Universiteit Leiden

Computer Science

Estimating Tool Damage & Remaining Useful Life of a CNC milling cutter by applying Time-Frequency Analysis, Machine Learning and Evolutionary Optimization

Name: Nikolaos Bismpikos
Date: 31/01/2017

1st supervisor: Thomas Bäck
2nd supervisor: Erwin Bakker

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

List of Figures

Abstract

The constant evolution of machinery and the increased degree of automation along with advances in technological knowledge have given rise to predictive maintenance (PM), a maintenance scheme that can diagnose the current state of machinery or even predict its remaining life based on collected data. In the scope of this thesis, a PM framework is designed for the estimation of tool damage and remaining tool life in CNC milling machine cutters, based on collected time-series data measuring force, vibration and acoustic emissions after each cut. Time-domain, frequency-domain and time-frequency domain features are extracted from the time-series. For the latter case, Wavelet Transform is used since it is much more suitable than Fourier Transform and its variants for time-frequency analysis of non-stationary signals. Further feature selection and feature transformation techniques are applied with the aim of dimensionality reduction and ultimately a final subset with relevant and non-redundant features. In all cases, a necessary transformation for the problem at hand is to take the cumulative values of the features, therefore adding all previous values to a feature's value. The final feature set is then used as input to a proposed modification of Extreme Learning Machine (ELM), a single layer feedforward network that performs learning of the connecting weights without iterations. A modification of ELM, ES-RELM is proposed which uses regularization and modified Evolutionary Strategy (ES) in order to find a stable model that can predict tool wear sufficiently for different test sets. ES-RELM significantly improves the generalization capabilities of the found optimal regression model and clearly outperforms ES-ELM, a similar methodology without the use of regularization. The feature set that performs best is acquired after PCA performed at the whole feature set, since it projects the feature set into 4 independent features, thus resulting in non-redundant features. Last but not least, recommendations for future research regarding all main parts of the framework are proposed.

Chapter 1

Introduction

1.1 Thesis Topic

There is a steady growing pressure on companies, urged by the worldwide competition, to streamline operations involving product and product related manufacturing system design, product manufacturing and system maintenance [1].

The main task of this thesis is to provide predictive maintenance solutions, by employing data mining techniques on real-world data. After a feature data set is acquired from the raw data by means of preprocessing such as noise removal, checking the quality of the data, feature extraction, selection and transformation, data mining and pattern recognition techniques are applied with the general aim of classifying normal from abnormal machinery function or determining and predictin machinery damage. Special focus is given on data measurements in the form of signals, consequently in time-series analysis, since in order to apply data mining techniques to time-series, certain factors need to be taken into account, for instance what time-series representation, distance measure or feature extraction technique to use. In the scope of this thesis, feature extraction is performed on time-series by using Continuous or Discrete Wavelet Transform. The extracted features, after all the preprocessing stages are finished, form the final data set on which data mining and pattern recognition techniques are applied. Some major time series related tasks include query by content [2], anomaly detection [3], motif discovery [4], prediction [5], clustering [6], classification [7] and segmentation [8]. In the scope of this thesis the main focus is on regression, which is

in essence a more generalized case of classification. The goal is to build a model that can predict tool damage and consequently the remaining useful life for a CNC milling cutter. For this goal, the experiments also involve clustering methods as well as methods to handle outliers and noise.

1.2 Thesis Overview

The rest of the thesis is organized as follows: Chapter 2 focuses on Maintenance Theory and background information. Specifically, a brief history of maintenance is provided, along with the three major types of maintenance strategies as well as the kind of maintenance data that are monitored and recorded. Focus is given on Predictive Maintenance (PM) along with literature review on PM that is relevant to the current thesis.

Chapter 3 is concerned with Time-series analysis. Since quite often maintenance data are in the form of time-series, also known as signals, some basic tools for processing these signals are defined and examined. Specifically, focus is on ways of acquiring the frequency content of a time-series. Starting with Fourier Transform and its drawbacks, Wavelet Transform is then introduced, a family of transforms that is ideal for multiresolution analysis of non-stationary signals.

In Chapter 4 some of the preprocessing steps applied to the data such as normalization and smoothing are described, along with feature extraction methodologies for time-series data. These methods are used in the experiments described in Chapter 7.

In Chapter 5, some basic background on Evolutionary Optimization is given. A typical Evolutionary Strategy (ES) algorithm is outlined along with some proposed modifications and a meta-ES scheme used to tune the ES parameters.

In Chapter 6 the focus is on Maintenance Decision support methodologies which are employed for the experiments described in Chapter 7. Specifically, the focus is given on supervised learning through Artifical Neural Networks (ANN) and a special type of ANN known as Extreme Learning Machine (ELM) Moreover, an optimization scheme which is combined with an ELM in order to augment the performance of the ELM as a regressor, is defined and explained.

Chapter 7 reports the experiments done on the real world data and the respective results, by using the methods described in the previous chapters.

Last but not least, Chapter 8 is devoted to discussion of the results as well as relevant future research.

Chapter 2

Maintenance Theory

2.1 Introduction

The essence of maintenance is to ensure that the respective machinery is at satisfactory condition with regards to a certain operation [9]. Although defining a satisfactory condition depends on a variety of factors such as the type of operation, industry and application objectives, to name but a few, there are a number of defined criteria that are used to evaluate machinery condition. These criteria and the conditions that have to meet are as follows [10]:
1) Performance: the ability of the machine to perform its functions.
2) Downtime: operation of the machine must be within acceptable level of downtime.
3) Service life: before replacement of the machine is necessary it must provide a good return on investment.
4) Efficiency: the level of efficiency of the machine must be acceptable.
5) Safety: the machine must be safe to the personnel.
6) Environmental impact: the operation of the machine must be friendly to the environment and other equipment.
7) Cost: it is expected to have a maintenance cost with in an acceptable level.

Hence, taking into account these factors it is now possible to define the goal of maintenance more precisely [9]: The goal of maintenance is to ensure that machinery performance is satisfactory, considering the above factors. In order to get a more complete picture of maintenance, the rest of the chapter

includes maintenance history, in specific how maintenance evolved and how the three main types of maintenance strategies emerged. A definition and brief description of these three maintenance strategies, namely corrective, preventive and predictive maintenance is given. The last section of this chapter focuses extensively on predictive maintenance, which is the main topic of this thesis.

2.2 Maintenance History & Evolution

As industrialization was in process and machinery became more complicated and the degree of automation increased, more and more focus had to be given on maintenance. Thus, along with the evolution of the machinery the maintenance process has been evolving and becoming of continuously more importance. In general, the evolution of maintenance is categorized into 3 different generations [9]:
1) the first generation, between 1930s and 1940s.
2) the second generation, between 1950s and 1970s.
3) the third generation, from 1980s till date.

During the first generation, the degree of industrialization was low. The machinery used in factories was simple and basic, therefore repairing and restoring was performed very fast. Thus, maintenance was not an important issue and it was limited to corrective maintenance, one of the three generic types of maintenance strategies.

As industrialization evolved, machinery became more complicated and dependence on machines was increasing. Consequently, repair became a more difficult and complex task, requiring more time and skills. Machinery failure resulted in longer downtime which led to the need of preventing these failures and consequently the resulting downtime. Inevitably, more focus was given on maintenance schemes, which resulted in the concept of preventive maintenance, which employs periodic maintenance operations on the machinery in order to reduce or delay machinery failure and downtime.

During the third generation, production's dependence on machinery increased even more, apparently with an accompanying increase in complexity and degree of automation. Machine failure and downtime could be detrimental for the industry's operation hence maintenance became a significant task of high priority. At the same time, maintenance tools improved and technology and knowledge to predict machine failure had become available. This led

to the third type of maintenance strategies known as predictive maintenance, which relies on collecting data indicative of machinery health and condition and based on these data predicts if the machine is due to failure.

2.3 Maintenance Strategies

In literature it is possible to find three generic types of maintenance [11, 12]: corrective maintenance, preventive maintenance and predictive maintenance. As mentioned, these maintenance strategies emerged during the three respective maintenance generations defined above.

Corrective maintenance, consists in repair actions when equipment or machine fails. The equipment is in action until the moment that it fails. At that moment it will be repaired or replaced. The main disadvantages of this approach include fluctuant and unpredictable production, high levels of non-conforming products and scraps as well as high levels of maintenance interventions motivated by catastrophic failures [13].

Preventive maintenance, also known as planned maintenance, is characterized by periodic maintenance operations in order to avoid equipment failures or machinery breakdowns, determined through optimal preventive maintenance scheduling using a wide range of models describing the degrading process of equipment, cost structure, and admissible maintenance actions [14]. The main drawback of preventive maintenance lies in the fact that, contrary to the past, equipment and machinery have become so complex that a periodic maintenance scheme is very expensive. Hence, the need for more efficient maintenance schemes gradually became more and more crucial, which gave rise to the predictive maintenance scheme.

Predictive maintenance (PM), also known as Condition Based Maintenance (CBM), is the task of predicting when machinery failure is due and therefore when service is needed, based on data collected from the machinery. While preventive strategies are generally suitable for equipment that is not process-critical and will cause little or no damage if allowed to run to failure, an effective predictive maintenance system can significantly reduce unexpected failures as well as repair costs. Thus, an accurate prediction of a potential problem can provide better maintenance at an overall lower cost. Moreover, unnecessary maintenance tasks are avoided by performing maintenance only when it is deduced so from the collected data.

2.4 Predictive Maintenance (PM)

PM consists of three main steps: data acquisition, data processing and maintenance decision-making [15]. The first step, data acquisition, involves information collection, therefore acquiring relevant data with respect to machinery condition. The second step, data processing, involves processing the data. This includes operations such as checking quality and properties of the data(eg cleaning data, handle missing values) and feature extraction. Last but not least, based on the processed data, the last step is the decision-making process, which decides the machinery health and consequently if or what maintenance action should be performed.

PM can be divided into two main categories: diagnostics and prognostics. Prognostics deals with fault prediction while diagnostics with fault detection. Fault prediction recognizes a forthcoming fault and also provides an estimate on the probability and timing of the respective fault. Fault detection indicates whether something is wrong in the current state of the machinery. Prognostics can be seen as a more important task since it can predict future faults and hence result in decreased machinery downtime and increased reliability. However, diagnostics can also be important in case prediction fails. Apart from fault detection, diagnostics also deals with fault isolation, therefore locating the root of fault, as well as with fault identification, therefore determining the type of fault. Hence diagnostics is also an important task since it can provide insights on the root causes of fault as well as classify it into categories.

2.4.1 Predictive Maintenance Data

Data acquisition is a process of collecting and storing useful data (information) from targeted physical assets for the purpose of CBM [15]. Collected data can be divided into two main categories, condition data and event data. Condition data are measurements that reflect the machinery's condition, while event data provide information about what happened at a certain time point (eg. a failure and its probable causes) or information on what action was performed (eg repair and possibly a brief description of the repair process). Apparently, condition data are indispensable for PM, and a PM scheme based only on condition data can reduce downtime. However, event data are also important since they can be used as flags at different time points and potentially provide further insights.

Condition data can be further divided into three categories: 1)Value data 2) time series data and 3) multidimensional data. Value data are single values collected at specific time points. Time series data, also known as waveform data, are signals collected for a specific time period and a respective sampling rate. It should be noted that value data include not only the raw collected data but also single value data extracted from time-series data after a feature extraction procedure. Multidimensional data are collected data that span more than one dimension, such as images. In this thesis, the condition data used are time series data. Value data are not used, although the proposed framework can be trivially generalized to also include value data along with time series data. Multidimensional data are beyond the scope of this thesis.

2.4.2 Literature Review

The literature on machinery diagnostics and prognostics is huge and diverse primarily due to a wide variety of systems, components and parts. Hundreds of papers in this area, including theories and practical applications, appear every year in academic journals, conference proceedings and technical reports.

Since it is apparently not feasible to cover the whole literature, a brief literature review is given with research that is relevant to the scope of this thesis, namely involving CNC milling cutters, wavelet analysis, neural networks and evolutionary optimization.

Li et. al [16] combine fuzzy inference logic with neural networks to build a Fuzzy Neural Network (FNN) in order to detect and define tool damage of CNC milling cutter, and consequently the remaining tool life. The training of the FNN is performed by an extension of the back-propagation method, which extension includes the learning of the fuzzy rules. Chen et. al [17] use Genetic Algorithm (GA) and Evolutionary Stratey (ES) along with a neural network classifier. Feature selection and subsampling of the dataset is performed by a GA, while the ES is used in the next stage to optimize the construction (number of nodes) and the training (connection weights) of the neural network. Yan et. al [18] have developed a fault diagnostic methodology for diesel engine combustion system, based on neural networks and evolutionary optimization. An evolutionary algorithm is used to adjust the connection weights of the neural network. In a similar but more complete and sophisticated scheme, Huang et. al [19] propose an evolving wavelet network for power transformer condition monitoring. The wavelet network

is a neural network with a layer consisting of wavelet nodes which perform multiresolutional analysis of a time-series in the time-frequency domain. The evolving wavelet network proposed by the authors optimizes the parameters of the neural network through evolutionary optimization, including both parameters that affect the wavelet analysis and the connecting weights of the neural network.

Chapter 3

Time-Frequency Analysis

3.1 Introduction

This chapter is concerned with the topic of time-series analysis. In specific, the focus is on different transforms available in order to acquire the frequency content of a time-series, as well as on ways to use these transforms for feature extraction. First ,a description of Fourier Transform (FT), which is in essence the frequency domain representation of a signal, is given and it is explained why it is not suitable for real-world applications with non-stationary data. A variation of FT, namely Short-Time Fourier Transform (STFT) that can deal with non-stationary data is also described along with its drawbacks. Furthermore, another family of transforms, Wavelet Transforms, which are superior for analysis of non-stationary time-series is described and defined. The aforementioned transforms are then used as tools for extracting features from the raw signals.

3.2 Fourier Transform (FT)

Fourier Transform (FT) provides information about the frequency content of the signal on which the transform is applied,according to the follow formula

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} \, \mathrm{d}t \tag{3.1}$$

FT provides information as to what frequencies are contained in the signal as well as the intensity of these frequencies. In essence, FT decomposes

the signal into a sum of sinuses and cosines. However, there is no time information, therefore no information regarding at which time period these frequencies are contained, since FT assumes that the frequency content of the signal is the same for the whole duration of the signal. Hence, it becomes obvious that FT is suitable for stationary signals. If it is applied to non-stationary signals, therefore signals with varying frequency content over time, these signals are treated as stationary, hence the temporal variation of their frequency content is ignored. In essence, if we apply FT to a non-stationary signal we get the frequency content of the signal averaged over the duration of the signal, which is not really useful in the vast majority of real-world data, since in non-stationary time series analysis it is important to know the frequency content of the signal at various time intervals.

