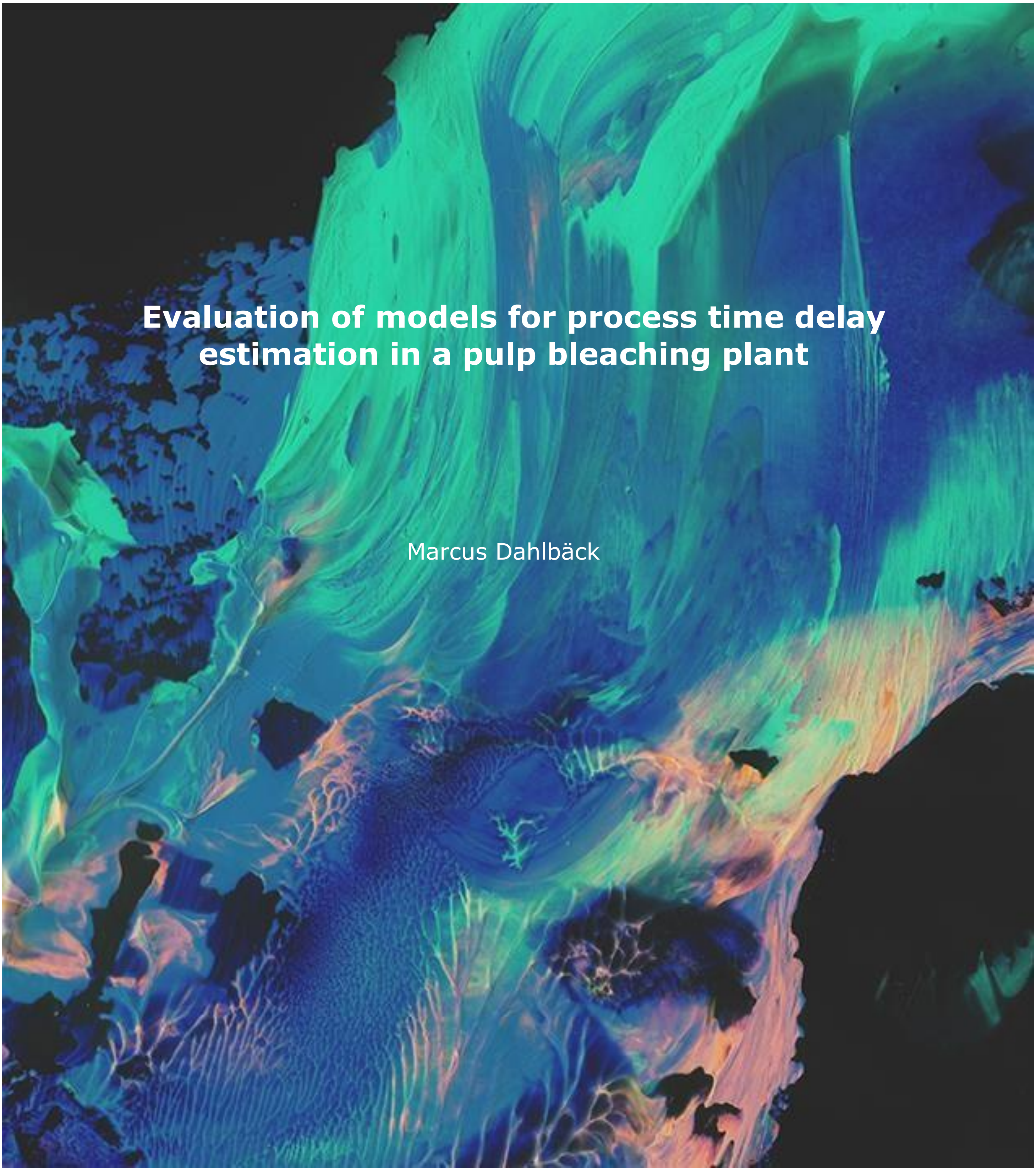
UMEÅ UNIVERSITY



Evaluation of models for process time delay estimation in a pulp bleaching plant

Marcus Dahlbäck

Master's Thesis in Engineering Physics, Department of Physics, Umeå University, 2020

Department of Physics
Linnaeus väg 20
901 87 Umeå
Sweden
www.physics.umu.se

Master’s Thesis in Engineering Physics
Umeå University
June 24, 2020

Author: Marcus Dahlbäck (mada0211@student.umu.se)
Supervisors: Rolf sandberg (rolf.sandberg@metsagroup.com)
Håkan Hägglund (hakan.hagglund@metsagroup.com)
Examiner: Martin Rosvall (martin.rosvall@umu.se)

Abstract

The chemical processes used to manufacture pulp are always in development to cope with increasing environmental demands and competition. With a deeper understanding of the processes, the pulping industry can become both more profitable and effective at keeping an even and good quality of pulp, while reducing emissions. One step in this direction is to more accurately determine the time delay for a process, defined as the time it takes for a change in input to affect the process's output. This information can then be used to control the process more efficiently. The methods used today to estimate the time delay use simple models and assumptions of the processes, for example that that the pulp behaves like a "plug" that never changes its shape throughout the process. The problem with these assumptions is that they are only valid under ideal circumstances where there are no disturbances. This Master's thesis aims to investigate if it is possible to measure the process time delay using only the input and output data from the process, and see if this estimation is more accurate than the existing model based methods. Another aim is to investigate if the process time delay can be monitored in real time. We investigated three methods: cross-correlation applied to the raw input and output data, cross-correlation applied to the derivative of the input and output data, and a convolutional neural network trained to identify the process time delay from the input and output data. The results show that it is possible to find the time delay, but with significant deviations from the models used today. Due to a lack of data where the time delay was measured, the reason for this deviation requires more research. The results also show that the three methods are unsuitable for real-time estimation. However, the models can likely monitor how the process time delay develops over long periods.

Contents

1 Introduction

1.1 Background

Kraft Pulping industry consists of several steps where wood is treated both mechanically and chemically to become pulp. The pulp can then be further processed to a large variety of items. The process of making Kraft pulp consists of many steps that are controlled by process parameters. A process parameter can be both properties of the pulp at a certain point in the process or the amount of additives, for example, that is added to the pulp to change the properties. The task of using the process parameters to control the process such that a particular end result is achieved, is not a trivial task. At the Husum Kraft pulp mill in Sweden, where this study is conducted, the process control strategy has been developed by testing and by experience of what works and what does not work. There are a couple of problems with this approach however. The first problem is that there is no guarantee that the controlling strategy is the most optimal, since it is only based on experience of what works and not. The second problem is that there will always be random variations in the process that must be compensated for, and this compensation can be done more efficiently if we have a deeper understanding of the process.

