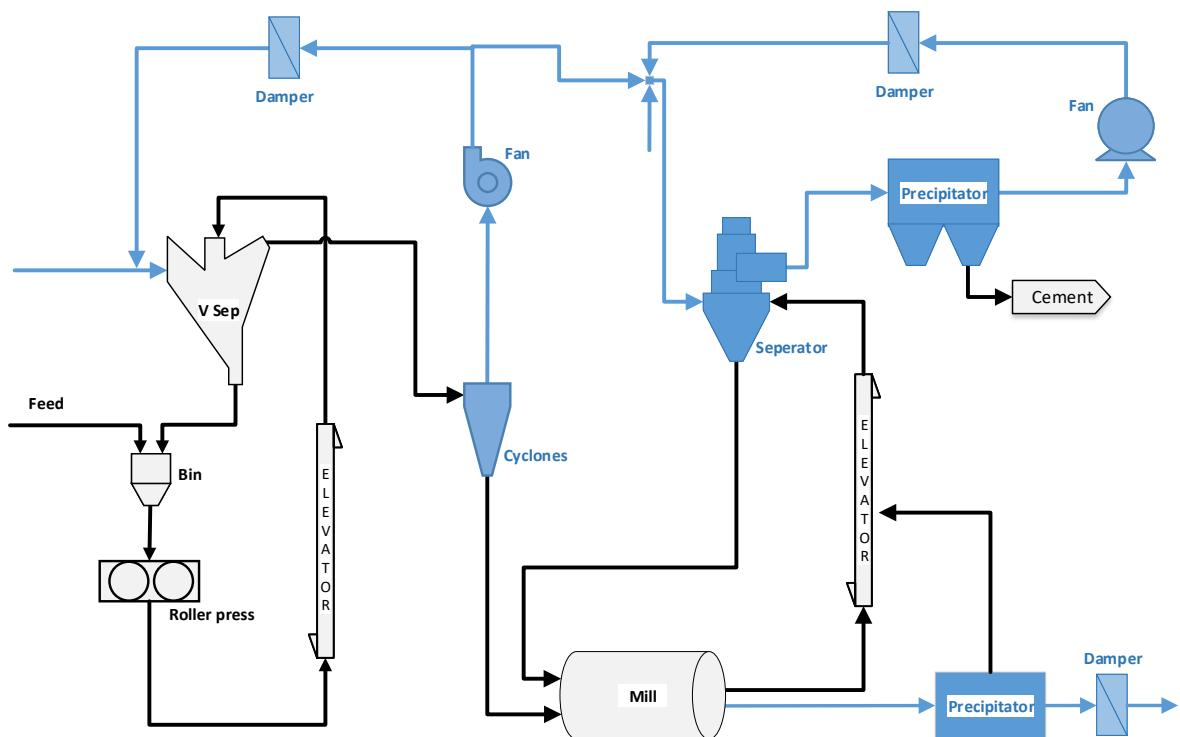


FIGURE 1.1: Milling circuit. Black arrows represent product flow and blue arrows represents airflow.

At a high level, the circuit has two significant sub-circuits, a pre-crusher sub-circuits and a main mill sub-circuits. At the start of the process clinker from the kiln is carried into a collection bin which feeds the rocks into a roller press. After being crushed in the roller press the particles are lifted by an elevator into a V-Separator which is a passive separator. As particles fall through the V-Separator lighter particles and cement dust are picked up by a pressurised airflow and taken to a cyclone which collects the coarse dust to be fed into the main mill. The main mill is a huge rotating steel drum where steel balls make high momentum impacts

with clinker rocks fracturing them into smaller particles. The movement of particles through the mill is assisted by pressurised air.

The particles leaving the mill are carried by an elevator into the plant's main separator which uses centrifugal forces and pressurised airflow to separate cement powder from coarser particles which are to be fed back into the main mill.

Another significant feature is the airflow from the V-Separator that passes through the cyclones and contains fine cement dust. This flow is split with part of the air flowing into the main separator and the rest flowing back into the V-Separator. One might expect that all of this airflow would be sent to the main separator as it contains cement dust that might be ready for final product. The reason that this is not the case and that some of the dusty airflow is fed back into the V-Separator is because plant operators would like to minimise the amount of fresh air that is sucked into the V-Separator.

1.6 Outline of paper

Chapter 2 provides a literature review that describes the difficulties in implementing a first-principles solution and then outlines data-driven regression techniques. The methodology is set out in Chapter 3 including a description of the data used in the modelling exercise. Chapter 4 reports on, and discusses, the results of the models. And finally, conclusions and recommendations are presented in Chapter 5.

Chapter 2

Literature Review

2.1 The importance of predictive models for control systems

Research into modelling cement-mill circuits models can be a vital step in the process of optimization and control. Over the last few decades industrial control has shifted towards model-based predictive control (MPC) in order to implement more efficient control systems [5]. At the heart of a MPC solution is a predictive model of the process. This model is used to calculate the most effective course of action to transition the plant into a desired state and maintain it there. An MPC solution requires a sufficiently accurate model of the underlying grinding dynamics [5]. Without a sufficiently accurate model, a MPC system could be unstable, potentially even navigating the plant into an unsafe state [10]. Therefore, creating a robust model of a cement mill circuit is an invaluable part of the control process.

The state of the industry appears to be a mix of low level automated controllers such as PIDs with high level control being performed by human controllers. Minchala et al. [6] compare their solution to manual control suggesting that, as recently as 2018, the industrial cement mill, that they implemented their system at, was using manual control at a high-level. In 2015, Dai et al. [10] state categorically that haematite processing plants still rely on human operators who at times use subjective evaluations of the state of the plant (such as the sound of the mill or the feel of the slurry) to control the plant.

The lack of fully-automated controllers might be the result of either from a lack of practical solutions or a limited understanding of the process in general, both of which motivate for further research into cement mill modelling, particularly models that could form part of a MPC system.

2.2 First principle models of a cement mill circuit

This section explores some mathematical models that have been proposed for modelling grinding circuits and how they run into practical difficulties. The biggest problem is that many of the variables discussed below for a first principles model of the cement grinding circuit are not measured online, such as the particle size distribution at various stages of the circuit. This means that any model/control system based off these mathematical models would have to make assumptions about key variables in the grinding process. Furthermore, keeping the model accurate may require costly experiments for re-parametrisation. The following discussion of cement mill modelling is based on the mill circuit given in Figure 1.1.

A common approach to the modelling of comminution processes, in general, is to create discrete bins for various particle sizes and to model the actual particle size distribution (PSD¹) through the circuit over time [5]. By analogising the discrete-state-space PSD to a discrete-state-space stochastic process, the change in particle sizes over time can be modelled using Markov chains [5, 7].

Working backwards from the final product, the first consideration would be the separator. The separator takes in a flow of ground product and produces a final product flow (cement) and a reject flow. Notably, one would expect the PSD of the final product exiting the separator to be affected by four things:

1. Separator speed: if the separator is spinning faster then the centrifugal forces are expected to push larger particles into the final product, resulting in a coarser cement.
2. Separator airflow: one would expect that a greater speed of airflow would also result in a coarser cement as airflow forces are more likely to carry large particles to the exterior of the centrifuge.
3. Incoming product mass flow rate: if the separator is being overloaded then one would expect the separator to operate less efficiency.
4. Incoming product PSD: A separator is imperfect and can only be considered to sort particles of various sizes into the final product and reject flows with certain probabilities. If the particles fed into the separator are more coarse, both the reject flow and final product are likely to become more coarse.

¹In literature the Product Particle Size (PPS) is referred to by different names such as particle size distribution (PSD), particle size measurement (PSM) and grinding particle size (GPS); but they all refer to roughly the same thing. It is unlikely to have a measurement for the full particle size distribution and therefore, in this paper, PSD refers to the theoretical notion of having a distribution of the particle sizes rather than a representative scalar as is more commonly measured at plants

A common mathematical approach involves deriving an efficiency curve as explained by Boulvin et al. [7]. Essentially this curve shows the probability of particles at various sizes being classified as final product. The control system designed by Minchala et al. [6] adopts a simplified version this system. Their application of the separation curve is dependent only on separator speed and airflow and ignores the effect that incoming mass flow might have on the separation curve. Empirically, Boulvin et al. [7] found that the mill flow rate had a significant non-linear effect on the separation curve casting doubt on the robustness of a simplified separator model as applied by Minchala et al. [6].

There are two flows into the separator, one directly from the pre-crusher circuit and one from the ball mill which has even more complicated dynamics, discussed next.

A simplified mathematical balance model for a ball mill was applied by Minchala et al. [6]. They used an assumption that the grinding process is consistent along the length of the mill which is common in other research [5]. Letting the PSD vector of a batch of particles at time t be \mathbf{f}_t and letting \mathbf{G} be the grinding function this model proposes that $\mathbf{f}_{t+1} = \mathbf{G}\mathbf{f}_t$. Per unit time, the grinding matrix is given as follows:

$$\mathbf{G} = \begin{bmatrix} 1 & s_2 & s_3 & \dots & s_N/(N-1) \\ 0 & 1-s_2 & s_3/2 & \dots & s_N/(N-1) \\ 0 & 0 & 1-s_3 & \dots & s_N/(N-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1-s_N \end{bmatrix}$$

In this example s_i represents the rate at which a particle in bin i is expected to fracturing per unit time and N is the number of bins for which different particle sizes have been represented. The fraction denominator results from assuming that, given a particular particle fragments, the resulting mass of smaller particles is expected to occupy all smaller sized bins with equal probability. This assumption was made in the system proposed by Minchala et al. [6]. The zeros in the matrix represent the fact that particles cannot becomes larger whilst being crushed.

After k periods of time in the mill one would expect the distribution of particles (\mathbf{f}) to become $\mathbf{f}_k = \mathbf{G}^k \mathbf{f}_0$. The residence time of a mill is the time for a particle that has just entered a mill to exit on the other side. If we assume that the residence time is constant for particles of all sizes in the mill and that for a given mill this is K time steps, then for an initial population of particles entering the mill (\mathbf{f}_0) we expect this exact same population to exit the mill after K units of time as $\mathbf{f}_K = \mathbf{G}^K \mathbf{f}_0$.

This grinding model makes simplifying assumptions relative to the general model of particles in a mill given by [7]:

$$\frac{\partial(Hm_i)}{\partial t} = -u_i \frac{\partial}{\partial x}(Hm_i) + D_i \frac{\partial^2}{\partial x^2}(Hm_i) - s_i Hm_i + \sum_{j=1}^{i-1} b_{ij} s_j Hm_j, \quad 1 \leq i \leq N$$

In the above model $H(x, t)$ represents the amount of materials at mill axial (length) position x and time t . Furthermore $m_i(x, t)$ is the fraction of that material that is of size i . The term s_i is defined as above and b_{ij} represents the proportion of particles that fracture into size i from a particle of size j given the particle has fractured. The term b_{ij} , thus, generalises the earlier discussed assumption of uniform fracture probabilities for all of the smaller bins. The terms u_i and D_i are convection and diffusion velocity coefficients respectively which allow for variable movements along the length of the mill for particles of different sizes.

The need for a more generalised model that does not assume constant residence time for all particles is best understood through a narrative example. If the mill is operating normally and suddenly particles are fed in with a much lower PSD, these particles are likely to be more rapidly pushed by airflow through to the output end of the mill. This change to mill feed would result in an increase of the mass flow and decrease in the PSD of the particles exiting the mill before the constant residence time model would allow for.

Another concern is how to set values for all parameters s_i and b_{ij} . Boulvin et al. [7] points out that other researchers have developed models for parametrising b_{ij} and s_i but state that these models are difficult to parametrise using lab experiments and require empirical estimation from industrial data. Additionally, these parameters would change depending on the hardness of the clinker or the load in the mill. Further questions then arise as to how frequently experiments would need to be run to re-parametrise these models.

The general grinding process would be expected to change over time depending on factors such as:

1. the PSD of clinker entering the mill,
2. the hardness of the clinker entering the mill,
3. the airflow through the mill,
4. the rotary speed of the mill,
5. the mass load within the mill, and
6. the state of wear of the balls in the mill.

A similar modelling exercise would need to be carried out on the pre-crusher circuit shown in Figure 1.1, that consists of the roller-press and V-Separator. A more complete model of the whole system would also need to account for the dust flows throughout the circuit.

Summary

The purpose of this section was to introduce the key systems considerations within a grinding circuit as well as to highlight the difficulties for modelling and control using first principles models. The largest obstacle is that these types of models require information on variables that are not measured online, most notably the PSD of the product at various stages of the process. Therefore, in practice, parametrisation of models require dedicated experimentation at the plant (as was done by Minchala et al. [6]). Furthermore, the models are highly non-linear and these parametrisations might become invalidated as plant conditions change and unmeasured disturbances occur. It is unclear how robust the models are to these changes and how frequently the costly re-parametrisation experiments will need to be performed. Certainly, Minchala et al. [6] highlight the limited plant conditions for which their model is valid.

2.3 Data Science solutions for a variety of problems

This section explains a data driven approach to modelling a system in contrast to the first principles approach described in Section 2.2. The differences are not binary, and much of the content that follows is generalisations. The Data Science approach could be thought of as deep neural networks whereas the physical systems approach could be thought of as parsimonious first-principle models of differential equations of fundamental variables in a system.

The 21st century has seen many problems being solved by advanced pattern recognition and adaptive computation algorithms: i.e. machine learning [15]. Growth in the volume of recorded data and an increases in computational hardware have worked in tandem with the ever expanding toolbox of machine learning algorithms to allow state of the art performance in a variety of problems. This has been most notable in the fields of image processing [16] and natural-language processing [14] with significant progress also being demonstrated for control, perhaps best illustrated by the resurgent research focus on self-driving vehicles [17].

The implementations vary widely but there are common threads in how the Data Science approach differs from classical approaches. Firstly, there is less need for domain specific knowledge of underlying systems. Pattern recognition algorithms are relied on to extract data regularities and provide a high performance solution by black box standards [15]. Secondly, the focus of research shifts from building an understanding of the underlying system to designing and implementing general purpose algorithms [15]. Implementing models involves selection of appropriate

meta-models e.g. neural network architecture, implementation of parameter optimisation algorithms, fine tuning of hyper-parameters and management of over-fitting. As an illustrative example of the approach, Bojarski et al. [17] train an end-to-end self-driving car algorithm using convolutional neural networks without any explicit programming of, or learning for, sub-goals like detecting road lanes.

Given the scope of mappings permitted by a neural network, often a parsimonious physical systems model is contained in the hypothesis space of the neural network. Yet if the system is well understood a first-principles model might still be more effective. The advantage, therefore, of a physical systems model is that it does not have the additional parameters of a neural network that might be used to over-fit a model to noise in the data set.