3.2.1 Short-Time Fourier Transform

A variation of the FT that can deal with non-stationary signals is based on dividing the non-stationary signal into short segments where the signal can be considered stationary and then apply the FT at each of these segments. This variation of FT is named Short Time Fourier Transform (STFT). In contrast to FT, STFT can provide information about the temporal variation of the signal's frequency content. In essence, the only difference between STFT and FT is that FT first divides the signals into non-overlapping windows. The only difference is therefore the use of a windowing function, as can be seen in the respective formula:

$$X(\tau, f) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i2\pi ft}\,\mathrm{d}t \tag{3.2}$$

The windowing function has a value equal to 1 for time points t that are within the window of center τ and zero for time points t outside the window. It is apparent from the above formula that temporal information is also taken into account. Apparently, if we do not want to lose any information regarding the varying temporal frequency content of the signal then window width must be such that each segment of the signal is stationary. However, although STFT can provide both frequency and time information contained in a time series, there is an unavoidable trade-off between those two, rooted in Heisenberg's uncertainty principle. Briefly, in quantum physics the uncertainty principle states that it is not possible to know at the same time the

exact momentum and the exact position of a particle. Mathematically,this is formulated as:

$$\Delta_p \Delta_x \geq c \tag{3.3}$$

Δ_p is the error in the calculation of momentum, Δ_x the error in the calculation of position and c is a constant (equal to Planck's constant h). Accordingly, in the field of signal analysis Heisenberg's uncertainty principle is stated as such: It is not possible to know the exact frequencies contained in a signal in specific exact time points. Mathematically, with Δ_t being the error in calculation of time and Δ_f the error in the calculation of frequency, this is formulated as:

$$\Delta_t \Delta_f \geq \frac{\pi}{4} \tag{3.4}$$

It is however possible to know what frequency bands are contained in specific time intervals. Thus, a resolution problem arises. Better time resolution corresponds to worse frequency resolution and vice versa. In the case of FT there is perfect frequency resolution but the time resolution is irrelevant since there is no temporal information at all. Using FT we can know the exact frequency value instead of a frequency band, which however does not violate Heisenberg's uncertainty principle since we do not have any time information at all. This becomes more clear if we see FT as a special case of STFT, with FT using a window of infinite width. In the case of a finite window, smaller window width corresponds to better temporal resolution and consequently worse frequency resolution. Since window width must be constant, the major drawback of STFT, especially in real-world applications is that it is not possible to define a window width that gives a satisfactory resolution trade-off for the whole duration of the signal. This is one of the reasons Wavelet Transform (WT) is superior, which will become clearer in the next section.

Briefly, using WT, time and frequency resolution are not the same for all frequency bands, but the trade-off between them changes for different frequency bands. In higher frequencies, WT provides better temporal resolution (consequently, worse frequency resolution) and in lower frequencies better frequency resolution (consequently, worse temporal resolution). In practice, this is really useful since usually high frequencies have short duration, therefore appearing as 'spikes', while lower frequencies have larger duration, usually being present for the whole duration of the signal.

3.3 Wavelet Transform

3.3.1 Wavelet Definition & properties

A wavelet is defined as a wavelike oscillation with an amplitude that begins at zero, increases, and then decreases back to zero. In other words, a wavelet is a function $\psi(\mathrm{t})$ which is wavelike near the start of the axes and zero everywhere else. For a function to be admissible as a wavelet, its mean must be zero and it should also be localized both in frequency and time domains. In detail, a wavelet function must meet the following three properties [20] :

$$\int_{-\infty}^{\infty} \psi(u)\,\mathrm{d}u = 0 \tag{3.5}$$

$$\int_{-\infty}^{\infty} \psi^2(u)\,\mathrm{d}u = 1 \tag{3.6}$$

$$0 < C_\psi < \infty, where \quad C_\psi = \int_0^{\infty} \frac{|\Psi(\mathrm{f})|^2}{\mathrm{f}}\,\mathrm{d}f \tag{3.7}$$

The first property states that the integral of $\psi()$ must be equal to zero, consequently the deviations of the function above zero must be equal to its deviations below zero. The second property states that the integral of $\psi()$ squared must be equal to 1, which means that the function cannot be zero at all time points but there must exist deviations from zero and they should be sufficiently small. The third property states that $\psi()$ must meet the admissibility condition, which means that the original signal can be acquired by using an inverse wavelet transform. Moreover, it should be mentioned that in the case of a complex wavelet there is a fourth property which states that the wavelet's FT must be real and equal to zero for negative frequencies [21].

3.3.2 Continuous Wavelet Transform (CWT)

While FT decomposes a signal into a sum of sines and cosines, CWT decomposes the signal into a sum of wavelet functions at different scales s and translation τ, according to the following mathematical formulation of the transform:

$$W(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} \mathrm{x}(t)\psi^*(\frac{t-\tau}{s})\,\mathrm{d}t \tag{3.8}$$

CWT is basically the representation of the signal's frequency content over time. As apparent in the mathematical formula, this is due to the convolution of the signal with different wavelets which are variations of the so-called mother wavelet. These wavelets are commonly referred to as daughter wavelets because they are copies of the mother wavelet in a different scale s and translation τ. Scaling of the mother wavelet enables decomposition of the signal in different frequency bands while translation enables decomposition at different time intervals. Hence, it now becomes clear why WT can provide different time-frequency resolution trade-offs, in contrast to STFT. An example of a mother wavelet for different scale and translation values is given in Figure 3.1. This wavelet is called Morlet wavelet and is a Gaussian-windowed complex sinusoid wavelet defined as in the following formula:

$$\psi_0(t) = \pi^{-\frac{1}{4}} e^{-i\omega_0 t} \tag{3.9}$$

The second order exponential attenuation provides good time resolution, while ω_0 is the central angular frequency of the wavelet and defines the time-frequency resolution trade-off. It can be proven that $\omega_0 = 6$ results in optimal time-frequency resolution [22]. In the following section it will be described in more detail how various mother wavelets can be ideal for different applications. Furthermore, the characteristics that affect the choice of mother wavelet are listed and analysed, along with the choice of set of scales.

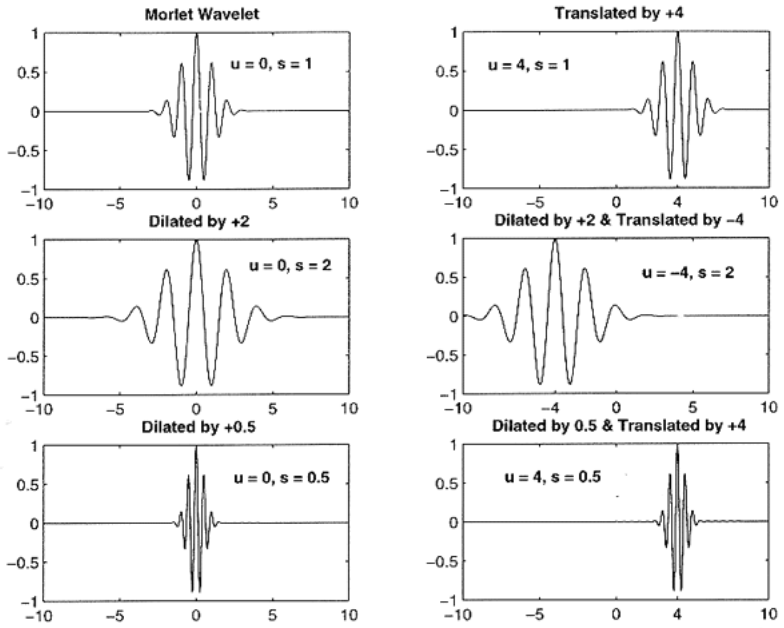


Figure 3.1: Morlet Wavelet for different scale s and translation u values (image taken from Kiymik et. al [22]
)

3.3.3 Choice of Mother Wavelet and scale

The choice of mother wavelet, along with the choice of scales are in general important factors in time-series analysis, since not only they can provide different time and frequency resolutions but also these choices should reflect the data to be processed and what kind of information to extract from these data. The following four factors are commonly taken into account as far as the choice of mother wavelet is concerned [23]:

1. Width: Width is defined as the e-folding time of the wavelet's amplitude, with e-folding time defined as the time interval in which an exponentially growing quantity increases by a factor of exp. Wavelet width in time and frequency domains define its resolution. For example, a narrow wavelet function provides good temporal analysis but bad frequency analysis. Conversely, a wide wavelet function provides frequency resolution but bad temporal resolution.

2. Shape: The shape of a wavelet function should reflect the characteristics of the data to be processed. For example, in time series that contain sudden leaps a boxcar-like function should be chosen, therefore a function which is everywhere zero except for an interval where it takes a constant value. However, in case we are just interested in the power spectrum of a time series the choice of wavelet function is not critical since all wavelet transforms decompose the signal into wavelets, hence the energy spectrum of the signal remains the same after the decomposition and is independent of the choice of wavelet function. However, there might still be differences depending on the choice of scales.

3. Orthogonal or non-orthogonal: Choice of an orthogonal base implies using DWT while a non-orthogonal wavelet function can be used with either DWT or CWT. In the case of an orthogonal wavelet, the width of the wavelet base in every scale proportionally defines the number of convolutions in this scale. Hence, the wavelet spectrum contains discrete blocks of wavelet power which is useful for signal processing as it gives a compact representation of the signal. In the case of non-orthogonal wavelets the wavelet spectrum is significantly correlated in neighbouring time points at larger scales (corresponding to lower frequencies) hence analysis is redundant at these points. Non-orthogonal wavelets are suitable for time-series analysis where smooth and continuous variations of the wavelet amplitude are expected.

4. Complex or real: There are real and complex wavelet functions. Their difference lies in the kind of information they provide. For example, a complex wavelet provides information about the amplitude as well as the phase of the oscillation, which is useful in identifying oscillatory behaviour, while a real wavelet function provides a single component and can be used to identify peaks or discontinuities in the amplitude of the signal.

The choice of a set of scales is also important and related to the choice of wavelet function. If an orthogonal wavelet has been chosen then the choice of scales is limited to a specific set of scales as defined by Farge et. al [23]. If non-orthogonal wavelets are chosen then an arbitrary set of scales can be used. In this case, a larger number of scales generally gives better frequency resolution but apparently also increases the computational cost since the WT

must be calculated for a larger set of scales. Scales are commonly written in fractional powers of two [23]:

$$s_j = s_0 2^{j\delta_j}, j = 1, 2...J \tag{3.10}$$

$$J = \delta_j^{-1} \log_2 (N \delta_t s_0) \tag{3.11}$$

Where s_0 is the smaller scale and J defines the larger scale. The choice of s_0 must be such that the respective Fourier period is approximately δ_t. Moreover, the choice of a sufficiently small δ_j depends on the width of the wavelet function in its frequency-domain. For example, for Morlet wavelet, δ_j around 0.5 is the largest value that gives sufficient sampling in the scale space, while in the case of other wavelets larger values can also be used. Generally, smaller δ_j results in better frequency resolution. It is possible to define a direct correspondence between scale and frequency and it is quite common in the visual representations of WT to replace the values in the scale axis with the corresponding frequency values, due to frequency being a more familiar measure. The following formula gives the relation between frequency and scale:

$$F_s = \frac{F_c}{s\Delta} \tag{3.12}$$

Where s is the scale, F_c is the central frequency of the wavelet and Δ is the sampling period. The reasoning behind this formula becomes more obvious if it is taken into account that for a mother wavelet, which therefore has not been scaled up or down, it is $s = 1$. Then, a periodical signal with frequency equal to the central frequency of the wavelet can capture the main oscillations of the wavelet. Accordingly, if the wavelet is scaled, this central frequency will be F_c/s and if the sampling period Δ is taken into account the formula of Equation 3.12 is derived.

3.3.4 Discrete Wavelet Transform (DWT)

Although calculating the CWT for a larger set of scales has the potential to provide more fine grained frequency resolution, at the expense of extra computational cost, there is still an upper threshold in the maximum resolution that can be achieved due to Heisenberg's uncertainty principle. DWT is an alternative WT that can also provide multiresolution analysis and similarly

to CWT can provide a good resolution trade-off. A major advantage of DWT compared to CWT is that it can be computed very efficiently using filters and downsampling. Specifically, at each level of the transform the signal is passed through a high-pass and a low-pass filter simultaneously. The filters are half-band filters, therefore divide a frequency band into two equal bands and are applied to the original signal through convolution. Apparently,for each filter's output half the frequencies of the original signal are removed, which means that half the samples can be discarded according to Nyquist's rule. The two filters $h(k)$ (low-pass) and $g(k)$ (high-pass) must be related to each other and form a quadrature mirror filter (QMF), defined as a filter whose magnitude response is the mirror image around $\frac{\pi}{2}$ of that of another filter:

$$g(k) = (-1)^k h(1-k) \tag{3.13}$$

Moreover, the mother wavelet function $\psi_{j,k}(t)$ and the respective scaling function $\phi_{j,k}(t)$ must form an orthonormal basis. This is mathematically formulated as:

$$\phi_{j+1,0}(t) = \sum_k h[k]\phi_{j,k} \tag{3.14}$$

$$\psi_{j+1,0}(t) = \sum_k g[k]\psi_{j,k} \tag{3.15}$$

In essence, the mother wavelet function gives the 'detail coefficients' of the transform (Equation 3.17) while the scaling function which is orthonormal with regard to the mother wavelet gives the 'approximation coefficients' (Equation 3.16).

$$A_{j+1,n} = \sum_k A_{j,k} h_j[k-2n] \tag{3.16}$$

$$D_{j+1,n} = \sum_k A_{j,k} g_j[k-2n] \tag{3.17}$$

It should be clear that changing the mother wavelet corresponds to changing the QMF. For example, if Haar wavelet is used, defined as $\psi = [-1, 1]$, then $g[k] = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$ and $h[k] = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$.

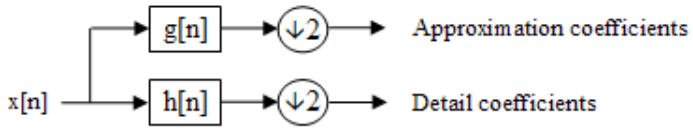


Figure 3.2: One level of DWT, resulting in approximation and detail coefficients of the original signal

In essence, the input signal is transformed by being convoluted with each of the bandpass filters and then each output is subsampled by 2 (Figure 3.2). However, calculating the convolutions first and then applying downsampling would be inefficient. The efficient calculations of DWT is due to the Lifting Scheme, according to which the signal is first divided and then convolution and accumulation operations are applied to the divided signal parts.

The above procedure can be further repeated to its outputs. This way, the approximation and detail coefficients can be further divided by applying the same halfband filters as well as downsampling. In signal processing this is referred to as a filter bank, defined as an array of band-pass filters that separates the input signal into non-overlapping frequency sub-bands of the original signal (Figure 3.3).