To control the process more efficiently, knowing the process time is essential. By knowing the process time, it is possible to see what effect a change in input parameters to the process has, since the process time tells when a change in input parameters will have an effect on the output parameters. This is not a trivial task because most processes in a pulping mill runs continuously and not in batches, which have a well defined process time. One way to determine the process time in a continuous process is by a technique called plug following. The main idea in this technique is that we imagine a cross section of pulp that enters a process and then follow that cross section until it comes out at the other side of the process. This is of course a very simplified model, since the original cross section of pulp almost never keeps its shape through the whole process. However, if the properties of the product close to the imagined cross section has similar properties, this model can still be used to determine the process time.

The science of determining the process time or equivalently, the time it takes for the process output to respond to a change in input, is known as time delay estimation (TDE). TDE is a well researched area that has many applications, for example, TDE can be used by sonar and radar systems to detect and localise targets [1]. Attempts to estimate the time delay in industrial processes has also been done previously, for example in [2] a neural network is used for this purpose and in [3] a more statistical model is developed. Common for these papers however, are that the models are not tested on a large scale industrial process, such as a pulp mill.

There are also many different TDE methods available, and a comparison of different methods can be found in [4]. The problem with most of the methods presented there however, is that they are based on a model over the process. This is hard to do in a pulp mill because most processes are very complex and sometimes a bit stochastic. The approach taken in this master's thesis work is instead to try to find the process time without any model of the process and by only looking at input and output parameters of the process. This is done using three methods: The first method is more general and based on a convolutional neural network (CNN), and the other two methods are a bit simpler and based on cross correlation, which is a measure of similarity between two time series. The reason for choosing these three methods is because we wanted to see if the more general CNN model would have any advantage over the simpler but proven cross correlation methods.

1.2 Aim

The aim of this work was first of all to find out if it is possible to calculate the process time using only input and output data from a process. A second aim was to investigate which of the three methods that are most suitable for the task and to see if the result from any of them are more reliable than the method used today. Lastly, we also investigated if the methods can be used as a base for controlling the process in real time.

1.3 Methods and key results

In all tests in the study, we evaluated the three methods on data from the bleaching plant of the Husum pulp mill. The methods were both compared to each other and with a calculation of the process time that is in use today. In order to find strengths and weaknesses of the three methods, we also tested the performance on different data sets that has different properties. The results show that it is possible to find the process time using only input and output data, however there are large differences when the three methods are compared to the calculated process time. The testing on different data sets show that all three methods are sensitive to the filtering of the raw data, and that different filtering is needed on different parts of the data to get better predictions. The results also show that all methods are unsuitable for real time process control because they need about a week of measurements in order to make accurate predictions. This, however, can be used as an advantage when calculating the average process time over long time periods, and there are promising signs that the methods can be used as a tool to monitor how the process time changes over long time periods.

2 Theory

2.1 Basics of pulp

Wood can be thought of as a composite material of lignin and cellulose, where the cellulose acts as a reinforcing fibre and the lignin as a matrix that keeps the cellulose in place. Pulp is the fibrous material that is created when the cellulose is separated from the lignin. The main problem when making pulp therefore consists of how to remove as much of the lignin from the cellulose as possible without damaging the cellulose. This can be done in two main ways, chemically, which gives the best quality in terms of brightness, and mechanically, which is cheaper but does not yield as good brightness. There are also multiple different chemical methods, however the most common is the sulphate process which gives a pulp known as Kraft pulp.

2.2 The sulphate/Kraft pulping process

The sulphate process begins with that trees are debarked and then turned in to wood chips that are 12–25 millimetres long and 2–10 millimetres thick. The wood chips are then mixed with chemicals, primarily Na_2S and NaOH, and fed in to a boiler which is also known as a digester. In the digester, the chemicals react with the wood and dissolves the lignin from the cellulose, which can then be collected. Adding more chemicals or using longer cooking times will cause more lignin to break down, however it will also cause the cellulose to break down. This means that there is always a trade off between a low lignin content and more damaged cellulose fibres, or higher lignin content and less damaged cellulose fibres. The lignin content in the pulp can be calculated by the kappa number of the pulp. The kappa number is defined by how many millilitres of a potassium permanganate solution that one gram of the pulp will consume [5], however this is very strongly correlated to the lignin content, where a lower kappa number correspond to a lower lignin content.

Since the lignin has a negative impact on the brightness of the pulp, it needs to be reduced further after the digester stage. This is done through a process known as bleaching. Figure 2.1 below shows a schematic view of the bleaching process.

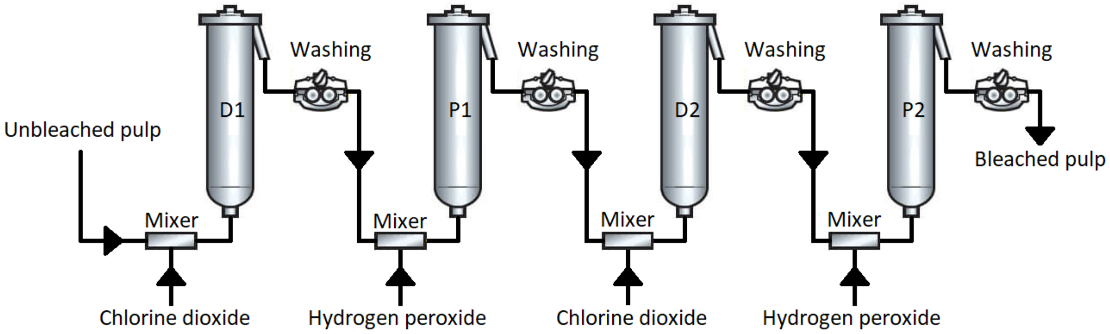


Figure 2.1 – Schematic view of the bleaching plant which shows the four stages where chemicals react with the pulp and reduce the lignin content. Between the stages residual lignin and chemicals are washed away in a washer.

The bleaching plant is divided in to four separate stages which we call D1,P1,D2 and P2 (fig 2.1). Each stage works on the same principle, first one of the chemicals, chlorine dioxide (ClO_2) or hydrogen peroxide (H_2O_2), is added to the pulp in a mixer. Then the mix is pumped to a large tank called a bleaching tower where the chemical and pulp reacts as it slowly rises to the top of the tank. Once the mix reaches the top, the reaction is complete and residual chemicals and lignin are washed away with water in a washer. The water that is used in the washers is also recycled backwards in the process, i.e washing water from the D2 washer is used in the D1 washer and similarly for the P-stages. This adds extra complexity to the process since the amount of chemicals used in the last two stages might affect the first two stages. We can also see in figure 2.1 that the pulp is mixed alternately with ClO_2 and H_2O_2 in the four stages. The reason for this is because it results in a higher brightness of the pulp compared to if only one ClO_2 - stage and one H_2O_2 - stage were used. A more thorough explanation of the chemistry and construction of a bleaching plant can be found in [6]. After the bleaching plant, the process of creating Kraft pulp is complete and the pulp can then be further processed to for example white cardboard or printing paper.