The main advantage of a neural network approach is the flexibility at capturing highly complex, unintuitive patterns that exists in real-world systems that are too convoluted to be adequately described or parametrised using a physical systems model. For example, convolutional neural networks are the most effective way of recognising cats in images². It would be very difficult to directly design an accurate model of cats in images for the purpose of classification due to variation in background, illumination, rotation, occlusion, deformation (cats are very flexible animals) as well as variation in cats themselves.

Due to over-fitting problems, current data-driven models tend to require very large amount of data to be successful³. One of the main reasons why neural networks have dominated research literature, recently, is that there are more big datasets available [15].

The decision was made for this research to take a data driven approach for several reasons. Firstly, data-driven predictive models of physical systems are achieving widespread state of the art success in several domains such as weather [20] and traffic [21]. Secondly, in the case of the cement mill circuit, data-driven approach can take advantage of large volumes of available plant data where the data required for modelling the physical system is not captured. For example, there exists months of plant data on changes in the power draw of the mill and main elevator in the circuit but no data on PSD throughout the process which would be expensive to collect. Thirdly, as described above, the cement mill system is complex and non-linear and literature suggests that lab models cannot be robustly fit to actual plants. On the

²There is no explicit research on the best cat detection algorithm, however a recent algorithm achieving state of the art performance for object detection (on datasets that include cats) is proposed by Ren et al. [18] and uses a convolutional neural network based architecture.

³For an interesting discussion of why machine learning requires troves of data vs human learning and how this might change see [19].

other hand, there is potential for all of these fundamental elements to show statistically significant second order effects that a data-driven model might discover.

The power of many neural networks is their ability to recognise patterns that humans may not be able to intuitively program. For example, there is a deep history of natural language processing research that explores grammar, syntax and semantics and yet many recent advances in the field such as Word2Vec [22] and image captioning models [23] are achieved using black box modelling approaches on large text databases.

In summary, this research is motivated by these successful applications of data-driven models to a variety of problems and attempts to determine if similar data-driven models can recognise important patterns in a cement mill circuit to produce a reliable soft-sensor for PPS.

2.4 Soft-sensors for grinding circuits

2.4.1 Soft-sensors for mills in control literature

Although there are limited examples of data driven control systems applied to cement mill circuits there are examples of lab-tested data-driven control systems for other types of mills. Dai et al. [10] designed a multi-level control system for a haematite grinding process, which does low level control and also features two pre-trained neural networks in series for dynamic optimisation of set-points based on plant conditions. The authors also highlight the importance of having an online estimate for PPS built using some non-linear mapping and include a neural network in their control system for this purpose.

Unfortunately, the accuracy of the soft-sensor designed by Dai et al. [10] is ambiguous for several reasons. Firstly, they report on root mean squared error (RMSE) but do not offer any means of contextualising performance such as would be achieved by reporting on the total variation for the data or the accuracy of a persistence model. Secondly, they mention a validation data set but they also mention a process of selecting the size of the hidden layer by trial and error and so it is not clear whether hyper-parameter selection or even weight initialization may have been based on the prediction results on the validation set which compromises the accuracy that can be inferred from the validation set results.

Zhou et al. [9] also build a neural network soft-sensor into their data-driven wet-mill grinding circuit control system. They use a Radial basis Function neural network (RBFNN) due to the capacity of the algorithm to learn non-linear patterns

in the data. They do not report on the accuracy of their soft-sensor, they only report on the improvements from the control system as a whole.

2.4.2 The evolution of soft-sensors for mills

Similar to control systems, there is little research on soft-sensors applied to cement milling circuits or dry milling circuits, more generally. However, there is a body of literature dedicated to particle size soft-sensors for wet-mills. Early soft-sensors used autoregressive, moving average with exogenous variables (ARMAX) type models [8]. ARMAX type models are argued by Zhou et al. [9] to be less robust at handling the non-linear grinding process. As early as 1997, Du et al. [12] showed that neural networks might lead to better predictions. There have been many applications of neural network based soft-sensors for PPS estimates in wet mills [9, 10, 12]. Other techniques have also been proposed for creating a soft-sensor for wet mills such as the approach taken by Zhou et al. [24] which is built around a kNN search.

Pani [11] has several publications and conference proceedings on data-driven soft-sensors designed for a cement mill which form part of his PhD thesis. He uses the same dataset for a cement plant. The circuit modelled by Pani [11] is different to the one modelled in this research report and instead involves a vertical roller mill with no pre-crusher circuit. Furthermore, only three process variables were considered by Pani [11] and only a snapshot of the plant was used for prediction, with no lagged values corresponding to past plant states being used for prediction. He reports on the results of linear regression, support vector regression with an radial basis function (RBF) kernel, various neural networks and an adaptive neuro-fuzzy inference system (ANFIS) of which the ANFIS was reported to have performed best despite having a lower R^2 than a support vector regression model and a neural network model. Pani [25] justifies his conclusion by arguing that an absolute error metric is a better metric than a squared error citing Willmott & Matsuura [26] to justify his decision. Willmott & Matsuura [26] argues that mean absolute error (MAE) is a better metric for climate models than RMSE but, Chai & Draxler [27] contend otherwise, arguing that the RMSE provides valuable information about the distribution of errors.

A mean squared error metric like R^2 or RMSE might be considered more valuable by an operator familiar with the assumptions behind a Gaussian distributions and the implications of a mean squared error in this context. However, in practice, MAE might be favoured for its simplicity. A further consideration is raised by Hyndman & Koehler [28] who discuss measures of forecasting accuracy in the context of comparing a different forecasting methods across a variety of problems. They argue that

performance metrics should be scaled by the performance of the naive (assume last value) forecasting method. Ultimately, this leads to the recommendation of mean absolute scaled error (MASE) given by [28]:

$$MASE = \frac{\frac{1}{n} \sum_{t=1}^n |\hat{y}_t - y_t|}{\frac{1}{n-1} \sum_{t=2}^n |y_t - y_{t-1}|} = \frac{MAE_{forecast}}{MAE_{naive}} \quad (2.1)$$

where y_t and \hat{y}_t are the actual and foretasted values, respectively, for time t. Essentially, this method is the MAE error for the forecast scaled by the MAE for the naive method. A MASE value less than one suggests the forecasting method outperforms the naive method and a value greater than one suggests the naive method is superior. An analogous scaled error could be derived using RMSE, but Hyndman & Koehler [28] prefer the MASE as MAE is easier to interpret and less sensitive to outliers than RMSE. MASE is not included as a metric in this report as it can be calculated using reported MAE values and unstandardised MAE can be more informative for a control practitioner trying to asses the reliability of the model.

Another notable observation from the research of Pani [11] is that, of the 14 models trained, the best performing model, using R^2 on the validation set, was a neural network with a R^2 of 0.8488 which was not a significant improvement on linear regression with an R^2 of 0.7685.

There are also potential issues with the methodological approach taken by Pani [11]. The first is the lack of a hold-out test dataset. Pani [11] states that the data set was split into a training and validation set but also suggests that cross validation was used during model selection and hyper-parameter fitting. It is unclear whether this cross validation uses a new 'validation' data set and if not, it casts doubts on the likelihood of attaining similar performance when actually taking the model online.

The second concern is the method of splitting the dataset into training and validation, Pani [11] uses the Kennard-Stone algorithm which attempts to try and increases the variability of the data in the training set. This results in a validation set which is likely to provide an optimistic estimate of model performance due to a lack of challenging, unusual observations. This conclusion is further reinforced by the fact that validation set performance reported by Pani [11] was generally superior to training set performance for most models. Normally, due to over-fitting, one would expect better accuracy on the training set.

Thirdly, it is worth noting that the Kennard-Stone algorithm results in a validation set with observations that were made before observations in the training set. This is not reflective of the real world application of a soft-sensor in which you cannot reach into the future for data to train a model that is used to predict the fineness

of cement being produced now. Pani [11] does note that he believes the cement grinding system is stationary, at least in so far as clinker raw-feed would be rejected if it does not meet plant standards. Even if this filtering is performed, there might still be a large range of variability in clinker quality. Furthermore, other parts of the plant might change such as equipment wearing or temperatures changing.

Finally, Pani [11] does not comprehensively explain the nature of his dataset, he states that there are 158 unique values for Blaine; but does not specify over what period of time these values were recorded or what method, if any, was employed to deal with the lag in lab testing times.

2.4.3 Research aim

The primary aim of this research is to report on the accuracy that can be achieved with a cement mill soft-sensor given that no literature was found reporting on the accuracy of a soft-sensor for the type of cement grinding circuit analysed in this paper.

Furthermore, the experiments reported on in this paper aim to address the following gaps in literature:

1. There was no literature reporting on the performance of a long-short term memory (LSTM)⁴ based soft-sensor for a grinding circuit.
2. None of the surveyed literature reported on the accuracy of a PPS soft-sensor using a holdout test set or real world online data.

2.5 Data Science models

Up until now, the literature review has focused on cement mill control and soft-sensors. The rest of the literature review is devoted to outlining data driven models available for regression.

Broadly, the class of general purpose data-driven regression algorithms can be thought of as having three parts. The first is the support of the functional mappings permitted from the feature space to the output space. Many of the more successful models of the last decade offer more general mappings. For example, every possible convergence of a simple linear regression is included in the hypothesis space of a neural network. The second is the loss function of the model, which includes but is not limited to cross-entropy, mean squared error and mean absolute error as well as potential regularization terms such as L1 and L2 norms of the parameters. Finally,

⁴The LSTM architecture is discussed in Subsection 2.5.3

there is the optimisation algorithms which seeks to adjust parameters of the model to best optimise the loss function and can include, for example, singular value decomposition, Newton-Raphson, stochastic gradient descent and genetic algorithms.

In the below outline, models are separated by the first factor, i.e. the potential functional mappings.

2.5.1 Linear models

The most simple class of regression models are linear models which for dataset $\mathcal{D}(\vec{x}_i, y_i), 1 \leq i \leq N$ are of the form:

$$\hat{y}_i = b + \sum_{j=1}^m \omega_j x_{ij} = b + \vec{\omega} \cdot \vec{x}_i, \quad 1 \leq i \leq N \quad (2.2)$$

where m is the dimensionality of the input vector, and x_{ij} corresponds to the j^{th} feature of the i^{th} observation.

Included in this category are multiple linear regression, ridge regression, lasso regression, elastic net regression and linear support vector regression (SVR) which differ only by their loss functions, detailed in Table 2.1.

TABLE 2.1: Table showing various loss functions for different regression algorithms

Model	Conventional loss function
Linear Regression	$\sum_{i=1}^N \ \hat{y}_i - y_i\ _2^2$
Lasso Regression	$\sum_{i=1}^N \ \hat{y}_i - y_i\ _2^2 + \lambda \ \vec{\omega}\ _1$
Ridge Regression	$\sum_{i=1}^N \ \hat{y}_i - y_i\ _2^2 + \lambda \ \vec{\omega}\ _2^2$
Elastic Net Regression	$\sum_{i=1}^N \ \hat{y}_i - y_i\ _2^2 + \lambda_1 \ \vec{\omega}\ _1 + \lambda_2 \ \vec{\omega}\ _2^2$
Linear SVR ⁵	$C \sum_{i=1}^N \ \hat{y}_i - y_i\ _1 + \ \vec{\omega}\ _2^2$

2.5.2 Kernel SVR

Kernel SVR generalises linear SVR by adjusting equation 2.2 to:

$$\hat{y}_i = b + \sum_{j=1}^p \omega_j [\phi(\vec{x}_i)]_j \quad 1 \leq i \leq N \quad (2.3)$$

⁵This formulation is not quite right, in reality the soft margin loss function is conventionally used which differs from the mean absolute error written in that it allows a margin of error before taking the absolute loss. For a very small margin, the soft margin loss function is essentially mean absolute error

where $\phi(\cdot)$ projects a vector of dimension m into a feature space of dimension p , generally $p > m$. However the actual mapping functions ϕ are often not explicitly calculated, instead optimization algorithms rely on distance Kernels ($K(\cdot)$) which are used to calculate the inner product of two points in the higher dimensional space.

There are many kernels that can be applied of which two of the most popular are the Radial Basis Function (RBF) kernel and the polynomial kernel [29].

The polynomial kernel of degree d corresponds to a transformation of all data points into a higher dimensional feature space which includes all feature interactions of degree d or less. The inner product in this space is given by,

$$\phi(\vec{x}_i) \cdot \phi(\vec{x}_j) = K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d \quad (2.4)$$

For $d = 2$, this corresponds to a feature mapping function:

$$\begin{aligned} \phi(\vec{x}) = & (1, \sqrt{2}x_1, \sqrt{2}x_2, \dots, \sqrt{2}x_m, \\ & x_1^2, x_2^2, \dots, x_m^2, \\ & \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \dots, \sqrt{2}x_{m-1}x_m) \end{aligned} \quad (2.5)$$

where \vec{x} is a single observation vector of dimension m and x_i refers to the i^{th} element of the vector \vec{x} . This can be seen by plugging Equation 2.5 into Equation 2.4. From Equation 2.5 it would seem that the polynomial kernel of degree two might allow for all first order interactions between variables. Practically, higher order polynomial kernels have a large risk of over-fitting to data, for $m = 20$ and $d = 3$ the polynomial feature space has thousands of dimensions.

The RBF kernel denotes a scaled euclidean distance and is given by:

$$K(\vec{x}_i, \vec{x}_j) = e^{-\gamma \|\vec{x}_i - \vec{x}_j\|^2}, \gamma > 0 \quad (2.6)$$

where γ is an adjustable 'spread' parameter. This kernel function corresponds to feature mapping that is theoretically infinitely dimensional.

2.5.3 Neural network type models

Multilayer perceptron (MLP)

A vanilla MLP involves successively stacked layers of linear transforms and 1-to-1 non-linearities. All ANNs surveyed in the grinding literature above were simple three layer, MLPs. One significant design choice for a neural network is the activation function for which common choices include logistic sigmoid (generally referred to as just 'sigmoid'), hyperbolic function (tanh) and rectified linear unit (ReLU).