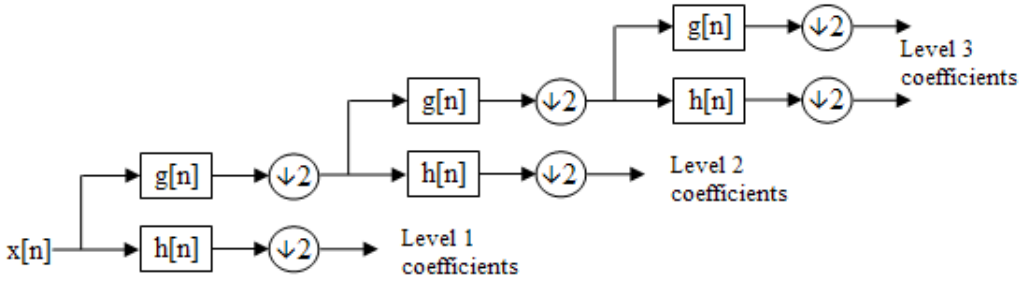


Figure 3.3: A filter bank for 3 levels of DWT

Due to downsampling, in each level of decomposition the number of samples is half of the number of samples of the previous level. Consequently, more samples are used for higher frequency bands. Similarly to CWT, this means that higher frequencies have better time resolution. It is noteworthy that the total number of DWT coefficients is equal to the length of the signal,

hence for a signal that is 128 samples long the resulting wavelet coefficients are also 128 in total (Table 3.1). On a related note, DWT assumes that the input signal length is a power of 2. In practice though, there exist various ways to deal with signals of other lengths as well, such as zero-padding or periodic padding.

Level	Frequency Band	Samples
1	$[f_{max}/2, f_{max}]$	64
2	$[f_{max}/4, f_{max}/2]$	32
3	$[f_{max}/8, f_{max}/4]$	16
	$[0, f_{max}/8]$	16

Table 3.1: 3 decomposition levels for a signal with 128 samples and frequency range 0 to f_{max}

3.3.5 CWT vs DWT

CWT and DWT both have their advantages and disadvantages, each being useful in different applications. A main advantage of DWT is its fast, efficient computation. In signal processing applications DWT's efficient computation allows for fast decomposition and recomposition of the signal. When only certain frequency bands of the signal are chosen to be recomposed by applying the inverse transform, DWT can act as zone-band filter. Moreover, DWT provides a very good energy compactification of the signal, therefore it is a suitable method for compressing the signals. As mentioned, the wavelet coefficients represent the frequency content of the signal over time, and their total number is equal to the signal's length.

On the contrary, for CWT the total number of coefficients is equal to the number of different scales multiplied by the signal's length. Thus, apart from the extra computational cost, lots of the information provided by CWT is redundant. On the other hand, CWT can potentially provide a more fine-grained resolution, which can be useful in some tasks such as anomaly detection. Moreover, CWT is time-invariant while DWT is not, which means that shifts in time-series can produce different results. However, there exist variations of DWT that are time-invariant. Last but not least, by using the CWT it is possible to get some measures that cannot be derived using DWT. These measures can be acquired when pairs of signals are compared

and they provide various types of information regarding first and second order correlation between the signals.

Chapter 4

Data Processing for Time-Series Data

4.1 Introduction

The representation and quality of data is first and foremost before running an analysis [24]. This chapter contains descriptions of the preprocessing techniques that are employed for the task of this thesis. Data preprocessing includes cleaning, normalization, transformation, feature extraction and selection, therefore data preprocessing refers to all the operations applied to the data until the final training data set is acquired. For conveniency, in the scope of this thesis data processing is distinguished between preprocessing (cleaning, normalizing and in general transforming data) and feature extraction and selection of data. Specifically, the preprocessing part is concerned with normalizing and standardizing data, as well as handling outliers and smoothing.

4.2 Data Processing for Time-Series Data

4.2.1 Data Preprocessing

Normalization & Standardization

One of the goals of applying normalization to a feature set is to scale the features so that they lie in a common range. This way features are directly comparable, and furthermore the performance of machine learning techniques can

be improved. For example, in several types of neural networks it is required that the input data sets are normalized. If input data are not normalized, it is quite possible that the effects of certain features are predominant, while other features are practically not taken into account.

Standardization is quite similar to normalization, since it also performs a rescaling of the data. However, instead of rescaling the data so that all of them are in the same range, standardization rescales that data so that they have zero mean and unit variance. One normalization method (min-max normalization) and one standardization method (z-score) are defined next:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

$$z_x = \frac{x - \mu}{\sigma} \tag{4.2}$$

In the above equations, x is the value of one feature observation, x_{min} and x_{max} are the minimum and maximum values of the feature vector respectively, μ and σ are the mean and standard deviation of the feature vector, while x_n and z_x are the new respective normalized and standardized values of the observation. It should be noted that for the case of min-max normalization, data are scaled to lie in $[0, 1]$, although this range can be then modified to be $[-b, b]$ through a simple linear scaling.

Handling Outliers & Smoothing

The median filter is a nonlinear digital filtering technique, often used in signal processing to remove outliers and noise. The main idea of the median filter is to replace each time-series point with the median of its neighborhood. The neighborhood is often called the window or the order of the median filter. For a window with an odd number of points, the median is trivially calculated as the middle value after all values have been sorted. If the window contains an even number of points the median can be calculated in various ways, usually by taking the average of the two middle values after sorting them.

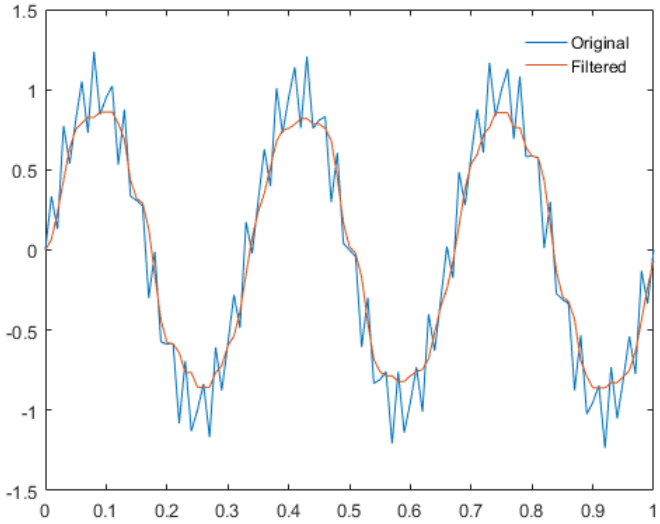


Figure 4.1: Plot a peaky time series and its smooth version after median filter is applied

If the appearance of outliers is rare, a narrow window is enough to remove them. However if outliers appear frequently and the signal is too noisy then a larger window might be necessary. In general, the larger the window the more intense the smoothing effect will be and the larger changes will be in the original signal. Too small window size values might not remove or reduce the outliers significantly, while too large values might distort the data and consequently the information they might provide.

4.2.2 Feature Extraction from Time-Series

The necessity of feature extraction

Starting from an initial data set, the goal of feature extraction is to derive values from the initial data, which derived values form the so-called feature set. The mathematical definition of a data set is a $r\,x\,c$ matrix, where rows r represent samples and columns c represent features. Hence, each column corresponds to a feature while each row corresponds to a sample, therefore a vector with values for each feature. Since the initial data sets are often large and contain redundant information, ideally feature extraction should

result in informative and non-redundant data. Feature extraction is related to dimensionality reduction, therefore the extracted feature set is usually smaller than the original. Since in many real world problems the initial data can be of huge size and possibly redundant as well, feature extraction is very often a necessary process which can transform the initial data into data more suitable for machine learning methods to be applied.

In general, feature extraction is almost always necessary for time-series data. For example, assuming a sampling rate of 100Hz, each second of the signal is represented by 100 data measurements, hence the number of samples required to represent a signal that lasts one hour is too large, consequently applying machine learning methods on such a data set would be very expensive computationally. Moreover, not all the information contained in the signal is meaningful or important since signals may contain noise or information that is irrelevant in the context of the respective application. Thus, when machine learning algorithms have to be applied on time-series data feature extraction is almost always necessary, since although it is possible to apply machine learning algorithms to raw signals the computational cost is high and the results are in general poorer. In the next section, some feature extraction methods from recent literature are reviewed. Furthermore, possible variations of these methods are proposed, in order to provide a variety of feature extraction methods and choose the most effective.

Feature extraction from time-series

There are some basic operations common to most feature extraction methods on time series. First of all, signal segmentation is usualy an essential step for lengthy signals. Signals can be segmented either using a standard window length or other methods such as the Bayesian approach [25], where the signal is segmented according to detected changes of its mean value and variation. Then, from a machine learning point of view each window of the signal represent a sample of the data set and the features derived from each window represent the features for this sample. Features can be derived from the time-domain of the signal or from the frequency-domain or both.

Prochazka et al. [26] use DWT to extract features from Electroencephalogram (EEG) signals and then classify EEG signal segments. First, they divide the signal into segments by using the Bayesian approach. Then features are extracted from each segment by applying multivelel wavelet decomposition. For each segment of length 2^s, a filter bank is used with DWT being applied

for s levels. As mentioned already in the respective chapter different techniques such as zero-padding can be applied in case the input signal length is not a power of 2. The resulting DWT coefficients are then used as features that are fed to an self-organizing map (SOM) for classification. It should be noted that it is also possible to compress the signal features in order to reduce the number of patterns that are fed to the SOM. The number of classes is known beforehand and is equal to four, so the SOM in essence clusters the signal segments into four classes. This methodology is compared with a similar methodology with the only difference being that DFT is used instead of DWT. DWT outperforms DFT for all experiments in terms of classification accuracy and furthermore it results in more compact clusters.

Phinyomark et al. [27] also use DWT in order to extract features from both the time-domain and the frequency domain of an electromyography (EMG) signal and find the most optimal feature set in terms of class separation. They use a 4 level filter bank to get the wavelet coefficients of the signal at different decomposition levels and furthermore they recompose specific frequency bands of the signal. Then they extract frequency-domain features by using the wavelet coefficients at different level and time-domain features by the reconstructed signals that contain specific frequency bands. Instead of using the wavelet coefficients as features, the mean absolute value (MAV) and root mean square (RMS) measures are used. These are also the extracted features from the time-domain recomposed signals. Then each of these feature sets are evaluated in terms of class separability for 6 defined classes, by using the scatter graph and the RES index, a statistical measure. It is found that the time-domain features recomposed from the detail wavelet coefficients of level 1 and level 2 provide the best class separability. This methodology is hence proved useful in defining the most informative features, which is in essence the primal goal of feature extraction.

Kilby et al. [28] also analyse EMG signals although they use a slightly different approach. They use CWT first to analyze the frequency content of the signal over time,for different scales.Then, by visual inspection of the scalogram, the most dominant frequency components of the signal are selected. For each of the selected scales, a signal which contains only these specific frequency band is reconstructed. For each extracted signal three features are derived: Mean Frequency (MNF), Median Frequency (MDF) and RMS. MNF and MDF are frequency-domain features of the mean frequency and median frequency of the signals power spectrum respectively, acquired by taking the DFT of the signal, while RMS is a time-domain feature. It should

be noted that despite, as mentioned, FT is not suitable for non-stationary signals, this is a different case because the extracted signals contain specific frequency bands of the original signal, hence the FT can provide some informative features. For 5 different scales and 3 features at each scale, the feature set consists of 15 features, which are fed to a multilayer artificial neural network (ANN), which is trained and tested for various input signals. For 30 different input signals, the lowest classification error is 3.33%, for an ANN of 6 neurons at the hidden layer.

The above methods provide different alternatives for feature extraction. Apparently some of these methods share a lot in common, such as signal segmentation, decomposition of the signal's frequency content via DWT or CWT and extraction of time-domain or frequency-domain features. It should be noted that more measures than described in the above methods (MNF,MDF,RMS,MAV) can be derived, such as zero-crossing rate (ZCR), energy of the signal in time-domain, or average power of the signal or isolated frequency bands of the signal in the frequency domain.

Apparently, finding the most optimal methodology for feature extraction is application-specific, as it depends on the data and the specific task of the analysis. Thus, alternative methods must be tried in order to find an optimal one for the respective problem and furthermore these methods need to be optimized as they depend on some parameters, such as window length, choice of scales (for CWT) or levels of decomposition (for DWT), choice of wavelet function and choice of measures to derive. One way to optimize the feature extraction process would be manually, by applying different methodologies and different parametric setups for each respective methodology. However, this is generally a naive approach since it requires manually running many experiments. A more sophisticated approach would be to view the feature extraction process as a formal optimization problem. The evaluation criteria for an optimal feature extraction process is therefore the minimization or maximization of an objective function. From this point on, two different approaches can be tried. Evaluation can be performed either based on a statistical measure for example the class separability of each feature set, or based on the classification accuracy for each feature set used. The drawback of the latter approach, commonly known as wrapper approach, is that evaluation depends not only on the feature extraction but also on the classification method used. Hence, from an engineering point of view it would be more efficient and meaningful to first optimize the feature extraction process based on a statistical measure (such as class seperability, cluster compactness or

correlation with targets) and then possibly optimize the machine learning task (such as classification, clustering, regression and more) separately.

4.2.3 Dimensionality Reduction, Feature Selection & Transformation

Apart from feature extraction, the process of feature selection is also necessary. While feature extraction involves generating new features out of an existing data set, the goal of feature selection is to select an optimal subset of features from an existing feature set. Ideally, the goal is to select highly discriminative features, while discarding redundant features, therefore also resulting in dimensionality reduction. In this section, some of the most commonly used dimensionality reduction techniques are briefly reviewed, such as Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA). Then, a recent framework for dimensionality reduction which can result in highly discriminative low-dimensional space at a reasonable computational time, Semi-Random Projection (SRP), is described along with a proposed variation that makes it suitable for regression problems.

Furthermore, another approach for feature selection suitable for regression problems, Fast Correlation-Based Filter (FCBF) is also described. The main reasoning behind FCBF is that for a regression problem, a highly discriminative feature is one that shows high correlation with the respective targets to be predicted. At the same time, a feature should also satisfy some uniqueness criteria, therefore even if a feature is highly correlated with the targets it might still be redundant to other features and should hence be discarded.

Fast Correlation-Based Filter (FCBF)

As mentioned the goal of feature selection and also feature engineering in general is to yield a final data set which contains high-quality features with regard to the machine learning task at hand. The quality of a feature set can be examined individually, therefore with respect to how informative this feature is regarding the targets (class labels for classification problems or numeric values for regression problems). In other words, this can seen as the relevancy of a feature.