2.3 Artificial neural networks

2.3.1 Fully connected neural networks

An artificial neural network is inspired by how neurons are wired in the brain of humans, and consists of at least 2 layers called the input layer and the output layer. Between these layers, it is common to have additional layers called hidden layers. A layer in turn consists of one or more nodes, which are called neurons. The most basic type of neural network is the fully connected feed-forward network. In this type of network, each node in one layer is connected to each node in the next layer and so on. A schematic view of this can be seen in figure 2.2 below.

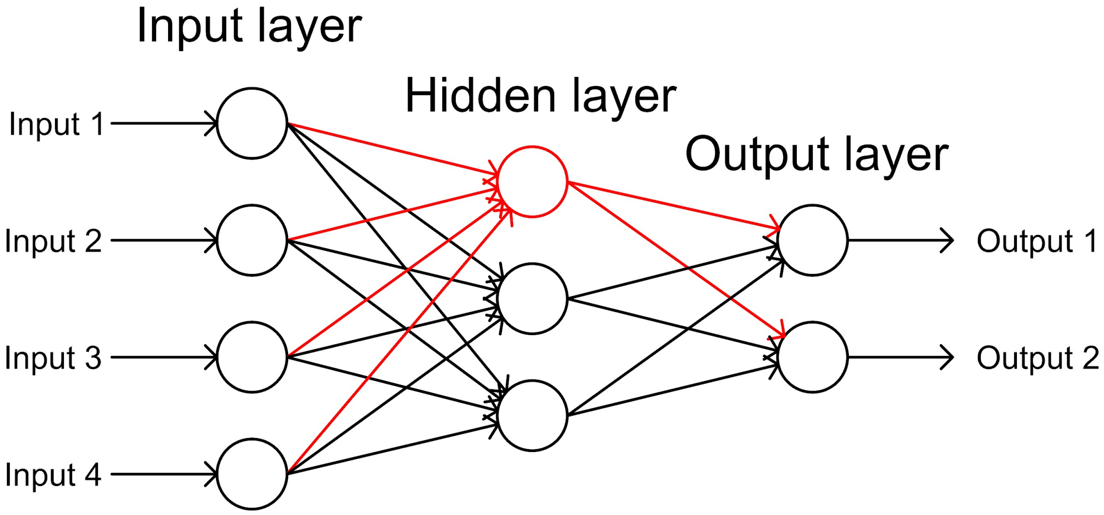


Figure 2.2 – Schematic view over a feed-forward neural network with 4 inputs, one hidden layer with 3 neurons, and 2 outputs. The inputs and outputs of one of the hidden neurons are marked in red.

Each neuron in the network consists of one weight for each input to the neuron, and an activation function. If we have n inputs to a neuron, the output, y is calculated as

$$y = f\left(\sum_{j=1}^{n} x_j w_j\right) \tag{1}$$

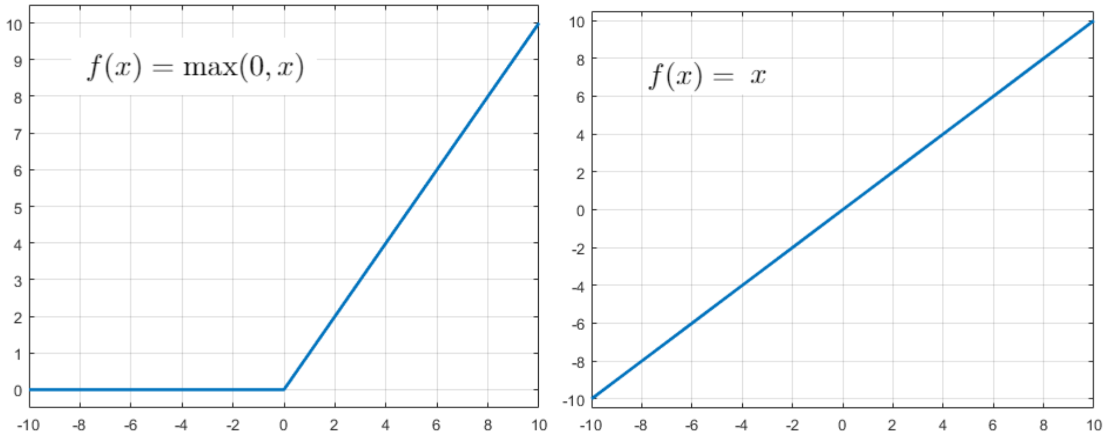
where f is the activation function and w_j is the weight associated with input x_j. Similarly, the output of a layer, $\mathbf{y}$, in the network can be calculated as

$$\mathbf{y} = f(\mathbf{W}\mathbf{x}), \tag{2}$$

where $\mathbf{W}$ is the weight matrix that contain the weights for neuron i on row i, and $\mathbf{x}$ is the input vector. Finally, if a network has m layers, the output, $\mathbf{y}_{out}$, and the input, $\mathbf{x}_{in}$, are connected by

$$\mathbf{y}_{out} = f\left[\mathbf{W}_m\left(f\left(\mathbf{W}_{m-1}(...f(\mathbf{W}_1\mathbf{x}_{in}))\right)\right)\right] \tag{3}$$

The purpose of the activation functions are to introduce non linearity to the network. It can be seen from equation (3) that if no activation functions are used, the equation can be reduced to a series of matrix multiplications. This means that a network with more than one hidden layer can be reduced to contain just one hidden layer. There are many activation functions available, however two of the most common are the Rectified Linear Unit (ReLU) and the linear activation functions. These are shown in figure 2.3 below.



(a) **(b)**

Figure 2.3 – ReLU activation function (a) and linear activation function (b). The Relu activation function is used in all layers except the last due to its non-linearity. The linear activation function is used in the last layer when we want the output to be able to take any value, for example when performing a regression task.

We can see in figure 2.3a that the ReLU activation function changes the most around $x = 0$. The reason for this is to mimic the behaviour of neurons in the human brain, where neurons will or will not activate to a given input. The linear activation function in figure 2.3b is mostly used in the last layer of the network when performing a regression task. There would be no point in having a linear activation function elsewhere in the network since this would not add the necessary non linearity to the network. Thus, the ReLU activation function is commonly used on all layers except the last due to its non linearity.

2.3.2 Convolutional neural networks

When creating a neural network for classifying objects in an image for example, one approach could be to use one input neuron in a fully connected network for each pixel in the image. This approach would however cause some problems. Firstly we would need a large amount of input neurons, even for small images of size 500x500 pixels we would need 250000 input neurons. Secondly and most importantly, it is usually not the individual pixel values that are important, but rather the correlations between the pixels, since this is what defines objects in the image. A convolutional neural network (CNN) aims to solve this issue by using filters, called kernels, which extract features from the image. A kernel can be seen as a matrix containing weights that are updated during the training of the network. The network layers can be visualised as in figure 2.4 below.