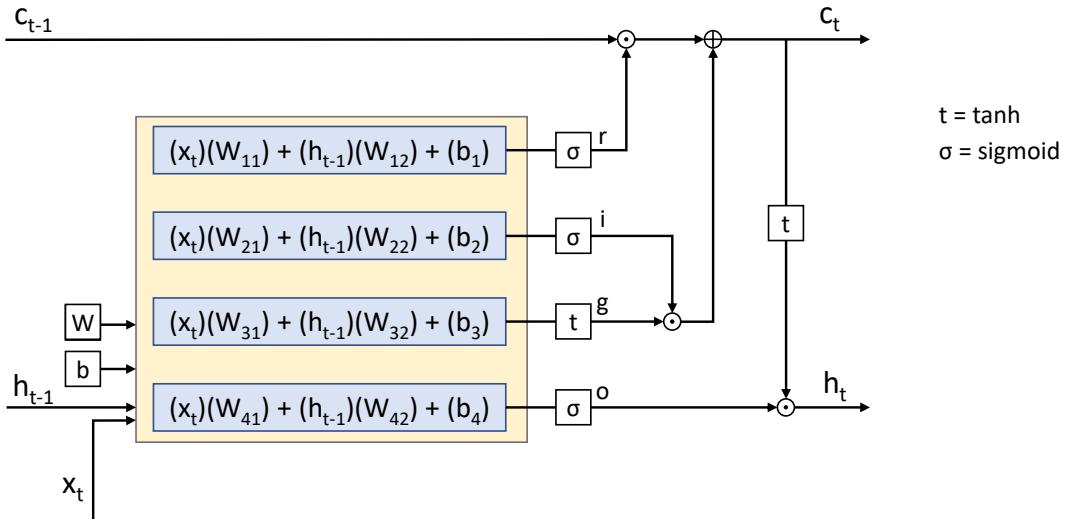


FIGURE 2.1: An illustration of the LSTM neural network architecture proposed by Hochreiter & Schmidhuber [32]

Since its introduction the ReLU activation function has grown in popularity, likely due to its simplicity, consistency and performance especially in deep-neural networks [30]. It has been argued that due to saturation problems the sigmoid and tanh functions are discouraged in favour of the ReLU function for feed forward networks [31]. As a result the ReLU activation function was selected for all MLPs implemented in this paper despite the fact that reviewed soft-sensor literature used the tanh [10] and sigmoid [11] functions.

Another different neural network architecture is the radial basis function neural network (RBFNN) which was used by Zhou et al. [9] and Pani [11]. An RBFNN is a form of MLP where a specific parametrised RBF activation function is used. An RBFNN is generally a shallow (1 hidden layer) network where the activations in the hidden layer are based on distances from centroids in the original feature space. For Pani [11], the RBFNN consistently performed worse than the MLP.

A long short-term memory (LSTM)

The LSTM architecture is a type of recurrent neural network (RNN). Early RNN architectures suffered from learning difficulties where backpropogated gradients would either explode or vanish resulting in models that took far too long to fit or were unable to fit at all [32]. A solution was proposed in 1995 by Hochreiter & Schmidhuber [32] as the LSTM architecture which involves a hidden state (a matrix) and four matrices known as 'gates', which control the flow of information in and out of the state as the model processes data in the sequence.

A feed-forward diagram of the network is shown in Figure 2.1.

To understand the LSTM network, it is worth noting that the difference between the sigmoid and tanh function is the output range which are $(0, 1)$ and $(-1, 1)$ respectively. Intuitively, the forget gate (r) controls how much of the state cell is forgotten at each time step. The input gate (i) controls how much is written to the cell. The state gate (g) controls what is written to the cell. The output gate (o) controls how much of the state cell will be written to h . If a predictive output is desired at time t , h_t is multiplied by a weight matrix and added to a bias weight. From the structure it is also clear that an initialisation is required for c_0 and h_0 for which zero matrices are commonly used. For a more formal exposition of the mathematics see Hochreiter & Schmidhuber [32].

Recently the LSTM has shown state of the art success at machine learning problems involving sequential data such as machine translation [33], speech recognition [13] as well as many other problems⁶. There are other recurrent neural network architectures but Greff et al. [34] found, after exploring tens of thousands of variants of RNNs on Google's servers using a genetic algorithm, that no algorithm could consistently provide superior results to the LSTM across problems.

Adaptive neuro-fuzzy inference engine (ANFIS)

Of all the methods Pani [11] used for soft-sensor prediction for a cement mill, he concluded the ANFIS to be most effective. The architecture was introduced by Jang [35] and represents a five layer neural network. Layer one separates each feature into membership of various fuzzy classes. Layer two nodes each represent a unique combination of possible memberships to the different class for each feature. This joint activation is calculated by multiplying incoming signals. Layer three normalises the firing strength of layer two by dividing by the cumulative sum of layer two firing strengths. Each node in layer four corresponds to a linear regression of the original input features multiplied by the firing strength given by layer three. The weights for the linear regression for each layer are trainable parameters. Finally layer five is a sum of layer four and represents the model output. The process is illustrated in Figure 2.2. The process is described in more detail by Jang [35].

2.5.4 Regularization

neural network type data-driven architectures have grown in popularity due to advances in computer hardware and growth in the availability of data. However, available data is not infinite and measures are generally needed to try to reduce over-fitting. In particular different regularization techniques are available which try

⁶For comprehensive lists of successful applications of the LSTM network see [31, 34]

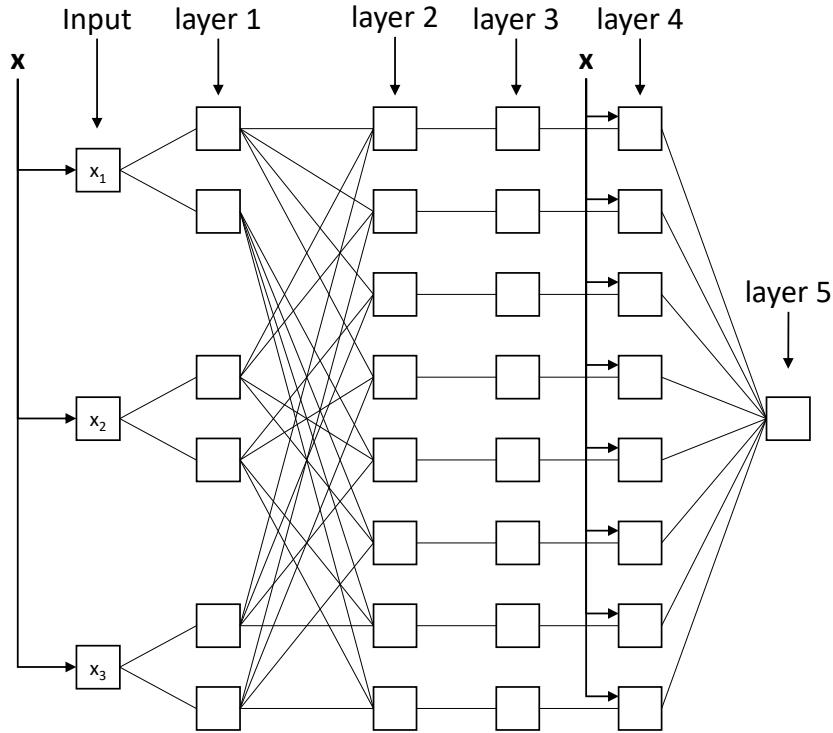


FIGURE 2.2: An illustration of the ANFIS neural network architecture proposed by Jang. [35]

to create penalties for a model over-fitting to a dataset. The most intuitive form of regularisation is the L1 norm as used in LASSO regression. Essentially, this favours sparsity by linearly penalizing the model for having non-zero weights. Another form is L2 regularization which adds a penalty term to the square of model weights. This could be interpreted as applying a Gaussian prior to the model weights and penalising the model for complexity by this metric.

In 2012, *drop-out* was proposed for improving the generalisability of neural networks [36]. Drop-out works by randomly setting network nodes to zero during training with pre-specified probabilities. As a result models would be penalised for relying too heavily on any particular combination of nodes for making a prediction, encouraging a node to develop independent predictive power rather than relying on other nodes to correct for its errors. Along similar logic, drop-out can be thought of as leading to a new sub-model being trained at each iteration during training, with the final network being a more robust quasi-ensemble. It was further shown that, for linear-regression, drop-out is a modified form of L2 regularization [37].

Applying the principles of drop-out to LSTM neural networks yielded limited results until a particular form was proposed by Gal & Ghahramani [38] that demonstrated state of the art performance on a natural language dataset.

2.5.5 Optimisation algorithm

As introduced at the beginning of Section 2.5 machine learning algorithms can often be considered as having three parts, 1) the model-form hypothesis space, 2) the loss function and 3) the optimization function. Optimization functions differ by model and implementation, for example neural networks are generally trained through some variety of back propagated gradient descent, SVM often solve the dual formulation of the problem. All the linear models including the kernel SVM are convex optimizations, and so local minima are not a concern and optimization algorithms differ most significantly by processing-time to solve.

The optimisation landscape for neural networks is more 'hilly', with many local minima, and gradient descent results in different convergences depending on the weight initialization. Although theoretical inquiry into optimization for neural networks is still somewhat recent [39], it has been argued by Choromanska et al. [40] that stochastic gradient descent algorithms converges onto one of several high-performing local minima as measured by performance on the test set. Choromanska et al. [40] do highlight that this convergence becomes less likely for larger neural networks⁷.

Given the importance of optimisation for deep learning, new algorithms are constantly being proposed to mitigate common problems such as slow convergence; getting stuck in a local minima, plateaus and saddle points; exploding or vanishing gradients as well as inaccurate gradients when using mini-batch estimation [31]. One group of solutions involve using adaptive learning rates for each of the various parameters with the goal of improving converge by, for example, increasing step size in dimensions where past steps have been small or in a consistent direction and decreasing step size in dimensions where the optimiser seems to be oscillating over a better value. One such adaptive learning rate algorithm is Adam, a gradient based optimization function introduced in 2014 that builds on the advances of methods like AdaGrad and RMSProp [42]. The update steps of the Adam algorithm is provided in Algorithm 1.

The success of Adam can intuitively be understood as leveraging off two ideas. The first is momentum, where the direction and size of steps in the optimization landscape are calculated using a weighted sum of all previous steps which pushes

⁷An interesting conclusion made by Choromanska et al. [40] is that poorer convergence for larger networks is irrelevant as training error and testing error decorrelate for larger neural networks due to over-fitting. They further conclude that recovery of a global minimum on the training dataset is therefore, unimportant when training a generalisable predictor. Yet, He et al. [41] made a breakthrough when introducing residual learning which creates a minor change to the optimization algorithms for neural networks but allows for significantly better convergence for deep (18 or more layers) neural networks and has been a part of state of the art algorithms for a variety of problems.

Algorithm 1 Extract of Adam algorithm set out by Kingma & Ba [42]. $f(\theta)$ is the objective function, e.g. a neural network with data and loss function. θ is the vector of trainable parameters. β_1 and β_2 are constants that determine the decay rates for the moment estimates. α is the learning rate.

$g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$	(Get gradients w.r.t objective for time step t)
$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$	(Update biased first moment - momentum - estimate)
$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$	(Update biased second moment estimate)
$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$	(Compute bias-corrected first moment estimate)
$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$	(Compute bias-corrected second moment estimate)
$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$	(Update parameters)

the optimiser in the same direction it has previously been travelling. The term β_1 can be understood as 'friction' which decays the historic momentum.

The second idea is to divide the step by a cumulative sum of the past gradients squared. This has the effect of slowing down the model in the dimensions for which it has been taking larger steps and accelerates the step size in dimensions where the past steps have been very small. This has the effect of reducing the time a gradient descent optimiser might spend oscillating around a point in the optimization landscape. This second moment estimate is also given a decay term (β_2) to prevent the optimiser from stopping prematurely in non-convex optimisation landscapes. Without the decay terms, step size in the optimization landscape would tend towards zero due to an endlessly growing denominator [42].

Furthermore the Adam algorithm removes a common training hyper-parameter; namely, the rate of decay on the learning rate. On the other hand, Adam does introduce two new hyper parameters, namely β_1 and β_2 . However, empirically, the optimiser has shown to be robust with default values of 0.9 and 0.999 for β_1 and β_2 respectively [31, 42]. This just leaves selection of the initial learning rate for which the main goal is to select a learning rate that is low enough that the optimiser can converge but high enough that convergence does not take too long. Selection of learning rate might also have a regularizing effect, as a low learning rate in combination with a prespecified number of training iterations might act similar to early stopping⁸. Goodfellow et al. [31] notes that no optimiser has been shown to be clearly best for neural networks but, adaptive optimisers (like Adam) tend to perform better.

Therefore, as a result of the superiority of Adam for neural networks (relative to simple gradient descent algorithms) and the sufficiency for convex optimization it is used for all models attempted in this paper, except for kernel SVR, for which the optimiser is discussed next.

⁸Early stopping is discussed by Goodfellow et al. [31] and Gal & Ghahramani [38]

2.5.6 Optimisation for kernel SVR

The primal formulation of the SVR is given by:

$$\begin{aligned} \min_{\vec{\omega}} & \left\{ \frac{1}{2} \|\vec{\omega}\|^2 + C \sum_i (\xi_i + \xi_i^*) \right\} \\ \text{subject to} & \left\{ \begin{array}{l} y_i - \phi(\vec{x}_i) \cdot \vec{\omega} \leq \epsilon + \xi_i \\ \phi(\vec{x}_i) \cdot \vec{\omega} - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{array} \right. \end{aligned} \quad (2.7)$$

where ξ_i and ξ_i^* are slack variables that combine to form the 'soft margin loss' function. C and ϵ are hyper-parameters of the algorithm. C is an implicit regularization term that when lowered gives more weight to minimising the norm of the parameter vector $\vec{\omega}$, rather than minimising the errors. This creates more regularization, potentially leading to a more generalizable algorithm. The hyper-parameter ϵ represents the acceptable error margin, a value of zero for ϵ turns the soft margin loss into mean absolute error. A larger value of ϵ can result in a lower computational cost when training and predicting using the algorithm.

Computing $\phi(\vec{x})$ can be computationally costly for the polynomial kernel and potentially intractable for the RBF kernel as a result of being a potentially infinite dimensional projection as discussed in subsection 2.5.2. However, the dual of the SVR optimization problem is conveniently given by [43]:

$$\begin{aligned} \max_{\vec{\alpha}, \vec{\alpha}^*} & \left\{ -\frac{1}{2} \sum_i \sum_j (\alpha_i - \alpha_i^*) \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) (\alpha_j - \alpha_j^*) - \epsilon \sum_i (\alpha_i + \alpha_i^*) + \sum_i y_i (\alpha_i - \alpha_i^*) \right\} \\ \text{subject to} & \sum_i (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \quad (2.8)$$

Note that $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j) = K(x_i, x_j)$ which, for a range of kernels, is a computationally less costly operation than actually projecting the data into the higher order dimensional space as would be necessary when solving the optimisation problem in the primal form.

Chapter 3

Methodology

This chapter explains the data preprocessing and the experiments performed. There are many ways in which data can be processed and models can be trained. The goal when designing the methodology for this research was to use enough data-preprocessing and training techniques to allow models to achieve the performance increases that would be obtained from some real world model tweaking whilst also trying to maintain a simple and consistent approach that could be repeatable, as opposed to a 'trial and error' approach.

3.1 Data and preprocessing

3.1.1 Description of plant

The data used in this research report comes from a large cement mill operating between the dates of 06/07/2018 and 28/11/2018.