However, evaluating and choosing features based only on their relevancy to the targets has some significant limitations, since a feature set might

contain highly relevant features which are redudant. Apart from irrelevant features, redundant features have also been shown to affect the speed and accuracy of learning algorithms [29]. Hence, for a feature to be considered good, not only does it have to be relevant but also not redundant to any of the other relevant features chosen in the final subset.

Feature selection algorithms fall into two broad categories, the filter model or the wrapper model [30]. The filter model relies on general characteristics of the training data to select some features without involving any learning (classification or regression) algorithm. The wrapper model requires one predetermined learning algorithm in feature selection and uses its performance to evaluate and determine which features are selected. Thus, for each new subset of features, the classifier or regressor has to be trained. The wrapper method utilizes more information and usually outperforms the filter method since feature subsets are evaluated according to the classification or regression performance. However, the wrapper approach can be prohibitive when the number of features is large enough, since the number of possible feature subsets to evaluate grows exponentially.

Yu et. al [31] propose a filter-based feature selection methodology that can detect feature subsets with features that are both relevant and non-redundant. As far as quantificiation of relevancy is concerned, two different correlation measures are proposed: one is the linear correlation measures, such as Pearson's correlation, the other is based on information gain.

For the purpose of finding feature subsets that contain relevant but not redundant features, the concept of predominant correlation is defined by the authors as follows [31]: The correlation $SU_{i,c}$ between a feature F_i and a class C (or in general, the targets) is predominant if 1) the correlation measure exceeds a predefined threshold δ, therefore $SU_{i,c} \geq \delta$ and 2) there exists no other feature F_j that has a higher correlation with F_i than $SU_{i,c}$, therefore $\not\exists F_j$ such that $SU_{i,j} \geq SU_{i,c}$. Consequently, a feature is defined as predominant to the class (or targets) , if and only if its correlation to the class (or targets) is predominant or can become predominant after removing

its redundant peers.

```
Input: a data set X of d features S(F_1, F_2, ..., F_d), class labels C
       (classification) or target values T (regression), threshold δ
Output: optimal feature subset S_best
for i = 1 to d do
    calculate Corr_{i,T};
    if Corr_{i,T} ≥ δ then
        append F_i to S_list;
    end
end
Order S_list in descending order;
F_p ← getFirstFeature(S_list);
repeat
    if F_q ≠ ∅ then
        repeat
            F'_q ← F_q;
            if SU_{p,q} ≥ SU_{q,T} then
                remove F_q from S_list;
                F_q ← getNextFeature(S_list, F'_q);
            end
            else
                F_q ← getNextFeature(S_list, F_q);
            end
        until F_q = ∅;
    end
    F_p ← getNextFeature(S_list, F_p);
until F_p = ∅;
S_best ← S_list;
```

Algorithm 4.1: Outline of the FCBF algorithm (pseudocode)

Dimensionality Reduction through Feature Transformation

Given that the data set is denoted by a matrix $\mathbf{X}$ where rows correspond to observations or samples and columns to features, dimensionality reduction can be mathematically formulated as a mapping from the original space to a low-dimensional space according to the following formula:

$$\mathbf{H} = \mathbf{W^T X} \tag{4.3}$$

Hence, $\mathbf{H}$ is the new low-dimensional representation of the data set, $\mathbf{X}$ is the original data set and $\mathbf{W}$ is the linear transformation matrix. In essence, a linear dimensionality reduction method is characterized by the way $\mathbf{W}$ is computed. Two of the most common feature transformation methods are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). The goal for both PCA and LDA is to find an optimal $\mathbf{W}$ for the following optimization problem:

$$\mathbf{W}^* = argmax \quad Tr(\frac{\mathbf{W^T L W}}{\mathbf{W^T B}}) \tag{4.4}$$

Depending on the quantities $\mathbf{L}$ and $\mathbf{B}$ different dimensionality reduction schemes can be implemented.

PCA

PCA is an unsupervised dimensionality reduction method that projects data into dimensions with maximum variance. For PCA, matrices $\mathbf{L}$ and $\mathbf{B}$ of Equation 4.3 are determined as follows:

$$\mathbf{L} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T \tag{4.5}$$

$$\mathbf{B} = \mathbf{I}_d \tag{4.6}$$

LDA

LDA is a supervised dimensionality reduction that projects data into dimensions that maximize discrimination between the various class labels. For LDA, matrices $\mathbf{L}$ and $\mathbf{B}$ of Equation 4.3 are determined as follows:

$$\mathbf{L} = \sum_{c=1}^{N} n_c(\bar{x}_c - \bar{x})(\bar{x}_c - \bar{x})^T \tag{4.7}$$

$$\mathbf{B} = \sum_{i=1}^{N} (x_i - \bar{x}_{c_i})(\bar{x}_i - \bar{x}_{c_i})^T \tag{4.8}$$

Semi-Random Projection (SRP)

Although PCA and LDA are the most common dimensionality reduction methods they have their limitations. The main drawback is that both these methods are computationally expensive since determining the solution to Equation 4.3 has a cost of $O(d^3)$. Moreover, PCA maps the data in such ways that variance is maximized, but this transformation does not guarantee a better classification or regression performance. As far as LDA is concerned, although it does project features to dimensions that maximize discrimination between classes, it requires class labels so it is useful only for classification problems and not for regression ones.

Zhao et. al [32] have proposed Semi-Random Projection (SRP), a dimensionality reduction method that employs LDA in combination with random feature subsampling to determine the transformation matrix $\mathbf{W}$, thus managing to get the best of both worlds, a computationally fast dimensionality reduction that also produces a space with discriminative power. After describing the proposed SRP, a variation of it will be presented which makes it suitable for regression problems.

SRP is basically using the idea of Random Projection (RP) but in a smarter way that allows learning from data. In RP, $\mathbf{W}$ is determined completely randomly, according to the following formula:

$$w_{i,j} = \sqrt{c} \begin{cases} 1 & \text{with prob } p = \frac{1}{2c} \\ 0 & \text{with prob } p = 1 - \frac{1}{c} \\ -1 & \text{with prob } p = \frac{1}{2c} \end{cases}. \tag{4.9}$$

Where c is set to $\sqrt{d}$. The fact that some of the weights of the transformation matrix are set to zero corresponds to a random subsampling of the original data set. SRP employs this idea, but in contrast to RP the non-zero weights are not assigned randomly but depend on the data. Specifically, d_s features are chosen each time from the original set of d features, and LDA is performed on the subsampled feature set, projecting it into the first most

discriminative dimension. This procedure is repeated r times until $r\hat{w}_i$ transformation vectors are produced, which means the new space will contain r features. Since d_s is typically set to a value around $\sqrt{d}$, performing LDA is significanly faster than performing it on the original set ($O(\sqrt{d^3})$ instead of $O(d^3)$). Specifically, in the ith iteration, the original data set is reduced to a submatrix $\hat{\mathbf{X}}_i$, which $\hat{\mathbf{X}}_i$ is then projected to a one-dimensional space according to the following formula:

$$\mathbf{h}_i = \hat{\mathbf{w}}_\mathbf{i}^\mathbf{T} \hat{\mathbf{X}}_i \tag{4.10}$$

The transformation vectors are determined from the data in accordance to the LDA equations 4.7 and 4.8 , the only difference being that they are performed on the subsampled reduced dataset $\hat{\mathbf{X}}_i$. Specifically, the LDA equations for the reduced data set become:

$$\mathbf{L} = \sum_{c=1}^{N} n_c(\bar{x}_c - \bar{x})(\bar{x}_c - \bar{x})^T \tag{4.11}$$

$$\mathbf{B} = \sum_{i=1}^{N} (x_i - \bar{x}_{c_i})(\bar{x}_i - \bar{x}_{c_i})^T \tag{4.12}$$

The term n is a regularization term. Moreover, since instead of a transformation matrix W a transformation vectow w is now computed at each iteration, equation 4.3 now can be formulated as an eigenvalue problem:

$$\mathbf{L}\phi = \lambda \mathbf{B}\phi \tag{4.13}$$

In the above equation, ϕ is the eigenvector and λ its corresponding eigenvalue. Hence, the optimal transformation vector $\hat{w}_i$ can be determined from the optimal eigencvector $\phi_\mathbf{i}$ corresponding to the largest eigenvalue λ_1:

$$\hat{w}_i = \sqrt{\lambda_1}\phi_\mathbf{i} \tag{4.14}$$

Input: data matrix of training samples X, labels y, number of features in subset ds, number of dimensions/features in new computed data set
Output: reduced dimensionality matrix H
for *i =1 to r* **do**
    randomly select d_s features from the original dataset
    perform LDA on the reduced training subset and project subset to first most discriminative dimension
    project respective test subset accordingly
end

Algorithm 4.2: Outline of the SRP algorithm (pseudocode)

SRP for regression problems (SRP-PCA)

As stated, the proposed SRP is suitable for classification problems since it employs LDA, which uses class labels. If SRP is to be used in a regression problem, as in the current thesis, it has to be modified. One apparent modification would be to convert the known targets of the training set into class labels. After this conversion, SRP takes place exactly like described in the previous section. However, issues arise with this approach regarding the number of classes to divide the samples and corresponding targets into, as well as what criteria to employ in order to assign samples to each class.

An alternative approach would be to just replace the LDA with PCA in the SRP algorithm. The difference is in the way the subset is projected to a one-dimensional space each time: In the case of original SRP the projection is such that it maximizes discrimination between classes, in a supervised learning way, while in the case of the proposed SRP-PCA variation the projection is such that the variance is maximized. The computational complexity of the algorithm remains the same in both cases ($O(\sqrt{d^3})$).

Input: data matrix of training samples X, labels y, number of features in subset d_s, number of dimensions/features in new computed data set r
Output: reduced dimensionality matrix H
for *i =1 to r* **do**
    randomly select d_s features from the original dataset
    perform PCA on the reduced training subset and project subset to dimension that maximizes variance
    project respective test subset accordingly
end

Algorithm 4.3: Outline of the SRP-PCA algorithm (pseudocode)

Chapter 5

Evolutionary Optimization

5.1 Introduction

Evolutionary optimization refers to a family of meta-heuristic algorithms, commonly called Evolutionary Algorithms (EA) that share in common the adoption of Darwinian biological concepts. As metaheuristic algorithms, they are designed to generate solutions that are sufficiently good for an optimization problem, for which usually none or limited information exists, while computational capacity is also limited [33]. In general EAs, as metaheruistic algorithms, traverse the search space intelligently instead of exhaustively. This is typically done by generating a population of candidate solutions. This population is evolved according to biological evolution concepts such as mutation, recombination and natural selection. One of the most signficant advantages of EAs is that few or no assumptions about the optimization problem are made, which makes them suitable for a variety of problems where the search space is multi-dimensional and no information is available [34].

In the scope of this thesis, the EA used is Evolutionary Strategies (ES). Like most EAs, ES employs stochastic optimization, meaning that traversal of the search space depends on probabilistic generation of random variables. Moreover, self-adaptation is used for parameter learning, therefore for instance the parameters that affect mutation are learned dynamically as the population evolves in each iteration. In the rest of this chapter, first the standard ES algorithm is outlined and then several modifications are proposed which are applied with the aim of augmenting it. The next section assumes some basic background on ES and EAs in general, hence the processes of the

standard ES version are described briefly. For a more extensive insight into the theory of ES and EAs the reader can refer to [35].

5.2 Evolutionary Strategy (ES)

The representation of a candidate solution, corresponding to an indivual of the population, depends on the problem-specific search space, for instance on the dimensionality of the objective function to be optimized as well as its function domain. In the scope of this thesis, the search space is real-valued and defined for a specified continuous range. According to the basic outline (Figure 5.1), the population is first initialized and then recombination, mutation and selection are applied in a loop, with each iteration of the loop corresponding to each generation of the population of candidate solutions.

5.2.1 Standard ES

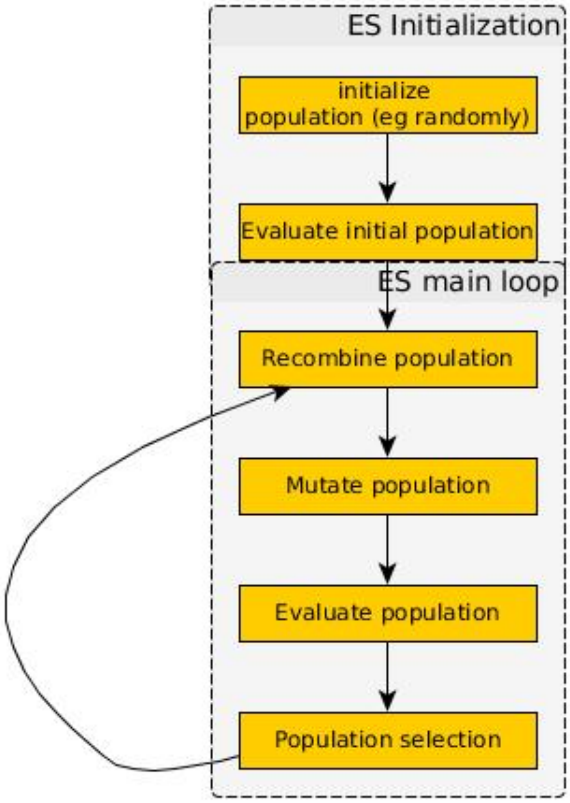


Figure 5.1: Outline of a typical ES algorithm

Recombination

During recombination individuals share information by creating new individuals (hereby referred to as offsprings) through exchange of their coordinates. Four different types of recombination were tried, which can be categorized into Intermediary (global or in groups) and Discrete (global or in groups) recombination.

Mutation

During the mutation process individuals (commonly referred to as parents) produce new individuals (commonly referred to as mutants) whose positions

depend on a normal distribution which is centered around the parent individual and has its variance defined by the step-size of the parent. In the current approach, individuals are treated as agents, while each agent uses different individual step sizes for every dimension of the search space. These step sizes also undergo mutation, according to global and local learning rates , using the values suggested by Schwefel for all agents. This approach can improve the adaptation of the individuals in the search space as there is the potential for diverse subpopulations that can adapt to the search space in different ways. Typically, in standard ES usually normal distribution is used, but other distributions such as uniform and log-normal distribution can be tried.

Selection

Initially, two different selection schemes are implemented. The first one is the greedy (μ + λ)-selection that selects the μ best individuals out of the group of μ parents plus the λ mutants and offsprings, resulting from mutation and crossover respectively. The second one, (μ,λ)-selection keeps the μ best individuals out of the group of λ mutants and offsprings. In the final implementation a combination of these schemes is used, with the details described in the next section.

5.2.2 ES modifications

These modifications regard: 1) a mechanism to deal with individuals whose position are outside the limits of the search space 2) an more sophisticated selection scheme that combines (μ + λ) and (μ,λ) selection 3) a more sophisticated mutation scheme that imitates the properties of Covariance Matrix Adaptation while maintaining a linear time complexity.