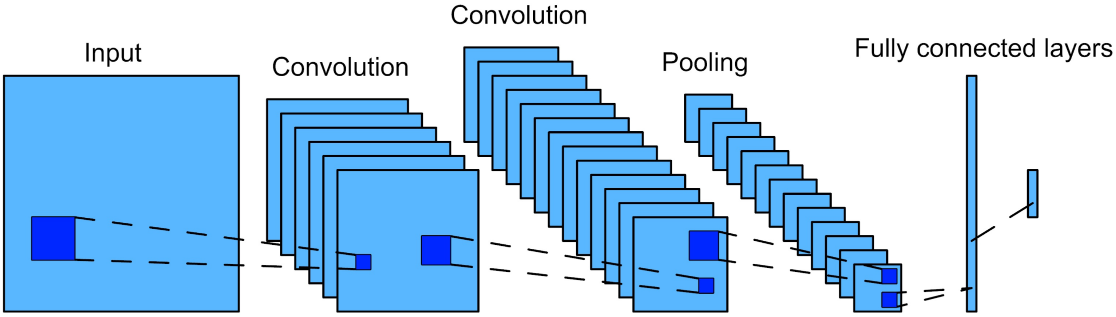


Figure 2.4 – Illustration of a convolutional neural network with convolutional and pooling layers. The convolutional layers extract features from the input data, which can be seen as the depth in the image. The pooling layer reduce the spatial dimensions of the data to make the network less computationally heavy. The fully connected layers at the end of the network performs the prediction based on the extracted features.

We can see how the input image first gets reduced in size spatially by convolution (fig 2.4). Convolution consists of moving the kernel across the image in steps, and for each step multiplying the pixel values encompassed by the kernel, with the weights in the kernel and summing the result. The number of positions that the kernel is moved between each step is called the stride. Each step thus gives a value which is placed in a new matrix, called a feature map. Often a CNN uses more than one kernel which means that we get more than one feature map. This can be seen in figure 2.4 as the depth of the layers. The convolutional layer thus acts as a feature extractor since each new feature map contains one feature of the input image. It is also common to have more than one convolutional layer in the network, in figure 2.4, there are two convolutional layers after each other, however this is completely arbitrary. Another common layer in a CNN is a max pooling layer. This layer helps reduce the spatial dimensions while still keeping the most important information. The max pooling is done by moving a kernel over the feature maps, however instead of doing the multiplication, only the largest value encompassed by the kernel will be saved to a new matrix.

After the pooling layer, more convolutional layers can be added, however as some point we also need to do a prediction. This is done by feeding the feature maps to a fully connected neural network. In this way, the extracted features that contains dependencies and patterns in the image are used for the prediction, rather than the pixel values themselves.

2.3.3 Training neural networks

When training a neural network we need to know what the output of the network should be, i.e we need ground truth data. The training of the network can be done by a method called backpropagation. Details of how this method works can be found in [7], however the idea is to first predict some output values with the network, and then adjust the weights using the errors between the ground truth data and the predictions. The number of predictions that is done before updating the weights is called the batch size.

The training of the network is often done in epochs, where one epoch is one presentation of all training data to the network. It is also common to split all training data in to a training and validation set. The network is then trained using the data in the training set only, and the performance of the network is evaluated on the validation set. The reason for this is to be able to detect overfitting of the network. The risk of overfitting the network can be further reduced by introducing dropout layers. Each time a batch is predicted during training, the dropout layers ignores, or drops, the output from the neurons with a certain probability. This forces the network to learn more robust features.

2.4 Cross correlation

A common method in signal processing for finding the delay between signals is cross correlation. Cross correlation is a measure of similarity between two functions, and for non-complex discrete functions f and g, it is defined as

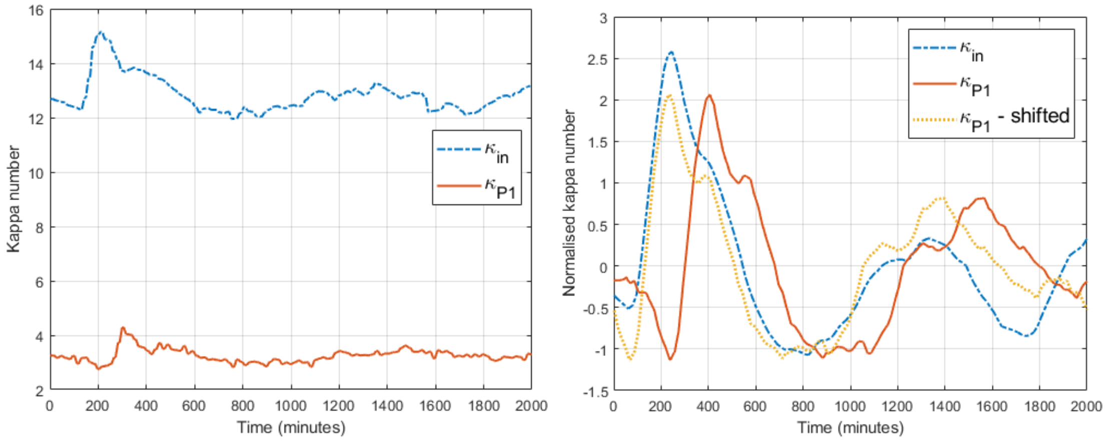
$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[m+n]. \tag{4}$$

Equation (4) can be interpreted as taking the dot product between the functions f and g, but where g has been shifted in time by n samples. If equation (4) is computed for all n, the cross correlation value will peak at the n where the two functions f and g are the most similar. Thus, to find the time delay (in number of samples) between two functions we need to find the n for which the value of the cross correlation is at a maximum.

3 Method

3.1 Variable selection and data preprocessing

To find correlations in the incoming and outgoing data, the first step is to select variables that affect each other. In the bleaching process, one variable that is measured before and after a bleaching step is the kappa number. Because of this, we used the kappa number of the pulp as it enters the bleaching step (κ_{in}) and the kappa number after the first peroxide stage (κ_{P1}) for the analysis. We also did some preprocessing of the data in order to more easily find correlations. The first step was to normalise the data by subtracting the mean and then dividing by the standard deviation. After this, noise was reduced by applying a moving average filter with a window size of 100 samples. The effect this has on the data can be seen in figure 3.1 below.



(a) **(b)**

Figure 3.1 – Kappa number as it enters the bleaching plant (κ_{in}) and the kappa number after the first peroxide stage (κ_{P1}), both before preprocessing (a) and after preprocessing (b). Panel (b) also shows κ_{P1} after it has been shifted 170 minutes back in time.

The incoming kappa number is higher than the kappa number after the first peroxide stage (fig 3.1a). This is reasonable since the purpose of the bleaching process is to reduce the kappa number. It is hard, however, to see any correlation between the two parameters. If we instead normalise the data, it is easier to see that the signals correlate. For example if we look around 100 minutes in figure 3.1b, the incoming kappa number can be seen starting to increase. Then, about 170 minutes later, the kappa number after the P1 stage also start to decrease. By shifting κ_{P1} 170 minutes back in time, we can see that the shifted signal correlate well with κ_{in}. Thus it is possible to manually determine the process time to about 170 minutes, since it takes this time for a change in the incoming data to affect the outgoing data.