A diagram of the circuit for the mills analysed in this report along with measured process variables is given in Figure 3.1. All measured process variables are typeset in all capitals to indicate they are recorded numerically as a feature in the dataset and a description of each feature is given in Table 3.1. A correlation plot of the feature variables is given in Figure 3.2 and helps to offer a high level intuition of the dynamics of the circuit.

The correlation plot given by 3.2 was calculated over almost five months of data and provides a high level aggregated picture. During this period the plant shifted between closed loop, open loop and partial closed loop as various control variables were taken in and out of closed loop control. To provide a more accurate picture of the plants dynamics, an analysis would be required that looks at how correlations change over time with different lags, for each different combination of control variables under closed loop. As different combinations of control variables were operated in closed loop the dynamics of the plant were likely to change dramatically. The correlations are presented for the total time period for two reasons; firstly, there

TABLE 3.1: Description of captured information in cement circuit given by Figure 3.1

Variable name	Unit-of-measure	Description
FEED	t/h	The incoming clinker feed rate.
BIN	%	A measurement of how full the bin is with clinker. The bin feeds directly to the roller press and the flow rate out of the bin is determined most significantly by the roller press performance.
RPAMPS	A	The power draw of the roller press mill. The roller press mill is set to run at a constant speed, therefore, the power draw is related to the characteristics of the particles being crushed.
RPEAMPS	A	The pre-crusher elevator is set to run at a constant speed so its power draw is indicative of the circulating load in the pre-crusher circuit.
VAMPS	A	The amp draw of the fan pulling air through the V-Separator. The V-Separator is passive and relies on airflow. The amp draw of the fan indicates the amount of dust being pulled out the separator.
VSEPDAMPER	%	The position of the damper controlling how much of the cement-dust-carrying airflow is returned to the V-Separator and, thus, not directed to the main separator.
MAMPS	A	The amp draw of the main ball mill which rotates at a constant speed. This features is most closely related to the load of the mill.
OUTDAMPER	%	The position of the damper which controls the airflow through the main mill.
EAMPS	A	The amp draw of the main mill elevator which is indicative of the circulating load for the main mill circuit.
SEPSPEED	rpm	The speed of rotation for the main centrifugal separator. A higher separator speed is expected to result in coarser particles in the final cement.
SEPDAMPER	%	The position of the damper controlling the circulating airflow through the main separator. An increase in the airflow might increase the PPS and flow rate of the final product.
TOUT	°C	This temperature of the final cement produced. A higher temperature might suggest more crushing due to the energy released during crushing.
RESIDUAL	%	The percentage of cement particles passing through a sieve. This is another metric for cement fineness/ quality
BLAINE	m ² /kg	The main measure for cement quality/fineness. A higher Blaine means more surface area for the same mass, implying a finer powder.

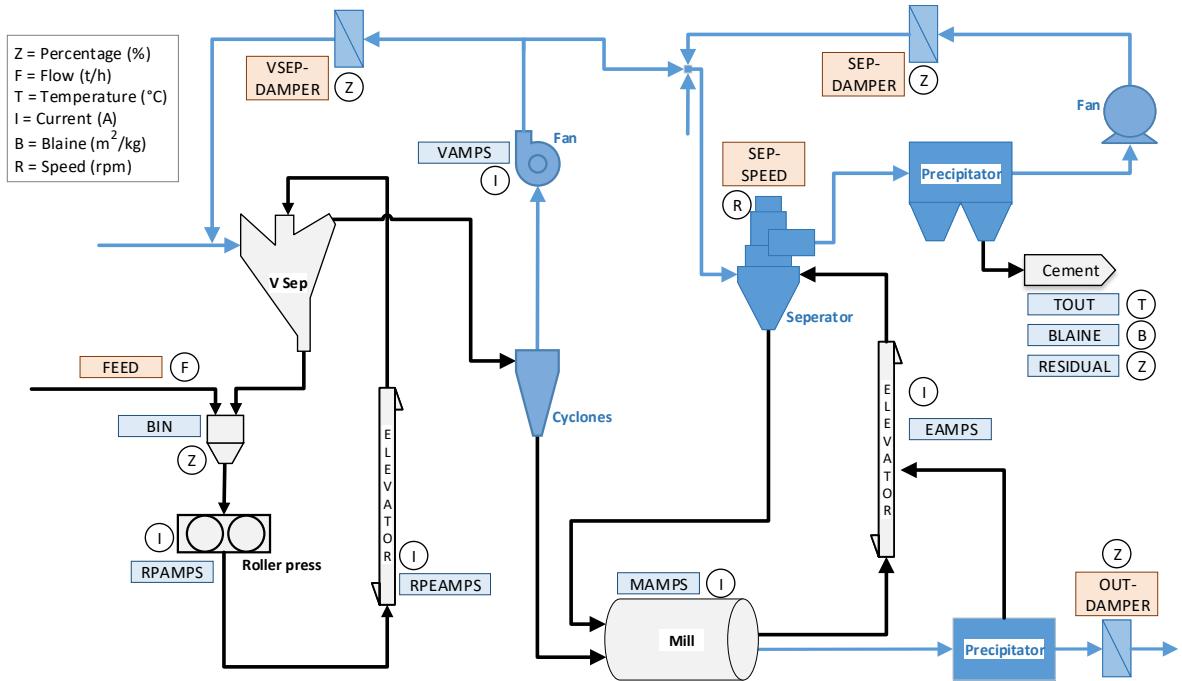


FIGURE 3.1: Milling circuit. Black arrows represent product flow and blue arrows represents airflow. Orange boxes represent controlled variables, blue boxes represent measured variables.

was limited data for the mill operating under full open-loop and, secondly, this research aims to build a model that could be robust for the full range of dynamics of the plant.

The strong positive correlations shown in Figure 3.2 between FEED, RPAMPS, RPEAMPS and VAMPS, suggests that a greater feed relates to more particles in the pre-crusher circuit and therefore a higher current draw by the components in this circuit. The effect of VSEPDAMPER is not well understood. It has a weak positive correlation with the amp draw of pre-crusher components and also shows a weak correlation to SEPSPEED, TOUT and BLAINE. These correlations suggest that the effect of this damper on controlling the proportion of cement dust redirected to the V-Separator (instead of the main Separator) has an effect on both the pre-crusher circuit and separator.

The negative correlation between MAMPS and EAMPS could be the result of many different factors. Generally, a ball mill is understood to have a 'n' shape relationship between the power draw of the mill and the load of the mill. An empty mill draws a fixed amount of power to rotate its mass and the steel balls inside. As more clinker is fed in, the power draw would be expected to increase as the mill churns more product. However, after a certain point, the mill becomes overloaded and the power draw decreases as less energy is transferred into crushing. Therefore,

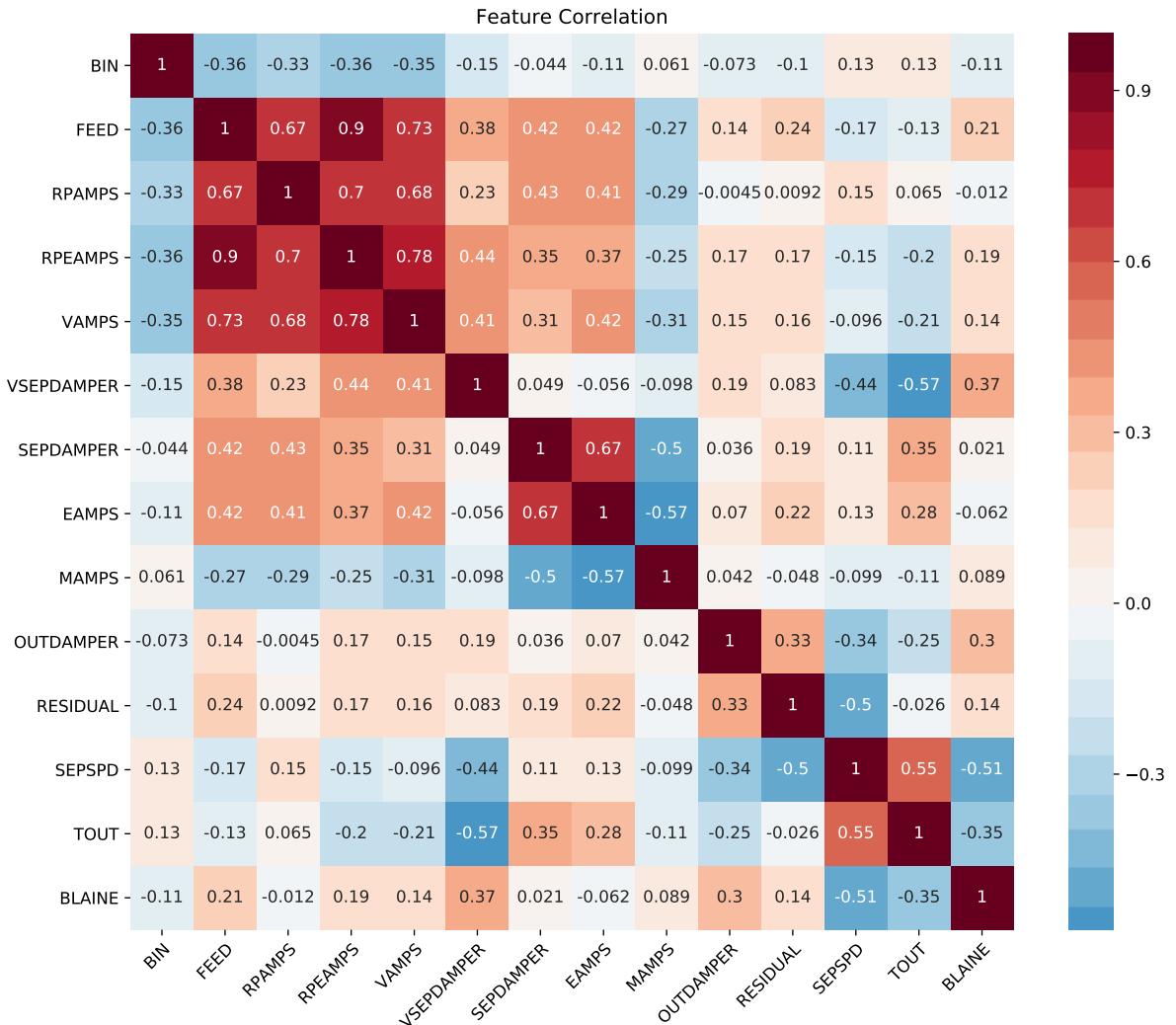


FIGURE 3.2: A linear correlation matrix heat map for process variables in the plant

the relationship between circulating load and mill power draw is not clear and linear. This negative correlation in the data might result from the plant spending more time on the overloaded side of this 'n' curve or it might result from human and automated controllers controlling the mill load on the 'under-loaded' side of the curve such that after a period of being close to full load the controllers have managed to reduce the mill load such that there is a lot of mass on the elevator but less mass (and therefore power draw) in the mill. SEPDAMPER has a strong positive correlation with EAMPS, which is possibly the result of operators trying to increase the circulating air-load through the separator when more product is being fed into the separator.

The positive correlation between SEPSPD and TOUT is possibly because a higher separator speed results in particles bumping around more and absorbing kinetic

energy as heat. The negative correlation between SEPSPD and BLAINE would intuitively be interpreted as the result of coarser particles being sorted into the final product when there are higher centrifugal forces. However, any correlation between BLAINE or RESIDUAL and other variables is complicated because they represent the correlation between the online feature and the last known value of the BLAINE which could be over an hour out of date.

Overall, this preliminary analysis of the correlation plot highlights the complexity of the dynamics and interactions of variables in the mill circuit. This provides further justification for the decision to try and create a model using a non-linear black-box modelling techniques that might be able to capture the complexity of this system similar to how these modelling techniques have succeeded in capturing the complexity of images and speech.

3.2 Data preprocessing

3.2.1 Filtering out periods of non-operation

An initial time series plot of a few select process variables, shown in Figure 3.3, demonstrates the mill being brought in and out of operation. There are periods of non operation such as the first half of August as well as periods of intermittent operation, such as the second half of August where the mill was cycled in and out of operation, alternating with the other mills at the plant (there are four separate mill circuits at the plant). Another reason for cycling the mill in and out of operation is to take advantage of off-peak electricity prices.

Periods of non-operation were filtered out by removing rows where the mill draw (MAMPS) was less than ten Amps. The clear separability of the data is shown in Figure 3.4. Data observations with non-numeric values have also been filtered out.

There are two potential output variables in this circuit that describe the cement PPS, namely Blaine and Residual. Both results stem from robotic lab tests. Blaine, as defined above, represents a measure of the surface area per unit mass. Residual represents the percentage of particles passing through a sieve. Only Blaine was modelled in this research report as it is the primary performance indicator of the plant being analysed. Modelling other measures of PPS should be quite similar.

Consultation with plant operators suggests that values for Blaine are sampled approximately every hour and then reported 20 minutes after sampling. Blaine values to be predicted are therefore connected to the state of the plant 20 minutes prior.