Reposition of solutions that violate search space boundaries

Since the coordinates of the population might violate the limits of the search space, a mechanism to deal with out of range positions should be implemented. A mechanism inspired from Particle Swarm Optimization (PSO), a swarm based optimizer where the position of an individual depends on the individuals best known position and the global best known position. In the proposed implementation one way to deal with individuals that are out of

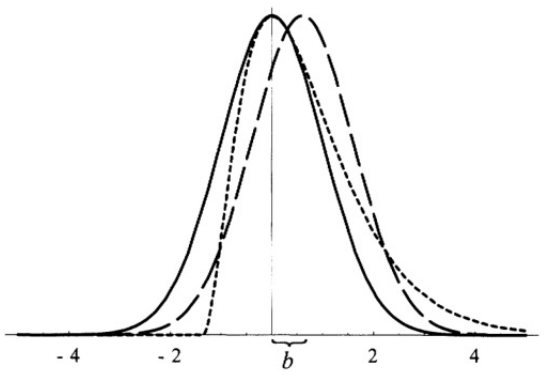
range is to distribute them around the current best of the population. This simplified PSO mechanism is applied only to a certain percentage of the out of range individuals according to a predefined probability. The other mechanism that deals with the rest of the out of range individuals is to reposition their violating coordinates within a normal distribution around respective violated limits of the search space. The motivation behind the PSO mechanism is to reduce possible function evaluations lost due to out of range positions as well as to potentially make advantage of such cases by integrating a simplified PSO search behaviour which in practice generates more candidate solutions near the best individuals of the population.

Mixed Selection Scheme

The implemented mixed selection scheme was inspired by Kramer [36] and was originally used for constraint-based optimization problems in order to separate the population in two different groups, one according to the minimum fitness found and the other according to the minimum number of constraint violations. This is achieved by employing a modified selection scheme: m_0 individuals are selected from the m parents according to minimum number of constraint violations and $m-m_0$ individuals from the λ new individuals according to minimum fitness.

Directed Mutation

Covariance Matrix Adaptation (CMA) is commonly used in ES algorithms as a means to rotate the coordination system. A covariance matrix is generated from the respective individ ual uncorrelated step sizes by multiplying with $n * (n-1)/2$ rotation matrices. This covariance matrix is then used to define the positions of the offsprings according to a multivariate normal distribution. Apparently, the advantage of this method is the increased directionality it provides, therefore its easier to escape premature convergence and global search is potentially more efficient. However, the disadvantage of this method is the increase in complexity of the mutation process to quadratic instead of linear. The performance is notably slower, especially if a different covariance matrix is estimated for each individual agent. Hence, it was our choice to experiment with other modifications in order to increase potential directionality and global search behavior, with the complexity remaining linear.



Comparison of standard mutation (solid), directed mutation (dotted), and biased mutation (dashed) with bias b.

Figure 5.2: Mutation Operators Comparison

The idea of Directed Mutation Operator (DMO) and Biased Mutation Operator (BMO) techniques was inspired from Kramer et. al [36]. Both these modifications alter the positions of the generated individuals (mutants). DMO achieves this explicitly by altering the skewness of the data, therefore adds a move to the positions of the mutants, while BMO moves the mean of the distribution to be used for the generation of mutants (Figure 5.2). In the current implementation, a small move is added to the positions of the mutants, according to a DMO operator that is learned and controlled similarly to the step size. Furthermore, the size of this movement is bound by the step size. DMO can be divided in two cases, according to how the directed movement is calculated. In the first case, the directed movement is generated randomly. In the second case, the directed movement is learned, based on the previous generated movements. In the current implementation, at first the directed movements are all generated randomly and then a percentage of them is learned while the rest are still generated randomly at each generation.

5.2.3 ES parameter tuning (meta-ES)

As far as parameter tuning is concerned, it is a common practice to perform several experiments with some predefined values for the parameters to be optimized. Since this is a rather naive method, we decided to use meta-optimization, therefore we used an ES optimizer to tune the ES parameters.

The advantages of this choice are several, for the same reasons a metaheuristic optimization algorithm such as ES is superior over exhaustive search. The advantages become much more obvious as the number of parameters to be optimized grows. The metaES enables efficient tuning of a large number of parameters and consequently enables us to parametrize more processes in the current implementation instead of using parameters with fixed values.

The meta-ES runs exactly the same algorithm as the ES (Figure 5.1) , the only difference being in the evaluation function that is used. In the case of meta-ES, evaluation of an individual corresponds to evaluating the fitness of a parametetric setup, which is done by running the nested ES for a given objective function and a given number of runs and evaluation budget. Thus, a single evaluation of the meta-optimizer corresponds to the whole evaluation budget of the nested ES, multiplied possibly by the number of times the nested ES is run.

Input: population matrix $\mathbf{X}$, objective function $func$, budget b, number of runs r
Output: fitness vector $\mathbf{f}$
decode population values into actual parametric values
for *each individual* $\mathbf{x_i}$ *of the population* **do**
  run ES with parameters $\mathbf{x_i}$ for the given function ,budget b and number of runs r
  f_i = mean f_{opt} over all r runs
end

Algorithm 5.1: The evaluation stage of the meta-ES

The population of the meta-ES represents parametric setups, with a parameter representing a dimension of the search space and the parametric ranges representing boundaries of the search space.

An encoding of the parameters takes place using min-max normalization (Equation 5.1) so that they all lie in the same range, which in here is chosen to be $[0, 1]$. These values are then decoded when needed for the evaluation part (Equation 5.2). The main reason that parameter normalization is necessary is due to the presence of discrete parameters (such as selection scheme, or recombination type) along with continuous ones.

$$f = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.1}$$

$$x = x_{min} + f \quad (x_{max} - x_{min}) \tag{5.2}$$

The meta-ES is capable of finding various, diverse optimal parametric setups for various objective functions. Hence, for the current impementation that includes that aforementioned modificiations and a large number of parameters, and depending on the optimization problem at hand, the meta-ES is able to find different parametric setups that result in good performance. Thus, the meta-ES has the potential to be a useful tool that can perform efficient application-specific parameter tuning of the ES.

Chapter 6

Maintenance Decision Support

6.1 Introduction

This section deals with the maintenance decision support system, designed to diagnose tool life of CNC milling machine cutter based on a feature set exracted from collected time-series data. The aim is to use Extreme Learning Machine (ELM), a specific type of neural network, in combination with ES, in order to find a stable model able to predict tool wear accurately.

6.2 Supervised Learning

6.2.1 Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) are types of network inspired from biological neural networks, where nodes correspond to neurons and edges between the nodes correspond to synaptic weights. ANNs can show great flexibility, depending on their structure and the activation functions used, which makes them suitable for a variety of applications. ANNs are shown to have the universal approximation property, meaning that they can approximate any function given enough training examples. ANNs are characterized by a multi-layer structure, typically having an input layer, one ore more hidden layer and an output layer (Figure 6.1). Each layer contains nodes, and each nodes applies an activation function to the summation of all its inputs.

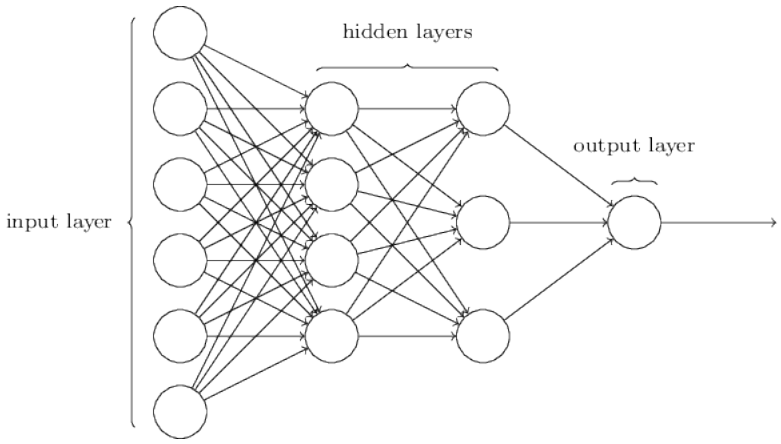


Figure 6.1: The generic structure of an ANN

6.2.2 Extreme Learning Machine (ELM)

ELM is in essence a standard feedforward NN with a single hidden layer that can be trained very fast since no iterations are required. The input layer weights are assigned randomly, while the hidden layer weights are determined analytically in one step. A standard ELM with a single hidden layer and N nodes in the hidden layer is described according to the following equation:

$$y_j = \sum_{k=1}^{N} h_k f(\mathbf{w_k}, \mathbf{x_j}) \tag{6.1}$$

N is the number of samples in the dataset, y_j is the output of the ELM for sample $\mathbf{x_j}$, w_k stands for the weights and biases of the kth node, h_k is the weight that connects the $k_t h$ hidden element with the output layer and f the activation function used (eg sigmoid). The above equation can be written in compact form as:

$$\mathbf{y} = \mathbf{Gh} \tag{6.2}$$

$\mathbf{h}$ is the vector of weights of the output layer and $\mathbf{G}$ is given by:

$$G = \begin{pmatrix} f(\mathbf{w_1}, \mathbf{x_1}) & ... & f(\mathbf{w_n}, \mathbf{x_1}) \\ . & . & . \\ f(\mathbf{w_1}, \mathbf{x_P}) & ... & f(\mathbf{w_n}, \mathbf{x_P}) \end{pmatrix}$$

Accordingly, the mean square error (MSE) function $E(\mathbf{w})$ to be minimized is given by the following equation:

$$E(\mathbf{w}) = \sum_{n=1}^{P} (y(\mathbf{x_n}; \mathbf{w}) - t_n)^2 \tag{6.3}$$

Since the weights that connect that input and the hidden layer are assigned randomly, the output weights $\mathbf{h}$ that minimize the error function given by Equation 6.3 are calculated by Moore-Penroses generalized inverse of G according to the following equation:

$$\mathbf{h_{opt}} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{T} \tag{6.4}$$

where $\mathbf{T}$ is the vector of targets.

6.2.3 ES-ELM

As mentioned, randomly assigning the input weights (connect input layer with the hidden nodes) and bias allows for an one-step calculation of the output weights (connecting the hidden layer with the output layer), therefore makes ELM extremely fast, while at the same time might, under some circumstances, improve the generalization capabilities of the network. However, the random assignment of input weights has its drawbacks as well, the main drawback being that randomness can cause unstable performance with lots of variance.

In this section, a modification of ELM is proposed where, instead of using random assignment, the input weights are being optimized by using an ES. The output weights are still determined analytically in such a way that the training error is minimized, by using the pseudoinverse as in Equation 6.4. The objective function to minimize is in essence the test error of the classifer or the regressor, therefore ES is trying to find optimal input weights and bias that minimize the error on the test set.

However, using the test set and its corresponding labels or targets to optimize the weights or other hyperparameters can lead to unrealistic, biased performance. Specifically, in this case, different input weights are tried and evaluated each time, using information from the test set to evaluate them. Hence, it is almost certain that an optimal weight vector will be found that

minimizes the test error sufficiently, but the same weight vector will perform very poorly on an unknown test set, therefore the actual generalization capabilities of the ELM will be poor.

Another optimization scheme would be to find an optimal weight such that the training error is minimized. The problem withthis approach is overfitting, therefore despite a minimized training error that will probably be quite low, the error on a test set will be much higher, hence the generalization capabilities of the ELM will still be poor.

A much better approach would be to optimize the k-fold CrossValidation (CV) error instead of the train set error or the error of a single test set. This way, an optimal weight vector is one that minimizes the mean CV error, therefore the optimal vector to be found by the ES must minimize the test error on all k folds. Thus, the goal of this scheme is to find a stable model where the model's parameters can yield efficient performance for various test sets. In this case, the model's parameters are the input weights of the ELM, but optimization can include more parameters such as the number of nodes of the ELM, or parameters that affect feature extraction or feauture extraction process. The mathematical formulation of the optimization problem to be solved by the ES are to be found next.

The goal of optimization is to minimize an objective function, which in this case takes as value the average MSE of a k-fold CV scheme for a given input vector $\mathbf{x}$.

$$f(\mathbf{x}) = \sum_{i=1}^{k} MSE(\mathbf{x}, CVfold_i) \tag{6.5}$$

Each CV fold represents a different split of the dataset into training set and test set, therefore there is a different training set and testset for each fold. The MSE for each CV fold is given by Equation 6.3, where $\mathbf{x}$ of Equation 6.5 corresponds to the weights and bias of the input layer of the ELM.

Consequently, the function domain is a vector $\mathbf{x}$, with the vector dimensionality depending on the number of weights and bias of the input layer, which in turn depend on the dimensionality of the feature set d and number of nodes N in the hidden layer in the following way:

$$\#dimensions = dN + N \tag{6.6}$$

In this formula, the product dN corresponds to the number of elements in the $d\ x\ N$ input weights matrix while the second term N corresponds to the length of the $1\ x\ N$ vector of the bias weights.

6.2.4 Regularized ES-ELM (ES-RELM)

Despite employing ES to minimize the mean CV error and find a stable parametric model, it is still possible that the solution found is suboptimal and that the classifier or regressor does not generalize well enough. A regularized version of ELM can further aid in finding a more optimal model with improved generalization capabilities.

Calculating the weights that connect the output and the hidden layer of the ELM can be seen as a regression problem between the hidden and the output layer, described by the following equation:

$$E(\mathbf{w}) = \sum_{n=1}^{P} (y_n - x_n \mathbf{w})^2 \tag{6.7}$$

This is in essence a rewriting of Equation 6.3 with a different notation in order to avoid confusion since Equation 6.7 describes the regression between the hidden layer and the output layer, and not between the input and output layer like in the case of Equation 6.3. According to this notation, in Equation 6.7 and the following Equations 6.8 & 6.9, P is the total number of samples, x_n is the output of the hidden layer for the n_{th} sample, y_n is the actual target value of the sample, $\mathbf{w}$ is the weight vector between the hidden and the output layer, while the rest of the variables are notated the same way as in Equation 6.3.

Regularization in essence adds an extra constraint on the minimization problem. In the case of regularization, an optimal solution $\mathbf{w}$ must not only minimize the mean square error between the predicted and the true targets, but also minimize a term that includes $\mathbf{w}$. For example, for the case that the first norm of the weights is used, the error function to be minimized becomes as follows:

$$E(\mathbf{w}, \lambda) = \sum_{i=1}^{P} (y_n - x_n \mathbf{w})^2 + \lambda \sum_{i=1}^{N} |w_i| \tag{6.8}$$

The added term to the error function is named L1 penalty, because the

first norm of the weights is used, while the respective minimization problem is also known as LASSO (least absolute shrinkage and selection operator) [37].