3.2 Time delay using cross correlation

Predicting the time delay using cross correlation can be done fairly straightforward by applying equation (4) on sequences of κ_{in} and κ_{P1} - data, and searching for the n that maximises the cross correlation value. When using cross correlation directly on the data however, the difference in value between the two time series are considered to some extent by the model. In our application, it does not matter if the differences in value are large or small, as long as the trends of the two time series are matching. This means that we are more interested to find when the sign of the derivatives of the two time series matches the best rather than when the original data matches the best. Because of this, a new data set was created that contains ones for every sample where the derivative is positive, and zeros where the derivative is negative. This data set was then filtered once more using a moving median filter with a window size of 50 samples. By filtering in this way, we remove some noise that is caused by calculating the derivative on data that is not perfectly smooth. The time delay was then also predicted with the cross correlation method on this data set, in the same way as for the original data.

One aspect when predicting the time delay with cross correlation is how many points to use for the prediction. Using too few points means we will not have enough data to see any patterns, and using too much data will be more computationally demanding. In order to find the most optimal number of samples in this situation, we first collected data when the bleaching plant operated with one specific production rate, i.e. the amount of pulp that is bleached per time unit is constant. Under these circumstances, the process time should be constant if we assume nothing unexpected happens in the process. This means that we can use the variance (or standard deviation) of the predicted time delays of this data as a measure of how stable the methods are. We then predicted the time delay on 700 intervals of this data using 1000 up to 20000 points for the prediction. By plotting the variance against the number of samples we get the result shown in figure 3.2

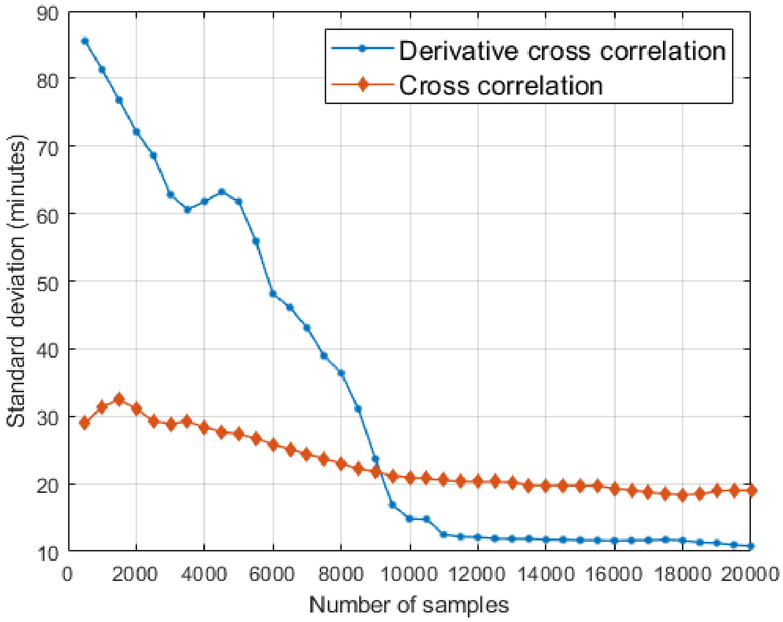


Figure 3.2 – The standard deviation as a function of how many samples that are used in the prediction. The standard deviation was calculated based on 700 predictions of data with a fixed production rate.

The standard deviation drops for both methods until 11000 samples are used for the prediction (fig 3.2). Using more than 11000 samples does not seem to give a significantly lower standard deviation. Because of this, we used 11000 samples in all subsequent tests of the two cross correlation methods.

3.3 Time delay using a neural network

The third method we used for predicting the process time was a convolutional neural network. We tested a network with one-dimensional convolution initially, however early tests showed that it was hard to get good results using this architecture. Instead a network with 2D convolution was used because it showed better results. In order to use a network of this type, the input data must have the form of a matrix. This was done by taking out an array of 11000 points from both κ_{in} and κ_{P1} and reshaping the arrays to a 3D-matrix with size 40 x 275 x 2. The reason that the depth is 2 in this matrix is because we have two variables, each represented by a 40 x 275 matrix. If we would want to use more than two variables, they could be added by simply increasing the depth of the input matrix. The choice of using 11000 points for the prediction was based on the conclusions from figure 3.2. Although it is only valid for the cross correlation methods, it still tells us something about how much data that is needed in order to have significant correlations and it serves as a good starting point for the neural network.

We based the construction of the network on the structure of AlexNet [10], which is a very famous convolutional neural network for image classification. The AlexNet network is constructed for input images with size 227 x 227 x 3, however this is too large for the 40 x 275 x 2 matrix we want to use as input. It is not possible to set the input to be 40 x 275 x 2 in AlexNet either, since this results in negative dimensions at some point due to the fact that AlexNet contains 8 convolutional and max pooling layers, which each reduce the spatial dimensions of the image. In order to solve this, some of the convolutional and max pooling layers were removed and the kernel sizes were changed on the remaining layers. This gave the network structure shown in figure 3.3 below.

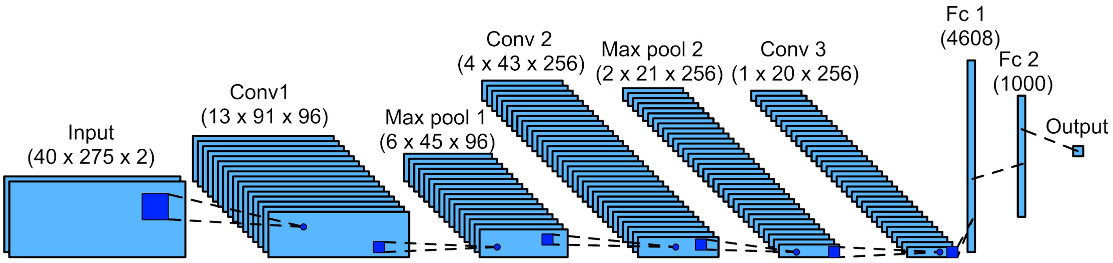


Figure 3.3 – The structure of the convolutional neural network used for the time delay estimation. There are three convolutional layers (Conv1-3) with two max pooling layers (max pool 1-2) in between. Lastly there is a fully connected network with one hidden layer that gives the predicted time delay as output.

The network is made up of three convolutional layers (Conv1-3) with max pooling layers (Max pool 1-2) in between (fig 3.3). After the third convolutional layer, the prediction is done by a fully connected network containing one hidden layer. ReLU activation functions are used for each of the three convolutional layers and the first two layers of the fully connected network. The output layer contains a linear activation function, since during training we want the network to classify both positive and negative time delays. A more detailed view of the network can be found in table A.1 in appendix A.1.