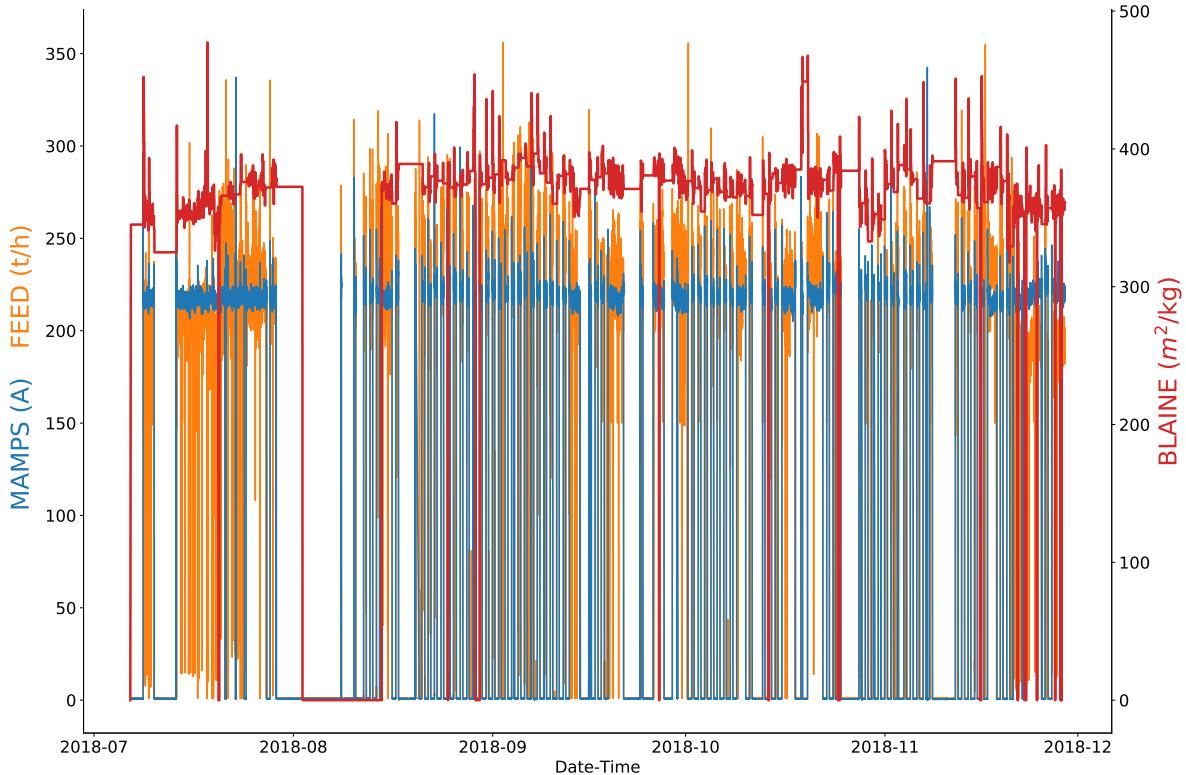


FIGURE 3.3: Plot of mill amps, Feed and Blaine values over the course of data capture

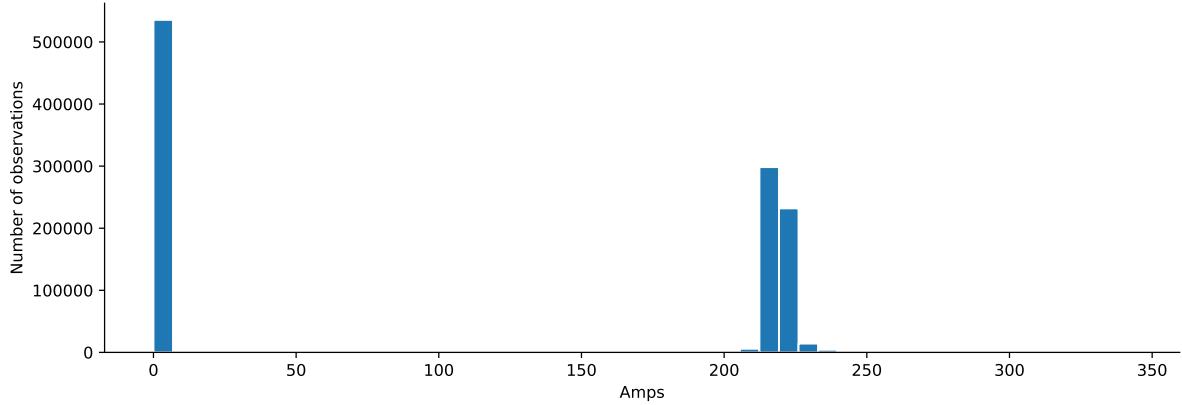


FIGURE 3.4: Plot showing the relative frequency of different observed values for mill current draw (MAMPS)

Filtering out noise

Inspection of data suggested some high frequency noise in measurements which can be seen in Figure 3.5. This problem was common in literature and some form of high-frequency filtering was employed [9, 10, 24]. There are many different algorithms for filtering noise but only simple, easily implemented algorithms are considered for this research report. Let $p = \{p_0, p_1, \dots, p_T\}$ be some sequence. Two common

low-pass filters are the simple moving average given by:

$$\tilde{p}_t = \sum_{i=0}^{i=n} p_{t-i}, \quad (3.1)$$

and the exponentially weighted moving average given by:

$$\tilde{p}_t = \begin{cases} p_t, & \text{if } t = 0 \\ \alpha p_t + (1 - \alpha) \tilde{p}_{t-1}, & \text{otherwise,} \end{cases} \quad (3.2)$$

where n and α are parameters to be selected and \tilde{p} is the filtered sequence. Both methods were developed for the purpose of improving time-series forecasts by filtering out high-frequency fluctuations to uncover an underlying trends and cycles [44].

Practically, an exponentially weighted moving average weights more highly recent observations, allowing the filtered values to respond quicker to jumps in the system. This feature of the algorithm was considered desirable and therefore exponential smoothing was used in favour of a simple moving average.

Exponential smoothing requires selection of α . A smaller value of alpha filters out more high frequency noise but also slows down the rate at which information from large jumps in a signal are included for the model. A decision was made to choose a value of α such that, with ten second sampling, the cumulative weight attached to values from the last five minutes exceeds 95%. This works out to an α of 0.095 as shown in Equation 3.3:

$$\begin{aligned} 0.95 &= \alpha + \alpha(1 - \alpha) + \alpha(1 - \alpha)^2 + \cdots + \alpha(1 - \alpha)^{30} \\ \implies \alpha &= 1 - (1 - 0.95)^{1/30} \\ &= 0.095 \end{aligned} \quad (3.3)$$

An example of the effect of the smoothing is shown for the variable EAMPS in Figure 3.5. Also shown is the smoothing that would be achieved if 95% of the cumulative weight was assigned to the last minute, in which case α would be set to 0.393. Setting α involves making a judgement on inherent bias-variance trade-off and, from inspection of Figure 3.5, it might be argued that setting α equal to 0.095 (black line) better captures the underlying process than setting α equal to 0.393 (red line). This conclusion follows from a prior understanding of the system where the change of particle mass on the elevator seems unlikely to result in such aggressive, approximately normally distributed, osculations over the period of 30 seconds to two minutes.

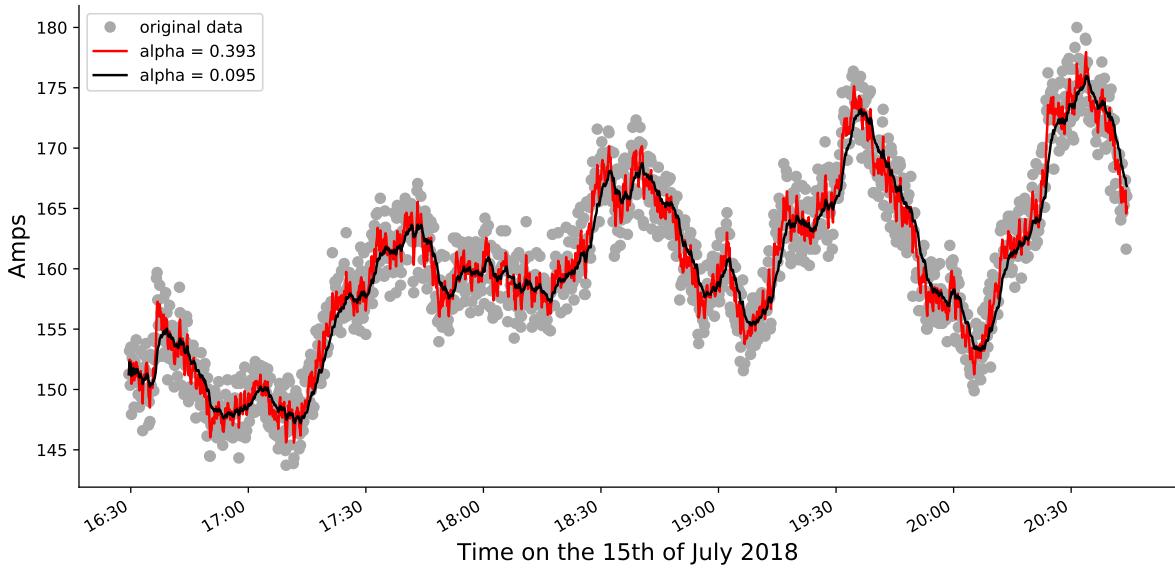


FIGURE 3.5: Plot showing effect of exponential smoothing applied to the variable EAMPS for random extract of data.

3.2.2 Including past plant conditions

Cross correlation calculations were used to determine the temporal relationship between process variables. By finding the highest absolutes cross correlation for various lag times an average delay time between the two variables can be determined. For example the cross correlation plot for RPEAMPS and EAMPS is given by Figure 3.6. This plot suggests that on average particles in the pre-crusher elevator take four minutes and 50 seconds (29 time steps of ten seconds) to go through the V-Separator and ball mill, reaching the main elevator.

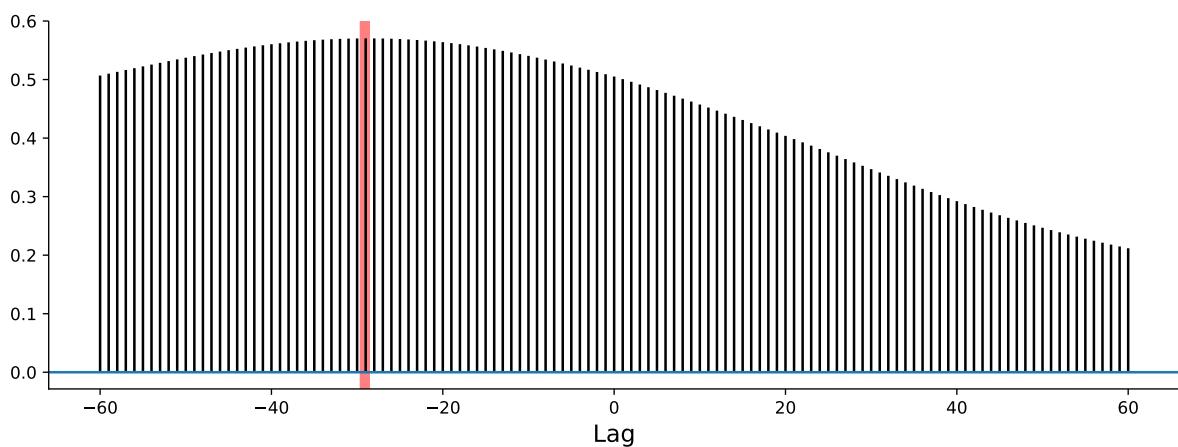


FIGURE 3.6: Plot showing cross correlation between RPEAMPS and EAMPS, with the maximum correlation at a lag of -29 time steps highlighted in red

This cross correlation process was performed for all process variables against all other process variables and displays a fairly consistent pattern in the data. Figure

3.7 shows the average position of process variables in time. Each line represents an estimate of the temporal position of all process variables based on their relative cross correlations with one particular process variable (specified in the legend). The y-axis displays the lag in minutes. For the purpose of a coherent visualisation each cross-correlation-line is connected at the process variable EAMPS. After having connected all the lines, the y axis is set by assigning a value of 0 minutes to the TOUT lag estimate relative to the variable EAMPS. TOUT is the temperature of the cement produced and therefore should be at the end of the process.

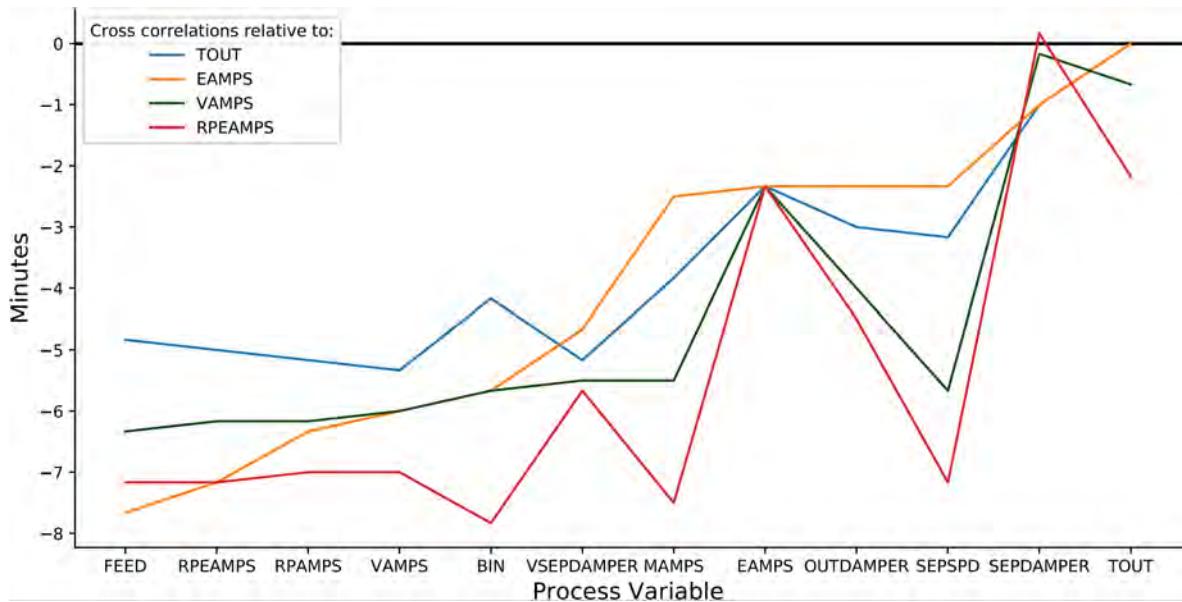


FIGURE 3.7: Plot showing average temporal relationship between process variables as would be suggested by cross correlation calculations

The information in Figure 3.6 is in many ways consistent with an intuitive understanding of the circuit. The order of most features relates to the order of circuit components one would expect a population of clinker to pass through on the way to becoming cement. The TOUT curve estimates a shorter time for the whole process, likely because some cement powder passes through the cyclones short-cutting directly into the main separator. On the other hand the EAMPS curve estimates the process to take longer as this curve would relate to particles that have to pass through both mills.

The most peculiar feature of this plot is the dip at 'SEPSPD' for the VAMPS and RPEAMPS curves. This may be a result of the split that happens at the cyclones where some particles move on to the main separator and other particles pass through the ball mill.

The plot given by Figure 3.7 does have several limitations. Firstly, it was calculated based on only linear correlations. Secondly, the cross correlations were calculated assuming stationarity. Thirdly, it is only a measure of correlation between

the process variables and these correlations could be caused by several factors including, not just the flow of clinker but also, feedback loops and closed loop control algorithms.

This analysis was used to inform the choice of lagged values for process variables for the regression models. Historic plant conditions are included in the dataset by adding one lag at eight minutes. The choice of eight minutes was motivated by Figure 3.7, where eight minutes seems to span the estimated time from clinker particles being fed into the circuit to their exiting as cement. It also allows for models to use information on the trends of process variables, such as their increasing or decreasing, over the previous eight minutes.

This approach favours simplicity and better models might be trained by more careful selection of how many lags and which lags to use for each individual process variable.

After adding lagged values to each data row, all rows without unique samples of Blaine were discarded for the purposes of modelling. Another option would be to interpolate between different measurements of Blaine in order to synthesise more data observations. This technique was not attempted for this research as it would result in further complications such as the reliability of results reported on synthesised (interpolated) test set data.