Accordingly, if an L2 penalty term is added to the error function, the error function takes the following form, also known as Tikhonov regularization or ridge regression problem:

$$E(\mathbf{w}, \lambda) = \sum_{n=1}^{P} (y_n - x_n \mathbf{w})^2 + \lambda \sum_{i=1}^{N} w_i^2 \tag{6.9}$$

In general, an optimal solution to the ridge rigression problem outperforms the Lasso solution in cases were the variables are highly correlated with each other [37].

A hybrid solution combining both L1 and L2 penalties called elastic net is proposed by Zou et. al [38], as a means to overcome the drawbacks of the L1 and L2 approaches. Specifically, elastic net combines both penalties in a weighted manner as given by the following equation:

$$E(\mathbf{w}, \lambda) = \sum_{n=1}^{P} (y_n - x_n \mathbf{w})^2 + \lambda_1 \sum_{i=1}^{N} |w_i| + \lambda_2 \sum_{i=1}^{N} w_i^2 \tag{6.10}$$

The equation can be rewritten as follows, where α controls whether the penalty term is closer to the L1 or L2 norm:

$$E(\mathbf{w}, \lambda) = \sum_{n=1}^{P} (y_n - x_n \mathbf{w})^2 + \lambda \sum_{i=1}^{N} a w_i + \frac{1-a}{2} w_i^2 \tag{6.11}$$

Elastic net is the same as LASSO when $\alpha = 1$ (only an L1 penalty exists). As α tends toward 0, the elastic net approaches ridge regression (L2 penalty becomes predominant).

Apparently, two λ values are included in the equation, λ_1 and λ_2 which therefore need to be optimized. Typically, optimization of the λ_1 and λ_2 (or, if Equation 6.11 is used, λ and α) is done using CV, which can be costly especially since a two-dimensional search is needed [39]. It is hence proposed to also include these parameters in the proposed ES-ELM optimization scheme. This has significant advantages over using CV, since in CV a predifined set of values is tried out, and the search space is searched in an exhaustive manner, while as mentioned ES traverses the search space more intelligently and the parameters to be optimized can be tested in a continuous range.

Chapter 7

Experiments and Results

7.1 Problem Description

The experimental part of this thesis is concerned with the estimation of remaining useful life (RUL) for high-speed CNC milling machine cutters based on collected condition data. The condition data collected for this task can be divided into three types of measurements: 1) force measurements, 2) vibration measurements and 3) acoustic emission measurements. The data are taken from the 2010 PHM Society Conference Data Challenge.

The current problem is of importance since tool failure may result in losses in surface finish and dimensional accuracy of a finished part, or possible damage to the work piece and machine [16]. It is hence important to find a way to predict tool wear, in order to schedule the cutting process accordingly and avoid inaccuracies or even worse surface damage or machine damage. Monitoring of tool wear in order to prevent surface damage is considered as one of the difficult tasks in the context of tool condition monitoring [40].

Specifically, the task is to predict the wear of the cutter's flutes (Figure 7.1). In this thesis, the CNC milling cutter examined uses a triple flute (Figure 7.1c). Diagnosis of tool wear corresponds to modelling the flutes' wear based on the measured time-series data. According to this model, it is possible to estimate how many cuts can be perfomed until the flutes need replacement, defined by a given upper limit in the flute wear. It is then possible to create a cutting schedule according to this model. Thus, it should be apparent that tool life is correlated to tool wear, therefore RUL estimation is performed implicitly through tool wear diagnostics.

(a) Single flute (b) Double flute (c) Triple flute

Figure 7.1: Single, double and triple endmill flutes of a CNC milling machine cutter

7.2 Data

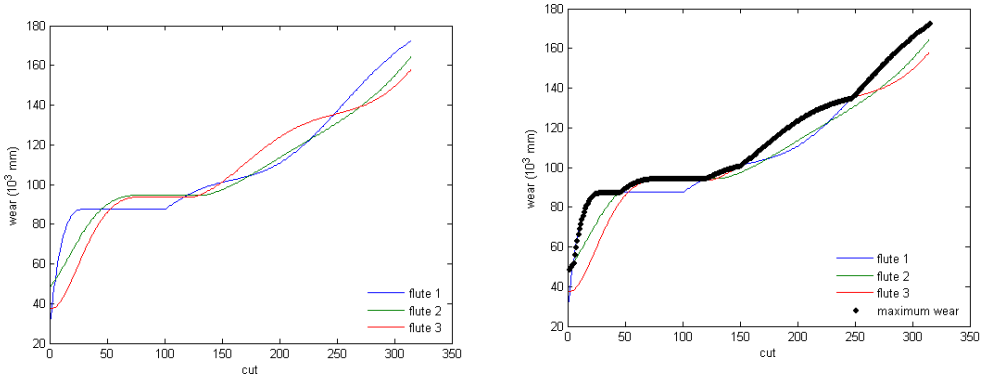
7.2.1 Data Acquisition &Description

All the data are time-series acquired from dynamometer, accelerometer and acoustic emissions, for each of the 315 cuts. Specifically, there are 7 monitored signals for each cut: 3 signals for force measurements (in N) in dimensions X, Y and Z respectively, 3 signals for vibration measurements (in g) in X, Y and Z respectively and 1 signal that measures the root mean square (RMS) value of acoustic emissions (in V).

The data acquisition details are similar to the paper by Li. et al [16], although it should be noted that Li. et al examined CNC milling machine cutter with double flute instead of triple and probably different operating settings too: In the current thesis, the spindle speed of the cutter was 10400 RPM; feed rate was 1555 mm/min; Y depth of cut (radial) was 0.125 mm; Z depth of cut (axial) was 0.2 mm. For the data acquisition, a Kistler quartz 3-component platform dynamometer was mounted between the workpiece and machining table to measure the cutting forces in the form of charges, and converted to voltages by the Kistler charge amplifier. Three Kistler piezo accelerometers were mounted on the workpiece to measure the machine tool vibrations of cutting process in X, Y, Z direction respectively. A Kistler acoustic emission (AE) sensor was mounted on the workpiece to monitor the high frequency stress wave generated by the cutting process.

Each of the 315 cuts is accompanied by a wear file listing the wear (in 10^{-3} mm) of each of the 3 flutes after the respective cut. All signals are sampled at a frequency rate of $50kHz$, although each signal can vary in length. The

time-series data as well as the respective wear data are all in CSV format. The size of all the collected time-series for each cutter is approximately 1GB. The wear value of a flute is in essence indicative of the damage caused to the flute. It should be apparent that the wear can only increase after each cut (Figure 7.2a). Diagnosis and modelling of flute wear is revolved around the maximum wear value out of the three flutes (Figure 7.2b).



(a) Wear across all 3 flutes for cutter c_1

(b) Maximum wear of all 3 flutes highlighted for cutter c_1

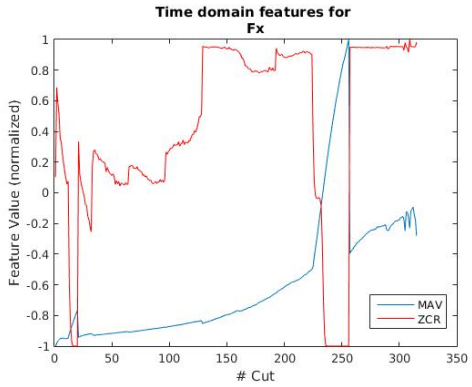
Figure 7.2: a) Wear values after each cut for all 3 flutes for cutter c_1 b) maximum wear highlighted (7.2b) for cutter c_1

7.2.2 Feature Extraction

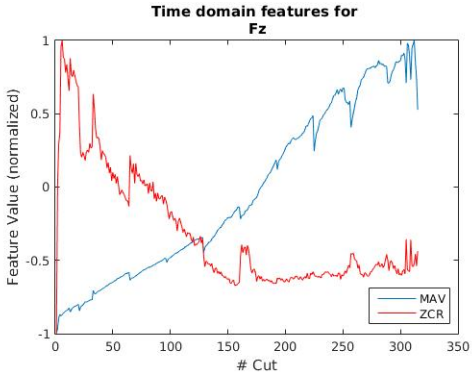
Since all data are time-series data, features have to be extracted from the original time-series data before the data set can be used as input to the algorithms used for the prediction process. Each feature value is extracted from each of the 7 types of signals for each of the 315 cuts , for all 3 available cutters. Thus, the feature set that represents each cutter is a Nxd matrix, where d is the total number of extracted features and $N = 315$ is the number of observations.

The type of extracted features can be categorized into time-domain, frequency-domain and time-frequency domain features (Section 4.2.2). The extracted time-domain features are Mean Absolute Value (MAV) and Zero-crossing rate (ZCR) (Figure 7.3), the frequency-domain features are mean frequency (MF) and median frequency (MDF) (Figure 7.4), while the time-domain frequency features are the average wavelet power at 9 different scales

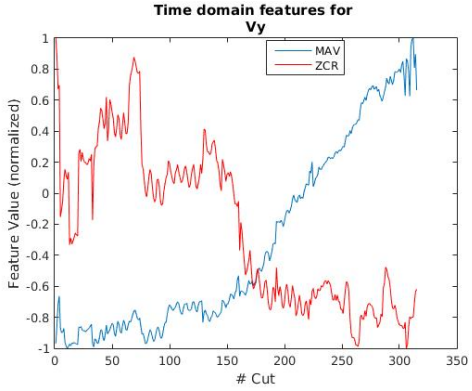
corresponding to 9 frequency bands, acquired through CWT.



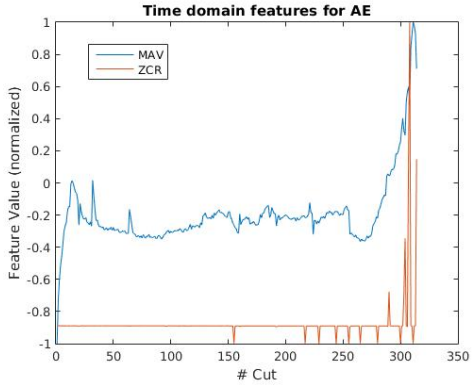
(a) Time-domain features for F_x



(b) Time-domain features for F_y



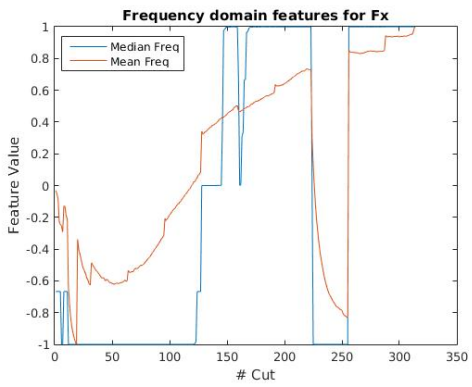
(c) Time-domain features for V_z



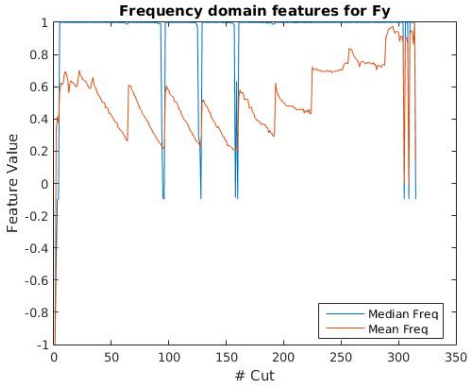
(d) Time-domain features for AE

Figure 7.3: Time domain features extracted from force (Fx,Fy), Vibration (Vz) and AE signals for cutter c_1

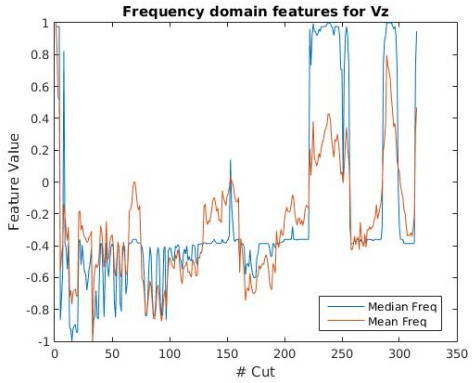
It can be seen that the MAV of force and vibration signals grows higher, which makes sense since higher tool wear values mean that higher force has to be applied in order to cut the piece. No useful information appears to be provided as far as ZCR is concerned.



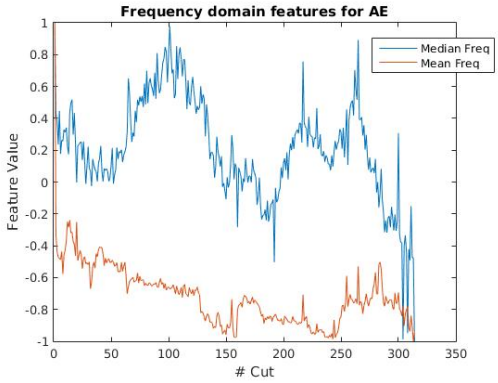
(a) Normalized frequency-domain features for F_x



(b) Normalized frequency-domain features for F_y



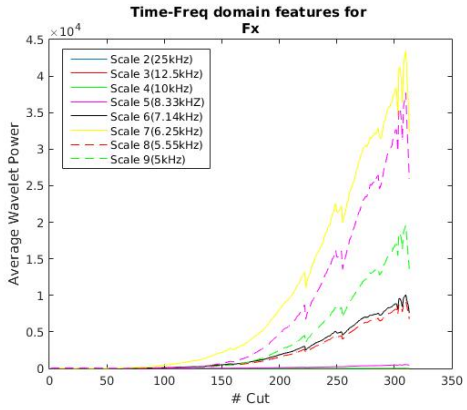
(c) Normalized frequency-domain features for V_z



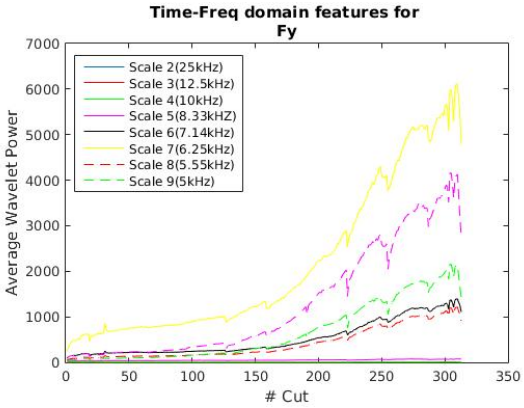
(d) Normalized frequency-domain features for AE

Figure 7.4: Frequency-domain features extracted from force (Fx,Fz), Vibration (Vx) and AE signals for cutter c_1

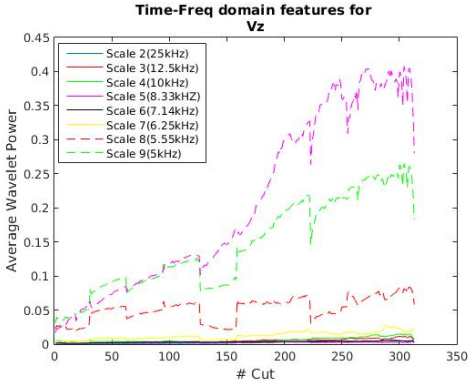
It can be seen that for some cases (Figure 7.4b, 7.4d) median frequency is a constant feature. For the rest of the cases, although there can be seen some relevancy to the targets especially regarding mean frequency, the features seem to contain large outliers and be noisy. In general it can be said that the quality of frequency-domain features is significantly lower than that of time-domain (Figure 7.3) or time-frequency domain features (Figure 7.5).