3.4 Training the neural network

We created training and testing data for the network by first dividing all data in to two sets, where the first 90% of all data was reserved for training and the last 10% was used for testing the trained network. In order to train the network, we need data where the process time is known. We did this by first taking out portions of the larger data set where there are visible correlations between the two parameters, and then shifting the κ_{P1} data backwards in time until the two time series overlaps the most. Since the data now has a time shift of 0 minutes, a data set used to train the network could be created by first randomly selecting a sequence with 11000 continuous samples from the κ_{in} data, and then choosing a sequence from the κ_{P1} data that has equal length but is shifted in time by anywhere from -250 to +250 minutes compared to the κ_{in} sequence. The reason that 250 minutes was chosen is because we know that the process time is less than this. In total, 28400 training samples were created in this way, and 80% of those were then used for the actual training of the network and 20% was used for validation during training. The network was then built and trained using Tensorflow [8] with Keras deep learning API [9].

One problem that arise when training the network is how many epochs we should train the network with to get a good result. Using too few epochs means there is a risk of underfitting the model and too many epochs brings the risk of overfitting the model. To solve this, we can look at how the root mean squared error (RMSE) of the model output and the known delay changes as the network is trained. This is shown in figure 3.4 below.

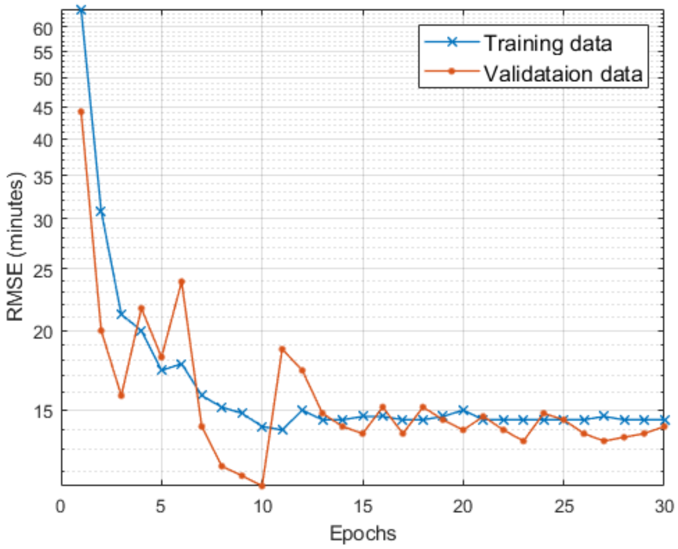


Figure 3.4 – The root mean squared error of the neural network outputs and the known values during training. It can be seen that training with more than 15 epochs does not yield a significantly lower error

The RMSE drops rapidly during the first five epochs and then stabilises (fig 3.4). After about 15 epochs, the RMSE does not drop significantly more and there is thus no advantage in training the network any more. Because of this, the network used in the time delay estimation was trained with 15 epochs, this gave a training RMSE value of about 15 minutes.

3.5 Testing of the methods

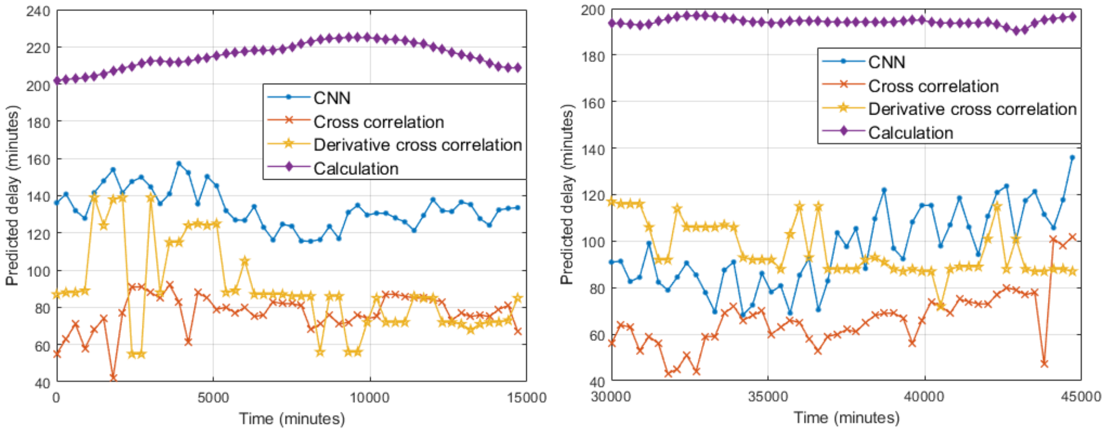
Testing of the methods was performed on the 10% of data that was not used for training the network. The reason for avoiding the training data is because the neural network generally performs better on the training data than on the testing data and thus it would get an advantage against the other methods if the training data is used. We compared the performance of the methods in three main ways: The first comparison is how well the methods correlate with each other and with a calculation of the process time in used today. This calculation is based on parameters such as the volume of the system, the concentration of pulp in the system and the rate at which pulp is being bleached. More details of this calculation can be found in appendix A.2. The second comparison is made based on data where the production rate is constant in order to see how stable the methods are. Lastly, a comparison is made based on portions of data where there are visibly large correlations, such as in figure 3.1b.

4 Result

The performance of the three methods are illustrated mainly using figures. We were not able to measure the exact process time and as a consequence, the results will focus mainly on comparisons between the models and the calculated process time, on different data sets.

4.1 Predictions on the testing data set

The results from predicting the delay on the last 90% of the data resulted in figure 4.1 below. In total, 50 predictions were made with 300 minutes between each prediction. This was also done with two different starting points in the data set, one which starts on the first sample in the data set, and the other which starts 30000 minutes later. The calculated time delay is presented as the average value of the 11000 minutes that the methods use for the prediction.



(a) **(b)**

Figure 4.1 – Comparison between the three methods and the calculation of the time delay. Panel a shows the predictions from the first sample in the test set and 15000 samples forward, panel b show the predictions after 30000 minutes and 15000 minutes forward. The results show a substantial deviation between the methods and the calculated time delay that is used today

All methods deviate significantly compared to the calculated time delay, for most of the predictions, the calculated time delay is about twice as large compared to the two cross correlation methods (fig 4.1). The neural network generally predicts values closer to the calculation, however there are still large deviations. We can also see in both figures that there does not seem to be any agreement between the calculated value and the other methods. For example in figure 4.1a, we can see that the calculated time delay increases slightly until around 10000 minutes and then starts to decrease. This trend can not be seen for any of the three methods, which means that the they both predict different time delays and a different trend of the time delay. Thus, the methods are not biased such that adding a constant value would create a better fit. The predictions are also varying a lot compared to the calculated value, which makes it harder to find trends. From figure 4.1a there also seem to be a difference between the method based on the convolutional neural network and the two methods based on cross correlation, where the cross correlation methods tend to predict values that are more similar. This, however, can not be seen as clearly in figure 4.1b, which indicates that the methods perform differently on different parts of data.