The LSTM model, on the other hand, is designed to handle panel data, where each observation is 2-dimensional: features and position in sequence. Using too much historic data as feature variables for each observation for an LSTM is too computationally costly and very likely to result in extreme over-fitting. Based on the results in Figure 3.7 a window of 20 minutes was chosen such that every LSTM prediction would only use data from the 20 minutes prior to the cement sample being taken for determining Blaine. Given ten second sampling for the features measured online at the plant, there are 120 sampling observations in those 20 minutes. To try avoid over-fitting, only five samples were taken corresponding to one sample every four minutes. Model performance might increase by choosing more suitable periods for selecting data for the LSTM.

3.2.3 Training, validation and test split

Finally a train-validation-test split was performed in the ratio 60%, 15%, 25% respectively. The data was not randomly shuffled ensuring that, instead, the oldest 60% would be training data and the newest 25% would be testing data. This choice was made to avoid data leakage which would result in an overly optimistic description of model performance compared to using a model online. When bringing a

soft-sensor online, one would not be able to use data on future plant operating conditions to build a model to predict current conditions. If plant conditions change dramatically over time and invalidated the reliability of a model trained on earlier data, this should come through as poor testing accuracy.

After filtering there were 1359 observations, yielding a train-validation-test split of 815, 204 and 340 observations respectively. This roughly resulted in the data for July through to the beginning of October being in the training set, the rest of October being in the validation set and November making up the test set. Note that the validation set is used for feature selection and hyper-parameter search algorithms, and therefore, in principle, performance on the validation set is not as reliable an indicator of online performance as performance on the test set. It is possible that changes in weather i.e. temperature and humidity may have an affect on the system but there does not appear to be any research exploring this relationship.

The training set features and output variable were normalised to have a mean of zero and a standard deviation of 1. The validation set and test were shifted and scaled using the same factors

3.3 Experiment description

The aim of the experimental design is to determine which model is most effective and what accuracies are possible for a cement mill soft-sensor. When designing a software model in practice, an engineer might try several different models with different parameters using trial and error to converge on the best model. Methodologically, for research, this process has several drawbacks. Firstly, it renders unclear how much customization and tweaking would be needed if the model were to be implemented in other conditions or on another plant. Secondly, it provides an unclear comparison between models, as some models might have received more effort when being adjusted for the problem. When a dataset is widely available this would not be a problem as many different researchers could compare their best version of a model.

On the other hand, not applying enough parameter search and model tweaking would be unrepresentative of the best models that would be derived in practice. Therefore, the goal when designing a methodology for this experiment was to create a training pipeline that allows for an automated, consistent training process which still provides each algorithm the opportunity to converge to an optimal model.

The model training pipeline generally included three parts.

1. A hyper-parameter grid search.

2. A Feature search using hyper-parameters from step 1.
3. A second hyper-parameter grid search using features chosen from step 2.

3.3.1 Grid search

Bergstra & Bengio [45] explore the process of hyper-parameter selection for machine learning algorithms and found that, given a set amount of computational resources, a random search is expected to result in better models than grid search. They believe this is because there are some hyper parameters that are likely to have a larger effect than others and a grid search strongly restricts the total number of different values explored on a given hyper-parameter dimension. However, this effect is less significant when the hyper-parameter dimensionality is small, i.e. 1-5 dimensions. Furthermore, a random search does not guarantee that the hyper-parameter space will be adequately explored whilst also making experiments more difficult to repeat. Therefore, a standard grid search was used.

3.3.2 Feature selection

Feature selection serves as a type of regularisation, it limits the model hypothesis space to the models that would have zero coefficient for certain features. Assuming variables that provide less, or redundant, information in predicting the output are removed, any trained model is more likely to generalise better by not over-fitting to the noise in these features.

Many feature selection algorithms involve filtering out variables that have a less interesting distribution or that are less effective at predicting the output in a univariate models, however, these methods are only indirectly related to the ultimate goal of developing the best generalising model.

On the other hand, selecting features based on performance on a validation set allows for the interaction of features in creating a good model. However, finding the best features by measuring the accuracy of a model on a cross validated dataset for all possible feature combinations is an NP-hard problem [46].

An alternative is to use a local search heuristic such as recursive feature elimination [47]. Recursive feature elimination requires that a model provides a coefficient for each feature and assumes that features with less significant coefficients are more likely to be redundant. As such, the algorithm iteratively removes the feature with the least significant coefficient and retrains the model. However, removing the feature with the least significant coefficient might not result in selecting the neighbour

model that performs best on the validation set. Note that in this problem description, a neighbour model is a model trained with one less feature. Therefore, another option might be an iterative feature elimination search which tests every model that can be created with one less features and chooses the model that performs the best on the validation dataset. Measuring performance by validation set accuracy, this algorithm is likely to result in a superior search to recursive feature elimination at the cost of higher complexity. The higher complexity results in acceptable training times given the problem at hand.

3.3.3 Measures of accuracy

The R^2 is reported for the training, validation and test set with the MAE being reported for the test as well. Furthermore, a demonstration of online performance on MAE is also given. Under this system, model features and hyper-parameters are chosen using the process described above after which the model is tested using a sliding system, where the model is retrained every ten observations (which corresponds to ten hours or roughly a day of operation). For the 340 test set observations this yields 34 different re-trainings of the model. This system is illustrated in Figure 3.8.

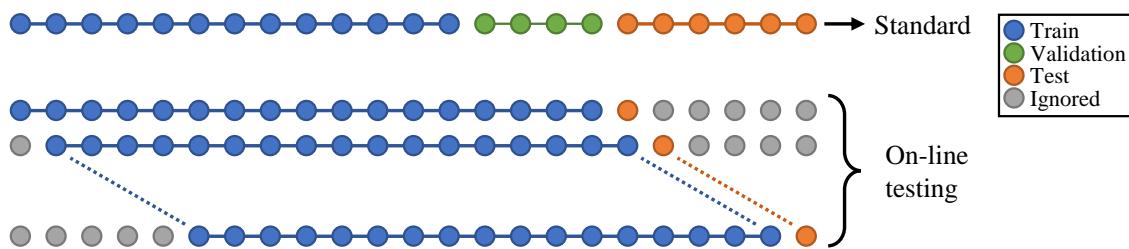


FIGURE 3.8: Illustration of how data was split for the standard and online simulated models

3.3.4 Tested models

The following models were included for comparison:

1. Persistence model,
2. Linear regression,
3. Lasso regression,
4. Ridge regression,
5. Elastic net regression,
6. Linear SVR,

7. SVR RBF,
8. SVR polynomial,
9. MLP-shallow,
10. MLP-deep,
11. LSTM and
12. ANFIS.

A brief description of each model implementation follows.

Persistence model (1)

This benchmarking model uses just the last recorded value for Blaine to predict the next and is also known as the 'naive' prediction [28].

Linear models (2-6)

The linear models were implemented as described in Subsection 2.5.1. All models were optimised using the Adam optimiser. Learning rate was set to 0.01 which resulted in convergence on the convex optimisations for all models. Where the loss function included a regularisation term, grid search was used only for λ ¹ or in the case of elastic net regression, both λ_1 and λ_2 were searched for. Grid search involved five equally spaced-values for $\log_{10}(\lambda)$ in the range $[-4, 2]$ and the secondary search after feature elimination used the initial λ and a range of $[\log_{10}(\lambda) - 1.5, \log_{10}(\lambda) - 1.5]$.

SVR (8-9)

From the formulation for kernel SVR, laid out in Subsection 2.5.2, there are two hyper-parameters that need tuning namely C and either γ or degree (d) depending on whether it is a RBF or polynomial kernel function. A third hyper-parameter that is often tuned is ϵ which controls the margins of the soft margin loss function. For $\epsilon = 0$ we have mean absolute error and for a very large ϵ only observations with predictions that are very far from their actual value will be considered in adjusting the parameters of the model, effectively reducing the number of observations used to define the fit. When optimising the dual, a larger ϵ might lead to a lower complexity and therefore shorter training time. Given that there is no need to reduce training time, ϵ is set to 0 for all experiments. Notably, Pani [11] consistently found smaller ϵ s to lead to better models.

¹for linear SVR this is equivalent to searching for $1/C$

Although γ was included in the hyper-parameter search for the RBF kernel, d was fixed at 2 for the polynomial model. A degree of 1 would just yield linear SVR and a degree of 3 led to extreme over-fitting during preliminary tests.

MLP (9-10)

For each of the two neural network models the number of hidden layers was selected in advance, one hidden layer for the shallow network and three hidden layers for the 'deep' network. The shallow model demonstrates the jump from linear regression to a neural network and the 'deep' neural network demonstrates the potential for a non-linear neural network to learn more complex mappings in the data. For the shallow model, the number of nodes in the first layer (h_1) was chosen using initial grid search of values of $\{2, 9, 16, 23, 30\}$. For the 'deep' model the number of nodes in the first layer utilised the same grid search and the number of nodes in subsequent layers (h_2, h_3) was calculated using the following formula:

$$h_2 = \lceil h_1/2 \rceil, h_3 = \lceil h_1/4 \rceil \quad (3.4)$$

This decision follows from the general trend of declining layer size used in state of the art neural networks [41, 48].

Furthermore the log of learning rate and $\log_{10} \lambda$ were also included in the grid-search for five values between $[-4, 2]$ for each. Post feature elimination grid searches of $[\log_{10}(\cdot) - 1.5, \log_{10}(\cdot) + 1.5]$ were used for the logs of learning rate learning rate and λ .

LSTM (11)

For the LSTM, 'drop-out' was applied as proposed by Gal & Ghahramani [38]. The LSTM training involved searching for two parameters in the grid search, learning rate and λ . Gal & Ghahramani [38] and Srivastava et al. [37] both found drop-out performance to generally be optimal when the drop-out probability is around 50%. Therefore, a flat rate of 50% was used.

ANFIS (12)

The ANFIS architecture was implemented based on the outline given by Jang [35]. Unlike Pani [11], Gaussian fuzzy membership functions were used instead of triangular or trapezoidal fuzzy membership functions. Furthermore, given the number of features in the data set, the model complexity would be too large for more than two class memberships per feature. Furthermore, for these Gaussian membership

classes, the centres (μ) were initialised at zero and one for each feature and standard deviation (σ) was initialised to one. However, both parameter vectors were trainable by the Adam optimiser.

Chapter 4

Results and discussion

4.1 Modelling results

The results for the various models are summarised in Table 4.1.

TABLE 4.1: Performance for various models

Model	R ²				MAE	
	Train	Validation	Test	on-line	Test	on-line
Persistence	0.257	0.271	0.154	0.154	11.108	11.108
Linear regression	0.541	0.545	0.334	0.481	11.365	9.272
Ridge regression	0.535	0.546	0.364	0.473	11.005	9.290
Lasso regression	0.525	0.538	0.425	0.468	9.771	8.993
Elastic regression	0.518	0.528	0.409	0.458	9.934	9.078
Linear SVR	0.524	0.543	0.368	0.453	10.408	9.152
SVR-Poly	0.569	0.583	0.404	0.430	9.526	9.361
SVR-RBF	0.516	0.521	0.390	0.384	9.706	9.106
MLP-Shallow	0.562	0.545	0.396	0.460	9.571	8.799
MLP-Deep	0.569	0.523	0.473	0.462	9.020	8.836
ANFIS	0.231	0.375	0.303	-225.597	10.095	233
LSTM	0.516	0.375	0.442	-0.114	9.594	14.392

The column 'online' in Table 4.1 represents the methodology described in Sub-section 3.3.3, which simulates online retraining of the model every ten observations. MAE is an unstandardised measure, in practice an engineer might use a qualitative heuristic to determine whether the accuracy is sufficient such as requiring the soft-sensor error to be less than 10% or 5% the range of the process variable. For the case of Blaine, with an empirical range of 160 for the dataset, the 10% and 5% of total range MAE values are 16 and 8 respectively. The persistence model has a MAE of 11.108 and therefore, by itself, might be reliable enough to practically implement a control system.

The results show that a better accuracy can be achieved relative to the persistence model. However, the accuracy is limited, with the best model providing a test set R^2 score of just 47.3% from the MLP-Deep model or MAE of 9.020. For the online methodology MAE drops further to a best case of 8.799 for the shallow MLP and the best R^2 is 0.481 for the linear regression.

Looking at the Persistence model, there is a significant drop in performance for the test set which is suggestive of a change in the statistical properties of the process, and evidence of time-variance in the system.

Interestingly, most of the linear models show greater signs of over-fitting than the multi-layered perceptron, as measured by the greater drop in test set accuracy. This might be the result of some features having a reliable linear relationship for the training set and validation set but not for the test set. This pattern was not repeated for the lasso regression which has a significantly greater test set R^2 .

The over-fitting is a particularly difficult problem as it is not clear how one would know to choose the best model in practice. For example, if a plant operator had trained both the lasso regression model and the ridge regression model, there is no clear reason why the lasso model should be chosen instead of the ridge regression model. When the algorithms are set to retrain online, the differences between models shrink, suggesting that the data becomes more relevant and the risk of over-fitting reduces.

The non-linearity introduced by the two types of kernel SVR did not result in an improvement against lasso regression although performance was better than the other linear models.

The best model by test set performance is the 'deep' MLP. The converged model did have a relatively low complexity as only seven features were selected and the three hidden layers have three, two and two nodes respectively. At first glance the performance of the 'deep' MLP might suggest that the neural networks are successful at creating a robust non-linear mapping but there is reason to believe the result might not be universal. The validation accuracy is relatively low for the deep MLP and across all the models there is an inconsistent relationship between validation and test set accuracy. This inconsistency suggests that there is time variance in the system, and different statistical properties for the two datasets. *A priori*, there is no reason to believe that a neural network with a given performance on a validation set is likely to perform better than a linear model with a higher validation set accuracy. In fact, using just validation set performance, the linear model would likely be considered more robust due to having a lower complexity¹. As such, the superior

¹For a springboard into Occam's Razor and its relationship with model complexity in Data Science see Schmidhuber[49].

accuracy of the deep MLP might not be generalisable.

Relative to the other algorithms the LSTM performed well on the test set, poorly on the validation set performance and poorly when trained online. Realistically, the online performance reflects the fact that hyper-parameters are sensitive to the dataset, possibly as the learning rate no longer resulted in optimal early stopping. The methodology proposed by Gal & Ghahramani[38] was utilised to try and reduce over-fitting. Another potential option to try and improve LSTM accuracy would be to try and train an LSTM to consistently predict Blaine and use interpolations or splines to infer Blaine values in-between hourly observations. This approach might also benefit from a method of handling noise in Blaine measurements (discussed below).