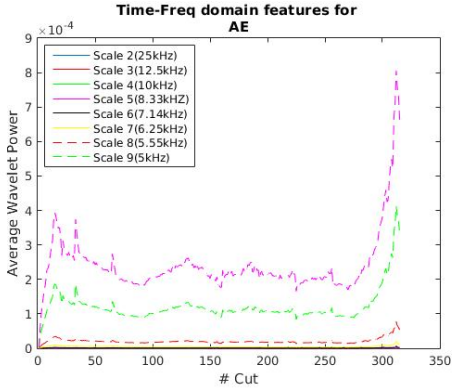
(a) Average wavelet power for F_x



(b) Average wavelet power for F_y



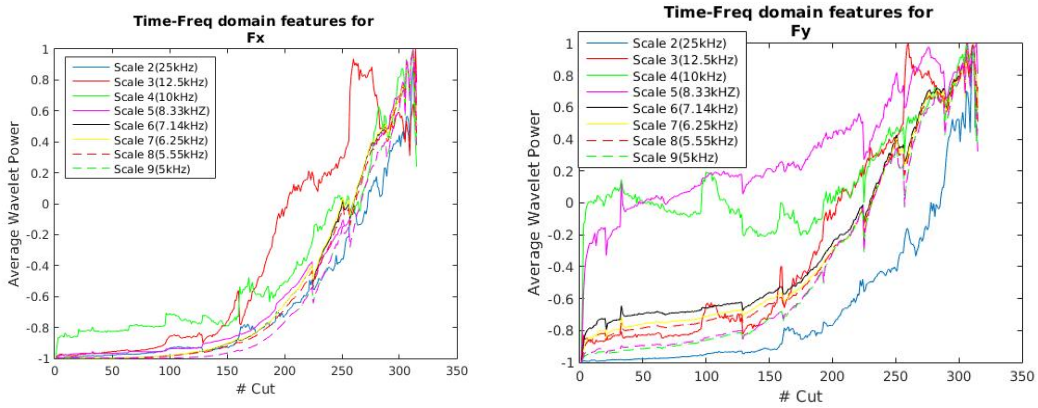
(c) Average wavelet power for V_z



(d) Average wavelet power for AE

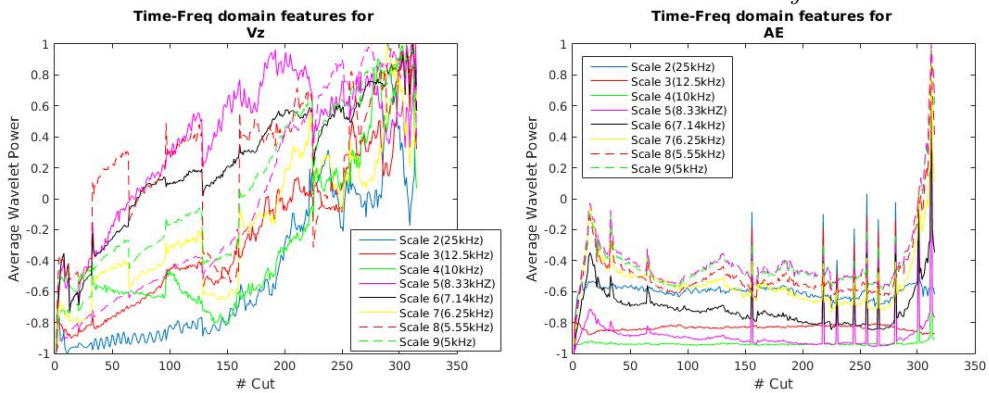
Figure 7.5: Time-Freq domain features extracted from force (Fx,Fy) and Vibration (Vz) signals for cutter c_1

From the time-frequency features it can be seen that some frequency bands (the 3 lowest ones) are more dominant than others (Figure 7.5). However, the extracted information from different frequency bands seem to be highly correlated, which can be seen more clearly in the normalized feature set of Figure 7.6.



(a) Average wavelet power (normalized) for F_x

(b) Average wavelet power (normalized) for F_y



(c) Average wavelet power (normalized) for V_z

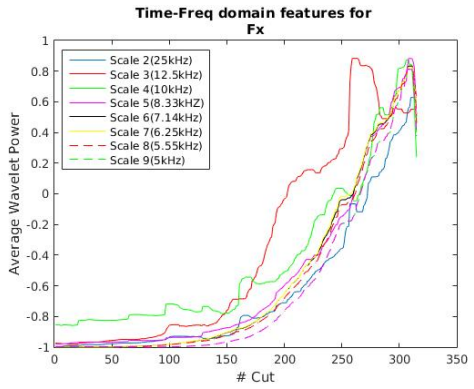
(d) Average wavelet power (normalized) for AE

Figure 7.6: Normalized Time-Freq domain features extracted from force (Fx,Fy) and Vibration (Vz) signals for cutter c_1

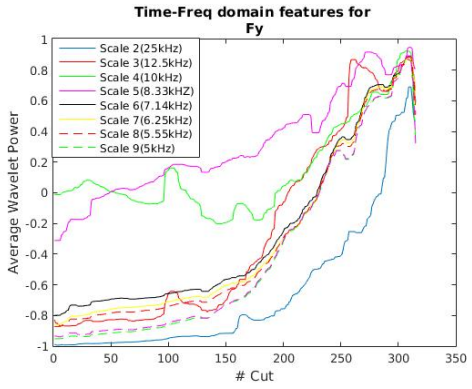
In general, it can be observed that the extracted features contain outliers, therefore observations that are distant from the rest of the observations. Specifically, it can be seen that after certain cuts, the observed measurements show deviations and fluctuations, which in overall results in a peaky and noisy feature set. These kind of fluctuations make the features less relevant and consequently might be an obstacle for the machine learning stage. Despite the outliers, which are present in all three types of features, it is noticed that time-frequency domain features and in some cases time-domain features can be informative and relevant to the targets, while frequency domain features

are the most noisy and irrelevant. It is also noticed that extracted features from force and vibration signals are generally informative, while acoustic emission signals are much more noisy and irrelevant.

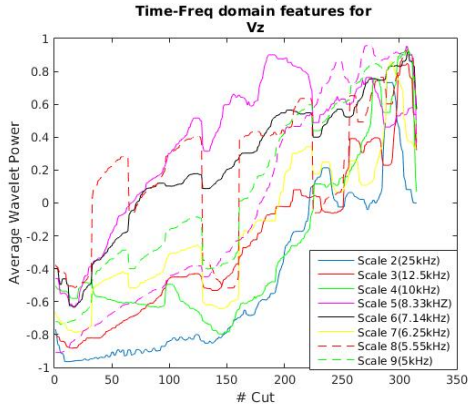
According to the above, one way to improve the features would be to remove or replace the outliers and smooth out the feature set as well. One way to achieve both goals would be to use the median filtering method proposed in Section 4.2.1. Median filtering can both smooth out the observed set and also attenuate the effect of outliers, as can be seen in Figure 7.7. However, although there is a small improvement in the sense that features seem smoother and the outliers are reduced or show smaller deviations, the aforementioned drawbacks of the features still remain: many of the features are not relevant enough to the targets, while even the features that show higher correlation to the targets are highly redudant, therefore providing very similar information. This can be seen as a sign that further feature engineering or feature transformation is required with the aim of facilitating the next stage of machine learning and regression in specific.



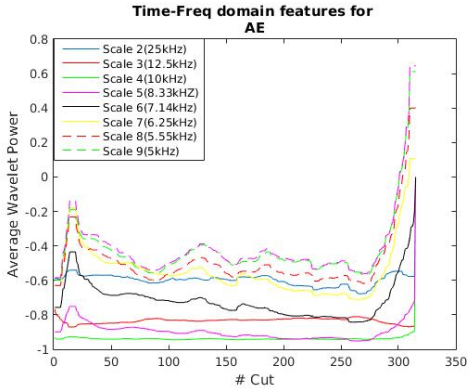
(a) Average wavelet power (normalized & smoothed) for F_x



(b) Average wavelet power (normalized & smoothed) for F_y



(c) Average wavelet power (normalized & smoothed) for V_z



(d) Average wavelet power (normalized & smoothed) for AE

Figure 7.7: Normalized & Smoothed Time-Freq domain features extracted from force (Fx,Fy) and Vibration (Vz) signals for cutter c_1

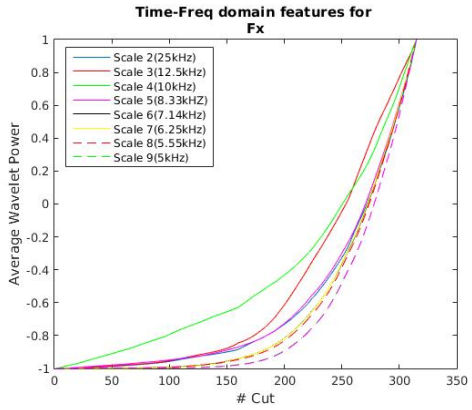
A closer examination of the targets (Figure 7.2) might hint at what could be a useful transformation of the features. As mentioned, the wear can only increase or stay the same after each cut, which consequently means that the wear is a monotonic function with regard to the number of cuts. Moreover, the wear of a flute at each cut depends on the wear value of the previous cut, which in turn depends on the previous cut, and so on. Hence, the wear value of a flute depends on all previous cuts.

Taking these two points into account, it should be clear that ideally a

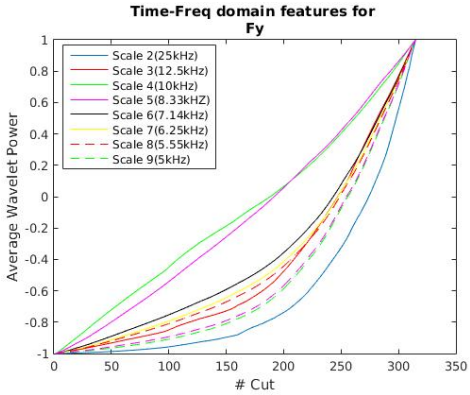
single relevant feature should meet the following criteria: First, it should have to be a monotonic function with regard to the number of cuts (although not necessarily increasingly monotonic) and secondly it should depend on all the previous feature values corresponding to earlier cuts. Accordingly, an efficient way to transform the features would be to take the cumulative sum of the feature values at each of the 315 cuts instead.

This approach is convenient and effective since all outliers are removed and features are smooth (Figure 7.8), plus at the same time features have the potential to become more relevant to the problem to targets. Indeed, it is found that the correlation of all features with the targets is increased. However, there is still high redudancy among the features. It should be noted that although these remarks are about cutter c_1, they hold valid for the other two cutters c_4 and c_6. A plot of all features, for the cumulative case, for all 3 cutters can be found in Figure 7.9.

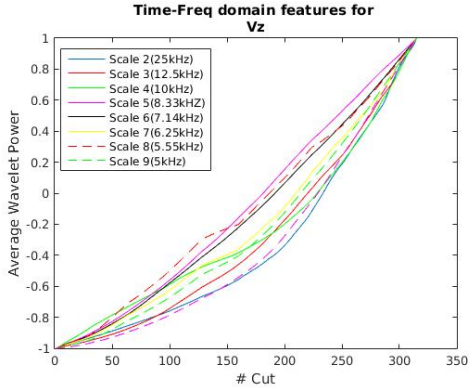
Since the number of total features is 91, a dimensionality reduction method should be applied in order to reduce the number of features and consequently the number of required numbers and the search space for the ES optimization scheme. For this purpose, the variation of SRP suitable for regression problems (Section 4.2.3) is used. Before using the SRP-PCA algorithm, the number of features is first reduced either by selecting the features that show that highest correlation with the targets, or by using FCBF (Section 4.2.3) which also takes into feature redudancy. The final feature sets for all 3 cases of a 3-fold CV scheme can be seen in Figure 7.10.



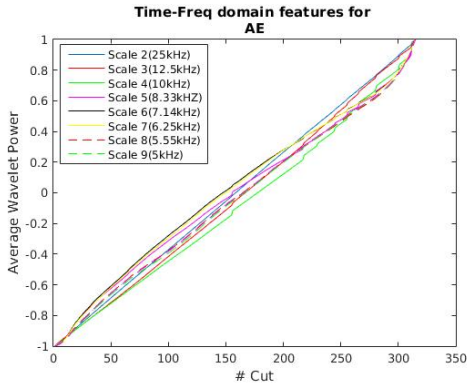
(a) Cumulative average wavelet power (normalized) for F_x



(b) Cumulative average wavelet power (normalized) for F_y

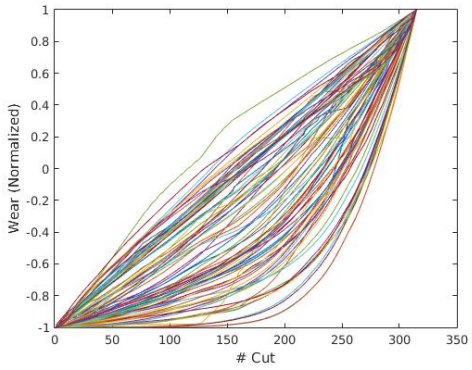


(c) Cumulative average wavelet power (normalized) for V_z

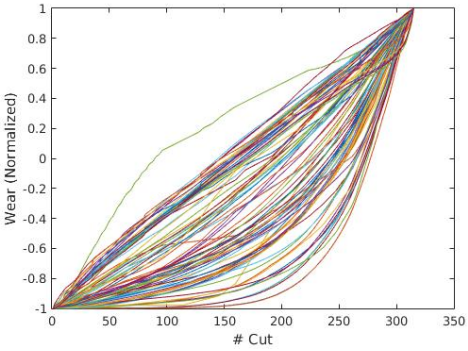


(d) Cumulative average wavelet power (normalized) for AE

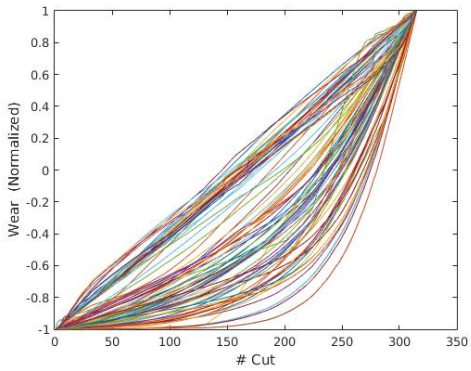
Figure 7.8: Normalized Cumulative Time-Freq domain features extracted from force (Fx,Fy) and Vibration (Vz) signals for cutter c_1