4.2 Predictions on data with strong correlations

Performing the same prediction but with sections of data that has visibly strong correlations gives the result in figure 4.2 below. The time delay was also predicted manually on this data as described in section 3.1.

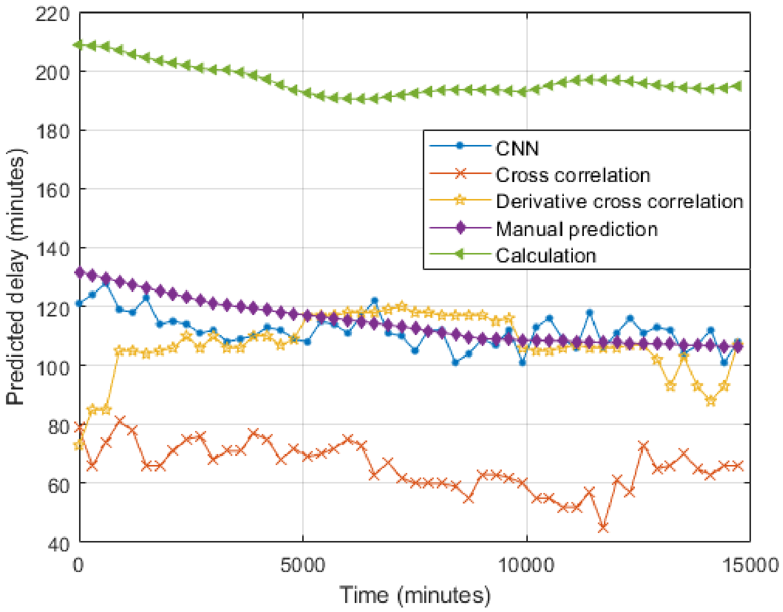


Figure 4.2 – All three methods compared to a manual prediction of the time delay on data where there are visibly good correlations. The results show that the neural network tend to follow the manual predictions the best.

The neural network and the manual prediction tend to follow each other the best (fig 4.2). This is reasonable since the training data for the network was created by manually predicting the delay in the same way. The method that use cross correlation on the derivative data also tends to follow the manually predicted values. The reason for this is most likely because there are stronger correlations in the data used here, compared to the data used for the predictions in figure 4.1. We can also see that the cross correlation method directly on the data differs from the other two methods, and that the large deviations between the three methods and the calculated time delay, are still present.

4.3 Predictions on data with fixed production rate

If the predictions instead are done on data taken from periods where the bleaching plant runs with a specific production rate, the result shown in figure 4.3 below is obtained. Here, 30 points are predicted with 300 minutes in between.

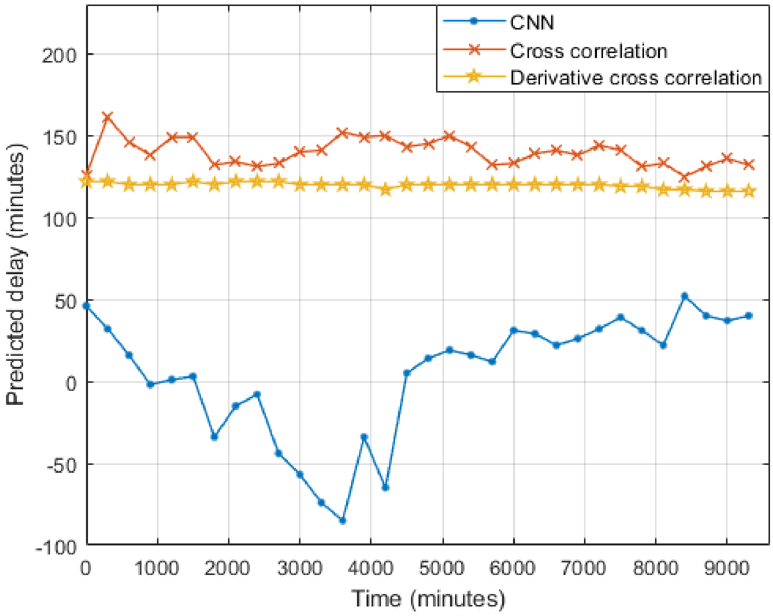


Figure 4.3 – Predicted time delay when all data has been taken from one fixed production rate. The results show that the cross correlation method on derivative data is the most stable method. The neural network was found to perform very bad because of the way the data with constant production rate was generated.

The cross correlation method based on the sign of the derivative gives the most consistent value (fig 4.3). The method that use cross correlation directly on the data tend to predict similar values but with more variation. The convolutional neural network, on the other hand, performs very bad on this data set. The reason for this was found to be because it does not exist a continuous section of data that contains 11000 points with a fixed production rate. In order to get the necessary length, data was taken from different points in time where the production rate is the same, and then added together. By allowing a larger span of possible production rates, such that 11000 continuous points can be found, the neural network predicts values more similar to the cross correlation methods. This shows that the neural network is sensitive to disruptions in the data, which might occur for example if there would be interruptions in the measurements.

5 Discussion

In this study, we successfully derive the process time delay by only analysing patterns in the measured data as the pulp enters and exit parts of the bleaching process. However, the predicted time delay from the three tested methods are substantially lower compared to the predictions used today. The fact that we can see this from both figure 4.1a and 4.1b tells us that the deviation is not just a random event that happens at a specific point in the data. There is no other way for the pulp to travel between the two kappa number measurement points, than through the bleaching towers. This means that the lower time delay can not be explained by influence from another system that allows pulp to flow faster between the points.

One possible reason to why we get the deviating results is that our assumption that the time delay can be measured by following a plug through the system is incorrect. We might for example have diffusion in the bleaching towers which mixes the pulp in the imaginary plug, with pulp further up in the towers. In this way, a change in incoming kappa number might travel faster through the system than we would expect a plug to do. The three methods tested here would be able to detect this, however the calculation in use today would not, because it is completely based on a plug flow. Another reason might be that the calculation used today have incorrectly set parameters, in particular the concentration of pulp in the process. By lowering the concentration to 5-10%, the calculated time delay fits better with the three methods tested in this study. Although this could be plausible explanations, we also have to consider that the calculated time delay is what is used today and it has thus been proven to work.

The fact that 11000 samples of data is needed to make accurate predictions is also a limiting factor to how the three methods can be used. 11000 samples correspond to 11000 minutes, which is about 7.6 days. Since the production rate in the bleaching plant changes every day, this means that the predicted time delay is an average of all measurements during the 7.6 days. Because of this, the methods can only be used to observe how the time delay changes over long time periods, and not for real time applications. The reason that 11000 points is needed is because the process changes very slowly, sometimes the kappa number can be on the same level for several hours at the time, and it is thus impossible to find patterns if too few points are used.