The general level of performance suggests that it is more difficult to build an accurate soft-sensor for the the circuit analysed in this research than for the circuit analysed by Pani[11]. For example, Pani[11] reports a linear regression R^2 of 0.7685 whereas in this paper the comparable model achieved an R^2 of only 0.346. This point is further stressed by the fact that Pani[11] used only three variables, 'Hot air flow, Classifier RPM and Clinker inflow', with no lagged observations or smoothing. The circuits are different but these features used by Pani[11] could be considered analogous to SEPSPD, SEPDAMPER and EAMPS. A linear regression trained on only these three features achieves an R^2 of 0.099 on the test set.

The ANFIS model demonstrated a high degree of instability, where an acceptable model was converged on during training, yet, when the model was retrained online, the model was prone to settle on highly inaccurate predictions.

Finally, it is clear that the online retraining of the model led to better performance, suggesting that consistent retraining may help deal with time variance problems. A further area for tweaking the model would be to find the optimal window for including historic data. A more complicated alternative would be to create algorithms that drop outdated observations but keep observations at the extremes of the feature space. This class of algorithm would try and maintain a relevant but comprehensive training set. A variation was applied by Zhou et al. [24] in order to reduce the size of the query set, and the resulting computational complexity, for a k-NN soft-sensor.

4.2 Plant information used by models

Figure 4.1 shows predicted values alongside actual values for the best performing model, the deep MLP model. Figure 4.2 removes periods of non operation which more clearly shows that the predicted values are closely related to actual values in

the previous time step. This is manifested in a predicted plot that looks vaguely like the actual plot shifted one step to the right. This observation suggests that the model is relying, to a large degree, on past values of Blaine to predict the present value.

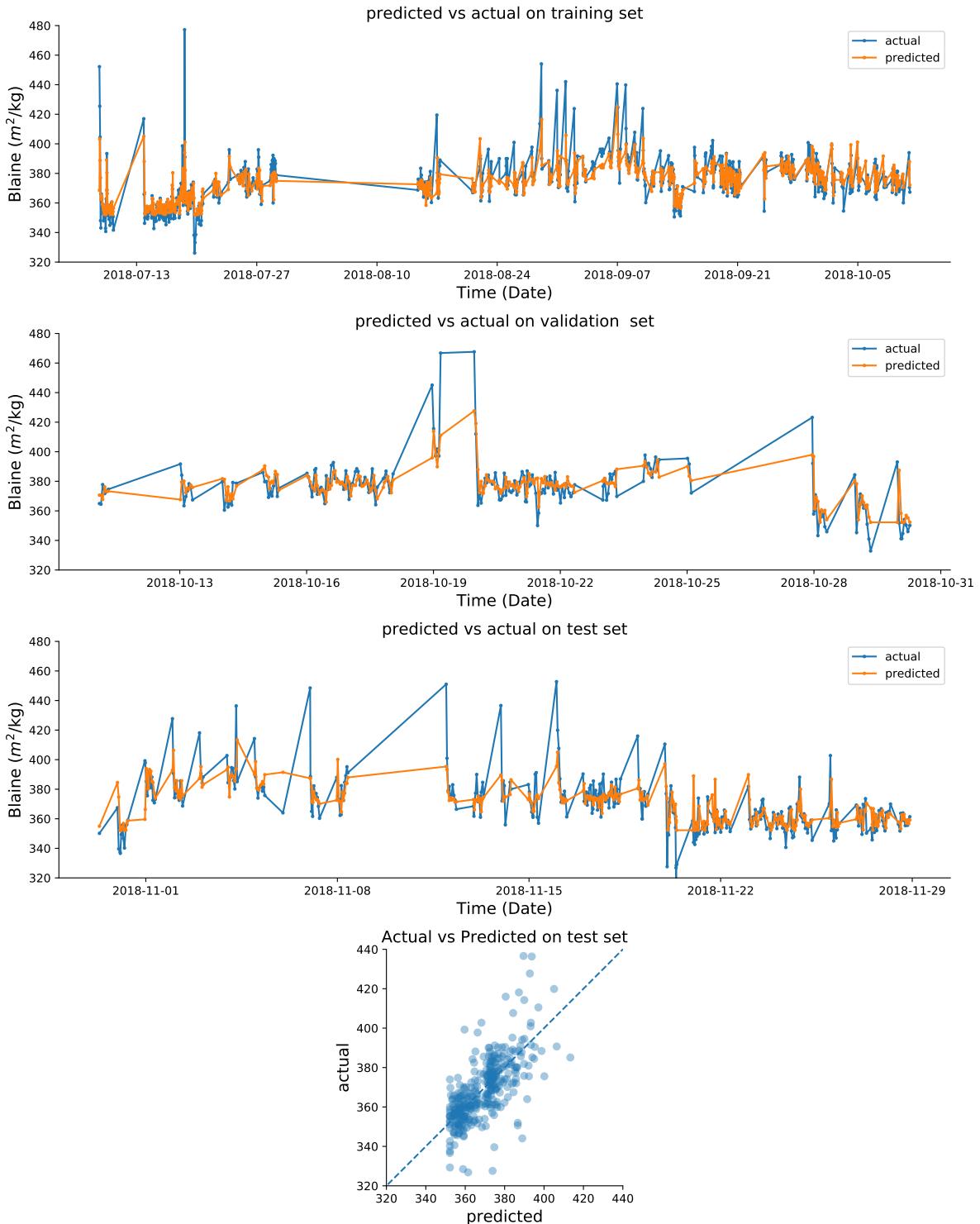


FIGURE 4.1: Various plots showing model performance on the train, validation and test sets for the optimal deep MLP model.

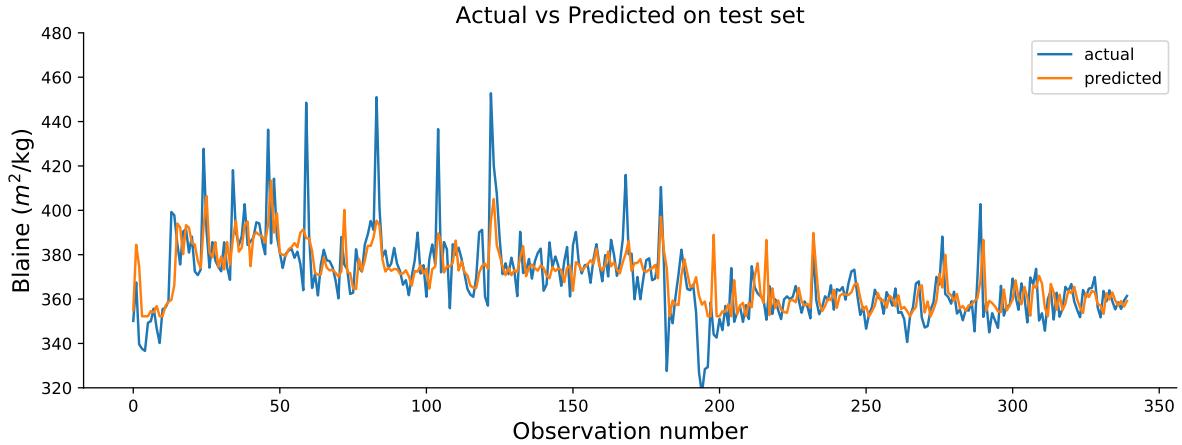


FIGURE 4.2: Repeat of test set plot given in Figure 4.1 with periods of non-operation removed

However, a model using just the last recorded values of Blaine is the persistence model which has a much lower R^2 of 0.141. The difference in performance can largely be explained by looking at the model performance for the lasso regression during the last stages of feature elimination given by Table 4.2.

TABLE 4.2: Accuracies of lasso regression variables as final few features are eliminated

Number of features in model	Train R^2	Validation R^2	Test R^2	Next Feature to be eliminated
1	0.395	0.374	0.294	BLAINE
2	0.476	0.435	0.140	EAMPS_8MinAgo
3	0.504	0.512	0.386	RESIDUAL
4	0.516	0.528	0.407	SEPDAMPER_8MinAgo
5	0.516	0.528	0.407	MAMPS
6	0.516	0.528	0.406	VSEPDAMPER_8MinAgo
7	0.523	0.532	0.416	SEPDAMPER
8	0.525	0.533	0.419	OUTDAMPER
9	0.526	0.537	0.422	RPAMPS

The trained lasso regression model that uses just Blaine has a test set R^2 of 0.294 which is much higher than the persistence model and shows the improvement in performance from having an intercept and allowing for regression towards the mean. Next, adding EAMPS_8MinAgo improves performance on the validation set but causes a drop on the test set accuracy. This suggests that during December, the month that the test set is drawn from, something changes in the system such that the elevator amp reading is no longer a reliable predictor for Blaine in the way it was for all the prior months.

The next jump in performance comes from including RESIDUAL which is another lab tested measure of product fineness. Finally, SEPDAMPER_8MinAgo explains almost all of the remaining performance.

The significant role of the Residual measurements is interesting as the lab tests for Blaine and Residual happen concurrently. Therefore, RESIDUAL as a feature is just as outdated as BLAINE. This suggests that either there is some information about the cement process that can be inferred from fineness but not Blaine alone or there may be error in the measurements for Blaine.

Consider Figure 4.3, which plots Blaine against Residual. Note that the few nega-

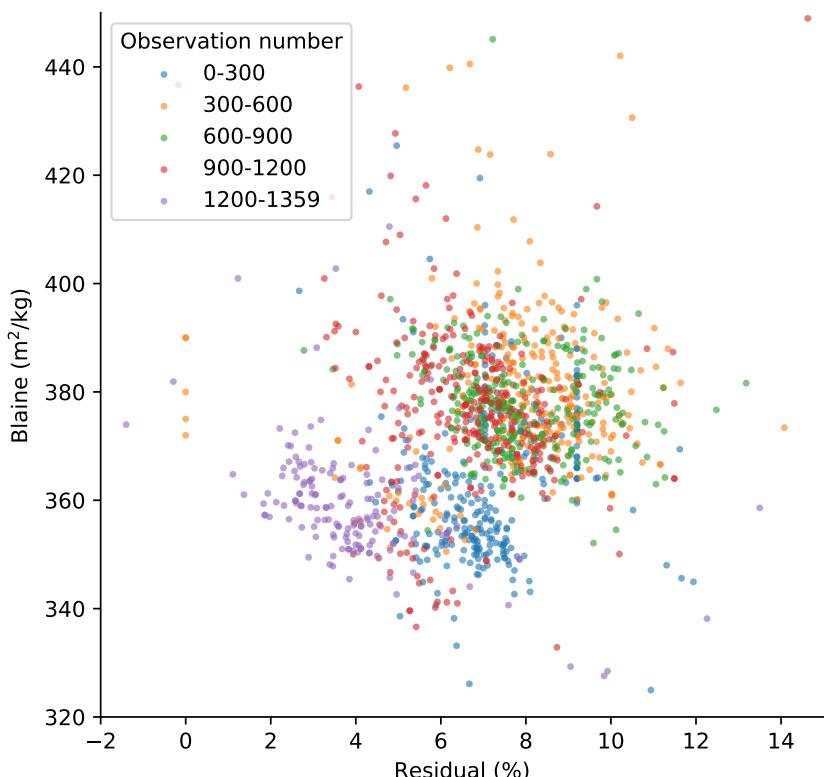


FIGURE 4.3: Plot of Residual vs Blaine for all observations, the observation number is directly related to time

tive values for Residual are invalid outliers and are the result of data capture errors. The data is coloured according to observation number which is ordered in time. The plot seems to suggest various distinct clusters in the data. The clusters also do not seem to move in a constant way through time as the cluster for the first observations to be recorded (blue) is closest to the cluster of the last observations for the period of data capture (purple).

The purple cluster demonstrate the time-variance of the system as most of these observations (which form part of the test set) are outside the range of training set observations. The plot also suggests that there is some discrete change to the cement

grinding process that results in significant changes to the shape of ground particles whereby the relationship between surface area and size changes.

Within clusters the expected inverse relationship between Blaine and Residual seems stronger but still not perfect. Assuming that cement particles have a relatively constant fractal dimensionality, regardless of particle size, one would expect a strong monotonic relationship to exist between the two. The lack of empirical support for this relationship suggests that there may be significant measurement error and/or sampling error. The potential for significant noise in Blaine measurements might be a cause of poor performance in the model for two reasons. Firstly, the models might be fitting to erroneous output values and secondly, even if there was enough data to train a robust model, its measured accuracy might be reported as unfairly poor due to noise in the test set output values.

As a result it might be worth using a compound measurement of Blaine and Residual for cement fineness to reduce the effect of sampling and measurement error. Additionally, further research into building a soft-sensor for this mill circuit might benefit from explaining the cause of the clustering.

A detailed enquiry into Blaine recordings would also be valuable. A useful question to answer would be, how much error exists in Blaine measurements from the robotic lab and what is the breakdown into measurement error and sampling error? Repeated experiments of Blaine measurements from the same sample or different samples would provide an insight on the cause and distribution of errors which might inform filtering algorithms for Blaine data. Another goal for this further research might be to determine the dynamics of the Blaine in the plant. For example, is it reasonable for Blaine to jump by $100 \text{ m}^2/\text{kg}$ in the space of an hour? If not, this information could be used to filter the Blaine measurements. Certainly the ranking of Blaine in Table 4.3 suggests that on the scale of one hour, Blaine has a significant autoregressive nature.

4.2.1 Feature importance

From all of the models trained above a heuristic can be determined for the importance of all the respective features for the purposes of a predictive model. This is provided in Table 4.3 and was derived by assigning a score based on the order of a feature being eliminated during the training of all the models. This measure is crude but begins to provide a picture of which feature might contain more valuable information for predicting Blaine.

BLAINE and RESIDUAL dominate for reasons discussed above and are followed by SEPDAMPER_8MinAgo. For the linear models, SEPDAMPER_8MinAgo had

TABLE 4.3: Average importance of features across predictive models, 1 is most important

1	BLAINE
2	RESIDUAL
3	SEPDAMPER_8MinAgo
4	EAMPS_8MinAgo
5	SEPDAMPER
6	TOUT
7	RPAMPS
8	VAMPS
9	EAMPS
10	SEPSPD_8MinAgo
11	OUTDAMPER
12	VSEPDAMPER
13	VSEPDAMPER_8MinAgo
14	TOUT_8MinAgo
15	SEPSPD
16	MAMPS
17	RPEAMPS
18	BIN_8MinAgo
19	RPAMPS_8MinAgo
20	OUTDAMPER_8MinAgo
21	VAMPS_8MinAgo
22	FEED
23	BIN
24	FEED_8MinAgo
25	RPEAMPS_8MinAgo
26	MAMPS_8MinAgo

a negative coefficient whereas VSEPDAMPER_8MinAgo had a positive coefficient which suggests that directing more airflow to the main separator is related to finer cement.

The power draw of the mill was one of the less favoured variables suggesting less predictive power. However, this is possibly a result of linear models being unable to utilise the non-linear 'n' shape relationship between load and power draw and therefore this variable was dismissed early in feature elimination. FEED is also not favoured, likely a result of the collection bin controlling the flow of clinker into the roller press, rendering the feed rate practically irrelevant.

4.2.2 Training a model without using past fineness measurements

The clustering in Figure 4.3 suggests that the dynamics of the plant are changing over time. The time-varying dynamics of the plant might explain why the best models relied so heavily on recent fineness recordings. Following from this observation an attempt was made to train the lasso regression and MLP-deep on the dataset excluding the BLAINE and RESIDUAL features. The results are presented in Table 4.4.

TABLE 4.4: Accuracies of models trained on data set with no past values of Blaine or Residual

Model	R^2				MAE	
	Train	Validation	Test	on-line	Test	on-line
Lasso regression	0.379	0.255	0.160	0.296	12.410	10.991
MLP-Deep	0.341	0.120	0.331	0.253	9.722	10.368

Table 4.4 shows that the performance has dropped moderately for both models. A more detailed predictive plot for the neural network is presented in Figure 4.4. For the first three plots, the observations are ordered temporarily, but not at a constant scale so as to avoid gaps in the graph when the mill was off-line. Without past values of Blaine, the predictive model starts to look more stable with fewer extreme predictions. The feature selection resulted in only two features, namely SEPSPD and RPEAMPS, which were ranked midway through the list of feature importance given in Table 4.3.

Figure 4.4 shows that the neural network model seems to capture discrete jumps in the process perhaps describing the same phenomenon that drives the clustering in Figure 4.3. This suggests that the clusters might be explained by a detailed look at the relationship between the main separator speed and the circulating load in the pre-crusher circuit.

4.3 Summary

The preliminary tests suggests that the best model is an online retrained shallow neural network or a linear regression depending on whether an absolute error or squared error metric is preferred. Without online retraining the best model was the 'deep' MLP which achieved a test set R^2 of 0.476. However, lasso regression also performed relatively well with a test set R^2 of 0.422.

Furthermore, linear models, such as lasso regression or linear regression might be preferred due to lower computational cost as well as being easier to implement

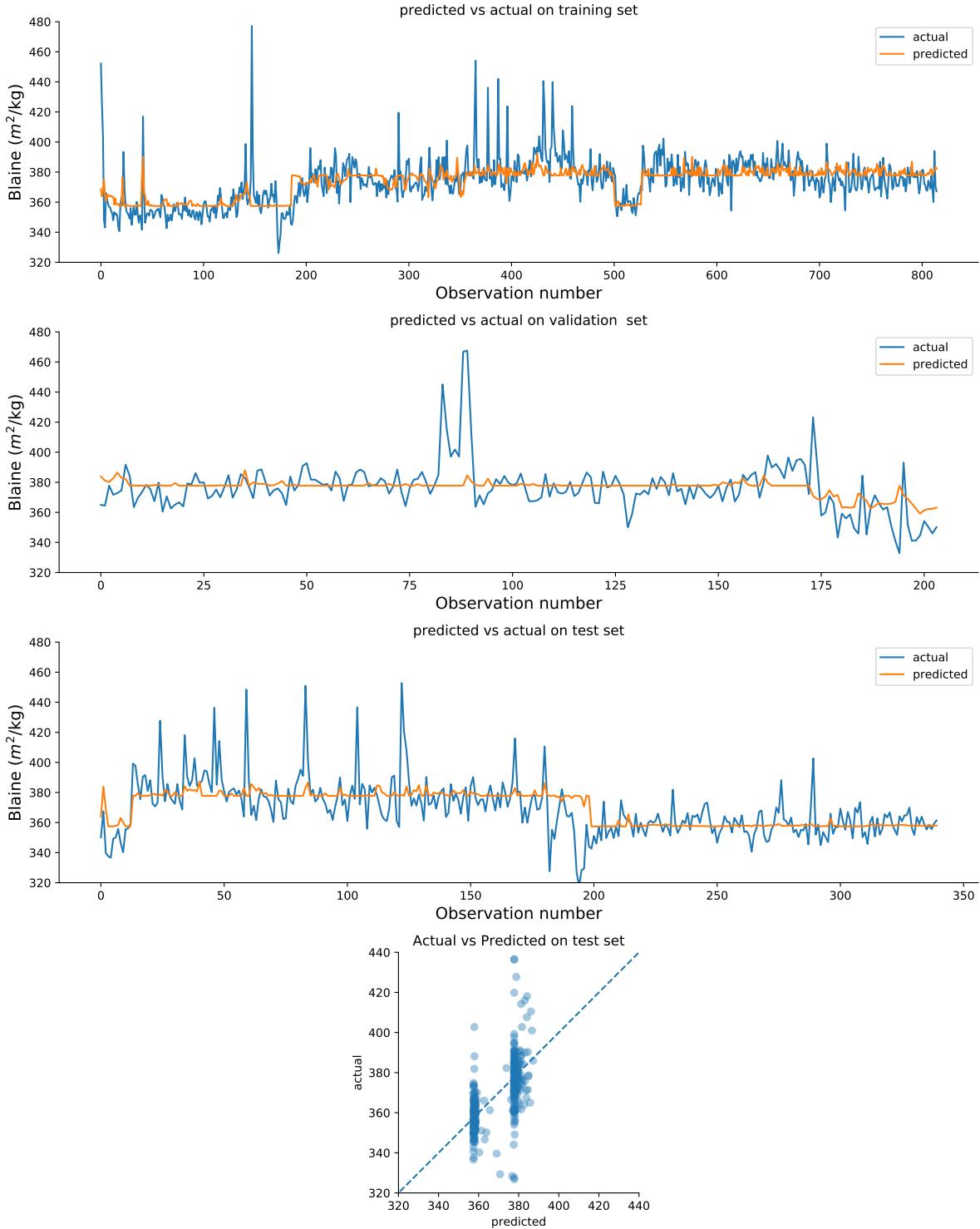


FIGURE 4.4: Various plots showing model performance on the train, validation and test sets for a deep MLP model.

and interpret. There was evidence of time variance in the system, with several models showing a sharp drop in performance for the test set relative to the data used during training. Due to the uncertainty created by an imperfect understanding of the dynamics of the circuit the most robust model might be a linear model, like lasso

regression, trained on as few variables as possible.

Neural network models like the ANFIS and LSTM demonstrated poor performance. It might be possible to improve performance by tweaking the model parameters or data-preprocessing, but preliminary evidence shows poorer performance than linear models and MLP given the same data.

Support Vector Regression using the polynomial and RBF kernels showed moderate performance but did not surpass the highest performing linear model.

There is evidence for discrete changes to the system that can be seen as clustering in the plot of Residual against Blaine. There is also evidence that information for predicting these changes in the system is present in online measured process-variables such as separator speed and amp draw for the elevator in the roller press circuit.

Chapter 5

Conclusions and recommendations

The cement circuit analysed in this paper is highly complex with non-linear dynamics, multiple time-varying instances of feedback as well as important unmeasured variables. There was no existing research on modelling or control for the particular circuit analysed for this research.

Many different data-driven techniques were applied to the problem of trying to create a soft-sensor for Blaine (cement fineness). Almost all models performed better than the persistence baseline but there is still a lot more research that could be done to better understand the cement circuit and to better tweak data driven models for the dataset. Some non-linear black box models show signs of capturing hidden dynamics in the plant, that could be used as a spring-board for building more effective soft-sensor models.

The best R^2 achieved on the test set was 0.481 for an online retrained linear regression. The best MAE achieved was 8.799 for a shallow neural network, this MAE is 5.6% of the total range of recorded values for Blaine. Both models show moderate improvement against a naive persistence benchmark R^2 of 0.154 and MAE of 11.108.

There are several potential avenues of analysis for future soft-sensor research on this circuit. Firstly exploring the change in variable correlations over time using different time lags could assist to create more relevant features. Secondly further exploration is required into the apparent discrete changes in plant dynamics which further requires an explanation of how this might result in changes to particle shape (as measured by comparing Blaine and Residual). Thirdly, the use of a composite fineness measure that combines Blaine and Residual observations can be explored to overcome sampling and measurement error. Fourthly, including data from in-between Blaine lab samples can be explored to increase the amount of information algorithms have access to. Finally an online learning algorithm could be explored as a solution for handling time variance in the system such as using an online .

For plant operators it is recommended that until this circuit is better understood

and modelled, an on-line linear regression based soft-sensor with a few select variables should be used due to ease of implementation, robustness and sufficient performance relative to other models. Following from this conclusion, attempts to apply a data-driven control similar to the systems recently proposed by Zhou et al.[24] and Dai et al.[10] might show limited success as the soft-sensor component is unlikely to be reliable.

Given that the accuracy of predictive models trained on this cement plant dataset show significantly poorer performance than reported by Pani[25], there is evidence that the dynamics of cement mill circuits can differ significantly and that methods and results inferred from one cement mill circuit might not translate successfully to another.

References

1. Birshan, M., Czigler, T., Periwal, S. & Schulze, P. The cement industry at a turning point: A path toward value creation. <https://www.mckinsey.com/industries/chemicals/our-insights/the-cement-industry-at-a-turning-point-a-path-toward-value-creation> (2015).
2. Ali, M., Saidur, R & Hossain, M. A review on emission analysis in cement industries. *Renewable and Sustainable Energy Reviews* **15**, 2252–2261 (2011).
3. Huntzinger, D. N. & Eatmon, T. D. A life-cycle assessment of Portland cement manufacturing: comparing the traditional process with alternative technologies. *Journal of Cleaner Production* **17**, 668–675 (2009).
4. Bentz, D. P., Garboczi, E. J., Haecker, C. J. & Jensen, O. M. Effects of cement particle size distribution on performance properties of Portland cement-based materials. *Cement and concrete research* **29**, 1663–1671 (1999).
5. Monov, V., Sokolov, B. & Stoenchev, S. Grinding in ball mills: modeling and process control. *Cybernetics and information technologies* **12**, 51–68 (2012).
6. Minchala, L. I., Zhang, Y. & Garza-Castañón, L. Predictive Control of a Closed Grinding Circuit System in Cement Industry. *IEEE Transactions on Industrial Electronics* **65**, 4070–4079 (2018).
7. Boulvin, M., Wouwer, A. V., Lepore, R., Renotte, C. & Remy, M. Modeling and control of cement grinding processes. *IEEE transactions on control systems technology* **11**, 715–725 (2003).
8. Casali, A *et al.* Particle size distribution soft-sensor for a grinding circuit. *Powder Technology* **99**, 15–21 (1998).
9. Zhou, P., Chai, T. & Sun, J. Intelligence-based supervisory control for optimal operation of a DCS-controlled grinding system. *IEEE Transactions on Control Systems Technology* **21**, 162–175 (2013).
10. Dai, W., Chai, T. & Yang, S. X. Data-driven optimization control for safety operation of hematite grinding process. *IEEE Transactions on Industrial Electronics* **62**, 2930–2941 (2015).
11. Pani, A. K. Design of soft sensors for monitoring and control of cement manufacturing processes (2015).

12. Du, Y.-G., del Villar, R. & Thibault, J. Neural net-based softsensor for dynamic particle size estimation in grinding circuits. *International Journal of Mineral Processing* **52**, 121–135 (1997).
13. Graves, A., Mohamed, A.-r. & Hinton, G. *Speech recognition with deep recurrent neural networks* in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on* (2013), 6645–6649.
14. Hinton, G. *et al.* Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* **29**, 82–97 (2012).
15. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521**, 436 (2015).
16. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *Imagenet classification with deep convolutional neural networks* in *Advances in neural information processing systems* (2012), 1097–1105.
17. Bojarski, M. *et al.* End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).
18. Ren, S., He, K., Girshick, R. & Sun, J. *Faster r-cnn: Towards real-time object detection with region proposal networks* in *Advances in neural information processing systems* (2015), 91–99.
19. Lake, B. M., Ullman, T. D., Tenenbaum, J. B. & Gershman, S. J. Building machines that learn and think like people. *Behavioral and Brain Sciences* **40** (2017).
20. Xingjian, S. *et al.* *Convolutional LSTM network: A machine learning approach for precipitation nowcasting* in *Advances in neural information processing systems* (2015), 802–810.
21. Ma, X., Tao, Z., Wang, Y., Yu, H. & Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* **54**, 187–197 (2015).
22. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
23. Vinyals, O., Toshev, A., Bengio, S. & Erhan, D. *Show and tell: A neural image caption generator* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 3156–3164.
24. Zhou, P., Lu, S.-W. & Chai, T. Data-driven soft-sensor modeling for product quality estimation using case-based reasoning and fuzzy-similarity rough sets. *IEEE Transactions on Automation Science and Engineering* **11**, 992–1003 (2014).

25. Pani, A. K. & Mohanta, H. K. Soft sensing of particle size in a grinding process: Application of support vector regression, fuzzy inference and adaptive neuro fuzzy inference techniques for online monitoring of cement fineness. *Powder Technology* **264**, 484–497 (2014).
26. Willmott, C. J. & Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research* **30**, 79–82 (2005).
27. Chai, T. & Draxler, R. R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geoscientific model development* **7**, 1247–1250 (2014).
28. Hyndman, R. J. & Koehler, A. B. Another look at measures of forecast accuracy. *International journal of forecasting* **22**, 679–688 (2006).
29. Cortes, C. & Vapnik, V. Support-vector networks. *Machine learning* **20**, 273–297 (1995).
30. Gulcehre, C., Moczulski, M., Denil, M. & Bengio, Y. Noisy activation functions in *International Conference on Machine Learning* (2016), 3059–3068.
31. Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. *Deep learning* (MIT press Cambridge, 2016).
32. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).
33. Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to sequence learning with neural networks in *Advances in neural information processing systems* (2014), 3104–3112.
34. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. & Schmidhuber, J. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems* **28**, 2222–2232 (2017).
35. Jang, J.-S. ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics* **23**, 665–685 (1993).
36. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).
37. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15**, 1929–1958 (2014).

38. Gal, Y. & Ghahramani, Z. *A theoretically grounded application of dropout in recurrent neural networks* in *Advances in neural information processing systems* (2016), 1019–1027.
39. Bottou, L., Curtis, F. E. & Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review* **60**, 223–311 (2018).
40. Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B. & LeCun, Y. *The loss surfaces of multilayer networks* in *Artificial Intelligence and Statistics* (2015), 192–204.
41. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778.
42. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
43. Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Statistics and computing* **14**, 199–222 (2004).
44. Box, G. E., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. *Time series analysis: forecasting and control* (John Wiley & Sons, 2015).
45. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**, 281–305 (2012).
46. Amaldi, E. & Kann, V. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science* **209**, 237–260 (1998).
47. Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine learning* **46**, 389–422 (2002).
48. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
49. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural networks* **61**, 85–117 (2015).