The daily changing production rate also creates some problems when manually preparing data for the neural network. Since the process is slow changing, at least 5000 points have to be used to be able to find the delay, however the production rate changes a few times during this period. This makes it hard to determine the correct time delay, since some manual averaging has to be done each time the time delay is determined. A better approach for creating the training data would be to directly measure the time delay in the process. However, this would require a lot of work because the way the process time is measured, is by releasing a tracing substance in the process, and then taking manual samples in another part of the process until the substance arrives. Since over a year of data is used to train the network, this would be very impractical to perform in reality.

The reason for choosing a neural network as one of the methods, is because of its generality. The cross correlation methods are limited to the behaviour of the data since they will only work with two patterns. The output parameter from the process must either follow the input, which is the case we have with the kappa numbers in this study, or be inverted compared to the input, meaning that an increasing input value results in a decreasing output value. This because we can only search for the cross correlation value when the signals matches the best, or when they match the worst, which would be the case for an inverted relationship. The cross correlation methods also lack the ability to allow more than two parameters for the prediction, which we might want to have if the correlations are not as clear as in this study.

All the disadvantages that the cross correlation methods has, are not a problem for the neural network. In theory we should just have to provide enough training data, and use as many parameters as we want and the network will find the patterns in the data that are best for the predictions. We are therefore not dependent on a particular relationship between the input and output data here, the only requirement is that there is a relationship that is not random. This makes the neural network a very general method to predict the time delay, and theoretically it could be applied on any other process for the same purpose, given the training data. However, all this comes with the great difficulties in producing enough training data to train the network.

It is also hard to determine what method that fits the bleaching plant the best due to a complete lack of data where the time delay has been measured. In terms of stability, the cross correlation method based on derivative data seem to be the most stable from figure 4.3, however figure 4.1a and 4.1b shows significant oscillations during some time periods, with a peek to peek value of at most 80 minutes, which does not seem reasonable. The reason for the oscillations is most likely due to the fact that the derivative is very sensitive to noise and we thus have to filter the data appropriately for it to work. The problem with this however is that there is no filtration that works best on all data. We can see this from the fact that in figures 4.2 and 4.3, there are not as violent oscillations, even though the same filtration is used. The only difference between these two figures and figure 4.1 is the data set the prediction is made on. The method that seem to be most stable on all tested data sets is the cross correlation used directly on the data. This method generally also predicts values the furthest from the calculated value, however compared to the other methods, it generally does not have as violent oscillations.

5.1 Future work

When working forward with the methods tested in this study, there are a few aspects that can be further researched. The first and most important aspect is to investigate why there is a large deviation between the method used today and the three methods tested here. A first step in this would be to see whether the concentration of pulp in the bleaching plant is set correctly in the method used today.

Another aspect that can be further looked into is the construction of the neural network. It is more intuitive to use one-dimensional convolution when you have one-dimensional data, and there is no obvious reason to why two-dimensional convolution should work better. We therefore suggest that this is investigated further in a future study. The filtering of the data is another aspect that can be further developed. A better approach than a moving mean filter might be to develop a filtration model that adapts to the data and filters it based on the current level of noise. Lastly, it would be interesting to test if the models can be used to monitor the development of the process time over longer time periods, rather than for real time applications.

6 Conclusion

We found that it is possible to derive the process time by only analysing data from two points in the process. However, the predicted process time from all three tested methods deviate substantially from the calculation that is used today. It is hard to know whether the three methods used here are more correct than the calculation used today due to a lack of data where the process time has been measured. The lack of data also makes it hard to determine which method that is best suited for the problem. The fact that 11000 samples (corresponding to about 7.6 days of measurements) are needed to get stable predictions also makes the methods unsuitable for real time process control. The methods are also sensitive to the filtering of the data and there is no filtering that works best for all portions of data. This causes some the methods to show oscillating behaviour that are unreasonably large on some parts of the data, which makes predicting the time delay unreliable. Despite these negative aspects, we found promising signs that the methods can be used as a tool to monitor how the process time develops over time. This is something we initially did not plan to use the methods for, however it it would be interesting to continue developing them for this purpose in the future.

7 References

[1] Quazi A. ”An Overview on the Time Delay Estimate in Active and Passive Systems for Target Localization”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29 (June 1981), pp 527 - 533.

[2] Tan Y. ”Time-varying time-delay estimation for nonlinear systems using neural networks”. In: *International journal of applied mathematics and computer science* 14 (2004), pp 63-68.

[3] Elnaggar A, Dumont G.A, Elshafei A.L. ”Adaptive control with direct delay estimation”. In: *IFAC proceedings volumes* 26 (July 1993), pp 349-353.

[4] Björklund S. ”A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems”. In *Linköping Studies in Science and Technology*, Thesis No. 1061, (2003).

[5] ISO 302:2004 Pulps — Determination of Kappa number

[6] Gellerstedt G et.al. ”Pulp and Paper Chemistry and Technology”. KTH Royal Institute of Technology, (2006), chapter 9-10.

[7] Rojas R. ”Neural Networks - A Systematic Introduction”. Springer-Verlag, Berlin, (1996), chapter 7.

[8] Google. *Tensorflow*, URL: **https://www.tensorflow.org/** (visited 2020-02-05)

[9] François Chollet. *Keras deep learning API*, URL: **https://keras.io/**, (visited 2020-02-05)

[10] Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton. ”ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems* 25 (Jan. 2012).

A Appendix

A.1 Layers of CNN

Table A.1 – Detailed list of the layers used in the convolutional neural network

Layer	Kernel Size	Stride	Nodes
Convolutional + ReLU	3 x 3 x 96	3	N/A
Max pooling	2 x 2	2	N/A
Convolutional + ReLU	3 x 3 x 256	1	N/A
Max pooling	2 x 2	2	N/A
Convolutional + ReLU	2 x 2 x 256	1	N/A
Fully connected + ReLU	N/A	N/A	4608
Dropout (40%)	N/A	N/A	N/A
Fuly connected + ReLU	N/A	N/A	1000
Dropout (40%)	N/A	N/A	N/A
Fully connected + linear	N/A	N/A	1

A.2 Process time estimation in use today

The time delay estimation for the bleaching plant in use today is based on the volume of the bleaching towers in cubic metres, the concentration of pulp in the system and the production rate. The production rate is measured in how many tonnes of dried pulp that is produced each hour, after the pulp has been dried to a moisture level of 10%. Denoting the volume of the D1 and P1 stage as V_{D1} and V_{P2}, respectively, the concentration as C and the production rate as P, we can calculate the process time, τ, in minutes as

$$\tau = \tau_{D1} + \tau_{P1} = \frac{V_{D1}C \cdot 0.95 \cdot 60}{P \cdot 0.9} + \frac{V_{P1}C \cdot 0.95 \cdot 60}{P \cdot 0.9},$$

where τ_{D1} and τ_{P1} is the process time for the D1 and P1 stage, respectively. This calculation also assumes that we have a plug flow. We also have that $V_{D1} = 500$ m^3, $V_{P1} = 430$ m^3, and $C = 0.15$. This gives the following expression for the time delay.

$$\tau(P) = \frac{8835}{P}.$$