(a) Normalized Cumulative features for all signals for cutter c_1

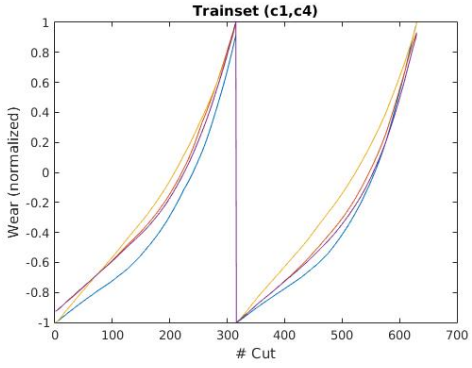


(b) Normalized Cumulative features for all signals for cutter c_4

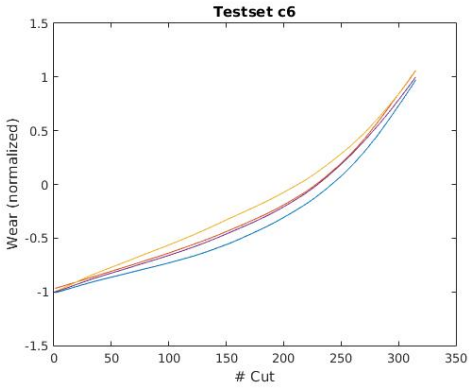


(c) Normalized Cumulative features for all signals for cutter c_6

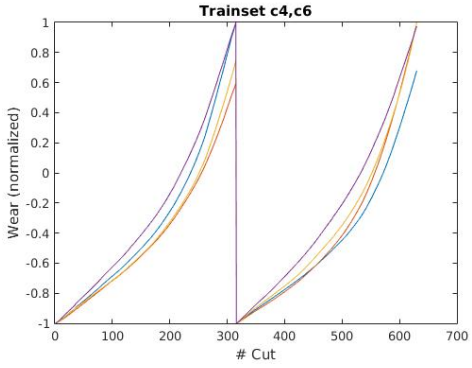
Figure 7.9: All Normalized Cumulative features for all measured signals for cutters c_1 (a), c_4 (b), c_6 (c)



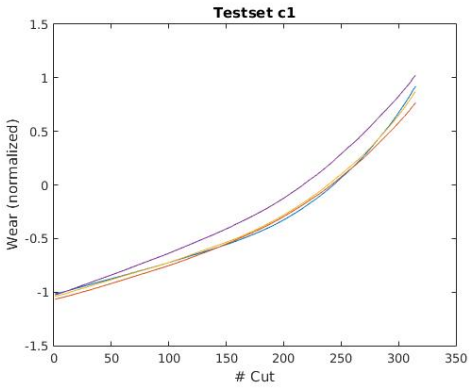
(a) Features after SRP for $c1$ and $c4$ as trainset



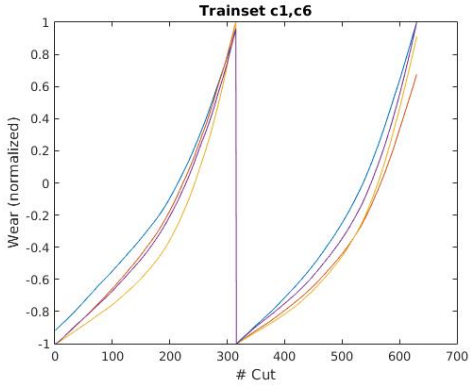
(b) Features after SRP for c_6 as testset



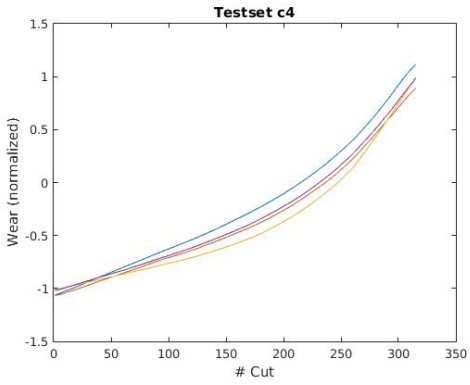
(c) Features after SRP for $c4$ and $c6$ as trainset



(d) Features after SRP for c_1 as testset

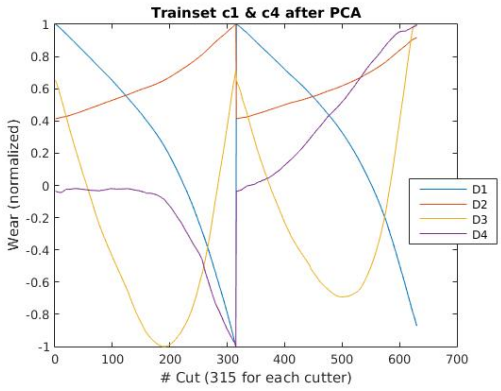


(e) Features after SRP for $c1$ and $c6$ as trainset

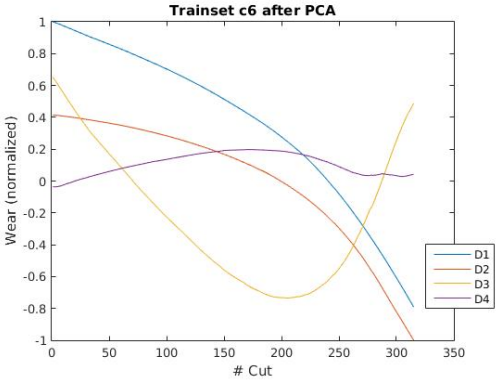


(f) Features after SRP for c_4 as testset

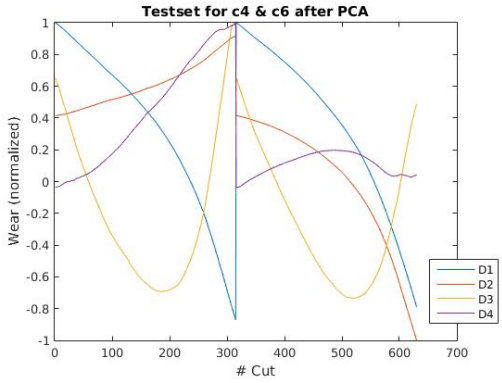
Figure 7.10: Features after SRP for all 3 folds of the 3-fold CV



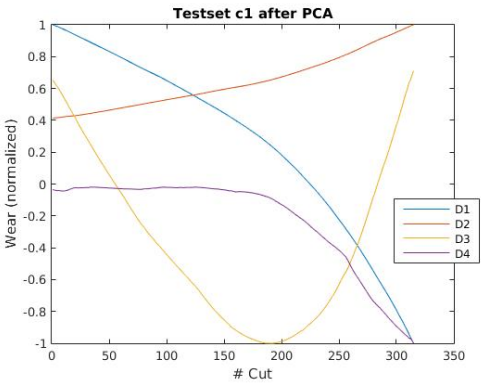
(a) Features after PCA for $c1$ and $c4$ as trainset



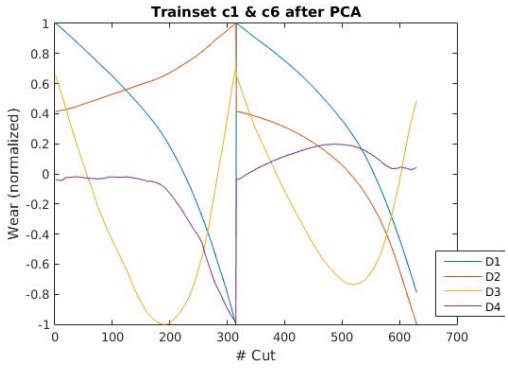
(b) Features after PCA for c_6 as testset



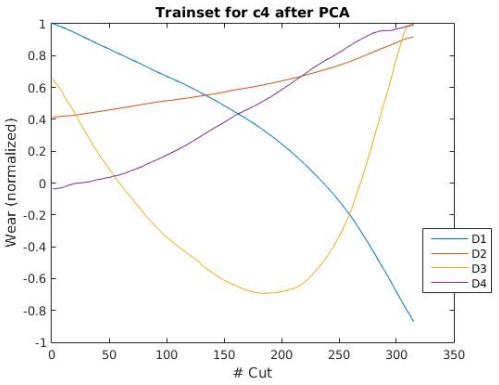
(c) Features after PCA for $c4$ and $c6$ as trainset



(d) Features after PCA for c_1 as testset



(e) Features after PCA for $c1$ and $c6$ as trainset



(f) Features after PCA for c_4 as testset

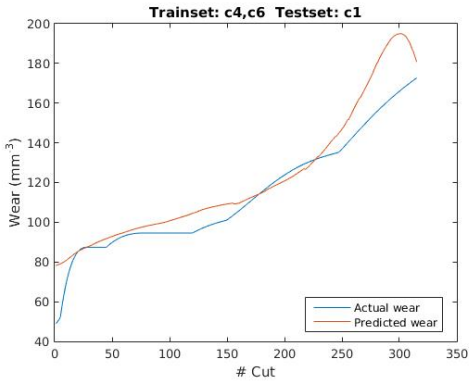
Figure 7.11: Features after PCA for all 3 folds of the 3-fold CV

7.3 Regression & Wear model

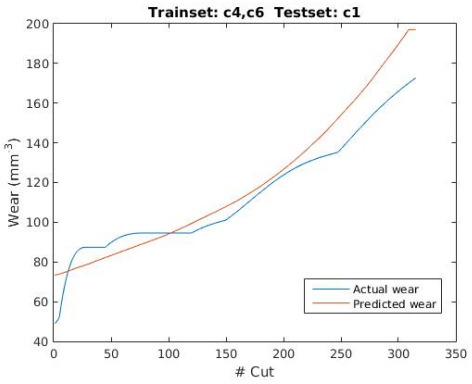
Once the final feature set is acquired it is fed as input to the maintenance decision support system. Within the scope of this thesis, experiments have been performed with different types of feedforward neural networks used as regressors, where the feature set is fed to the input layer of the neural network and the output layer corresponds to the predicted wear values.

As mentioned already in Section 6.2 feedforward ANNs and ELM both have their limitations. Briefly, ANNs are slow to learn the weights due to the iterative nature of the back-propagation algorithm, and furthermore it is quite probable that back-propagation gets stuck in local optima. On the other hand, ELM might be fast and avoid iterative learning, but the totally random assignment of the input weights and biases results causes large variance in the performance. Hence, all the experiments reported here involve employing either ES-ELM or ES-RELM (proposed in Section 6.2.3). All experiments that are reported here use 10 nodes in the hidden layer.

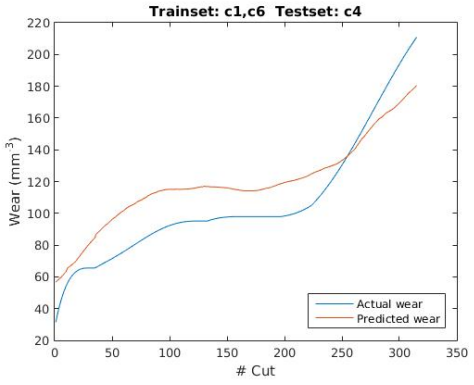
As far as the ES optimization part is concerned, both experiments involving ES-ELM and ER-RELM aim in finding a stable model, in specific a set of input weights and biases that minimize the average MSE for 3-fold CV. Since all experiments involve 10 nodes and a final number of 4 features, the objective function to be optimized by the ES has $4 * 10 + 10 = 50$ dimensions (according to Equation 6.6) or $50 + 2 = 52$ dimensions for the case of ES-RELM where regularization is used through the elastic net (Equation 6.11). Using a higher number of nodes should not change the results dramaticaly because regularization is used by means of elastic net which applies a mixed L1 and L2 penalty. Thus, many of the weights connecting the nodes of the hidden layer with the utput layer are practically pruned or forced to take lower values, affecting the output in a lesser degree. Moreover, it is noteworthy that median filtering (Section 4.2.1) with a neighborhood size of 11 is applied at the output of the layer. In practice this is found to perform better, most likely because the predicted output is smoothed by taking into account the neighborhood of a prediction instead of a single prediction. This way, it is possible for the ES to find some solutions that can be improved when median filtering is applied at the end. Hence, overall there is a slightly increased probability of finding better solutions. The evaluation budget for all experiments reported here is 5000 objective function evaluations.



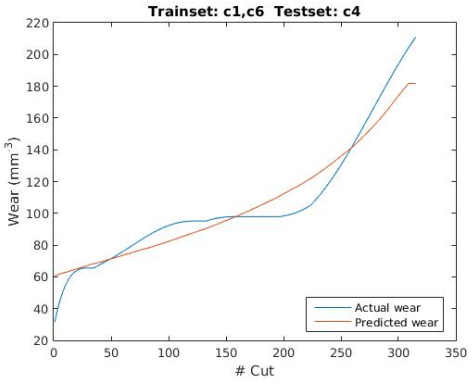
(a) Best model found for testset c_1 using SRP feature set (ES-ELM)



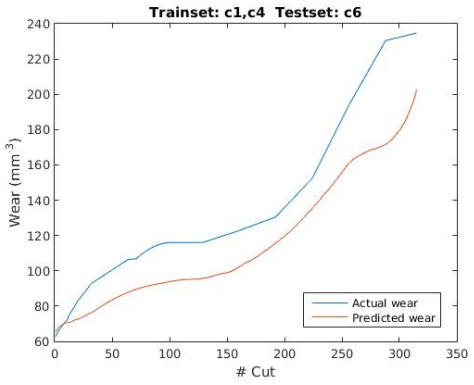
(b) Best model found for testset c_1 using SRP feature set (ES-RELM)



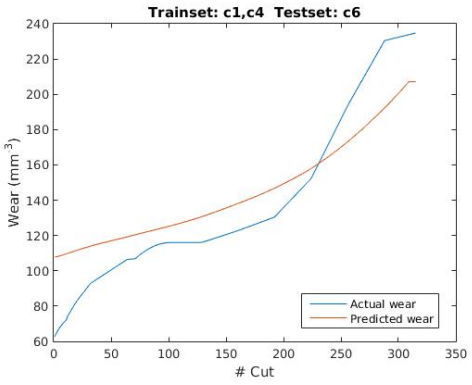
(c) Best model found for testset c_4 using SRP feature set (ES-RELM)



(d) Best model found for SRP feature set (ES-RELM)



(e) Best model found for testset c_6 using SRP feature set (ES-ELM)



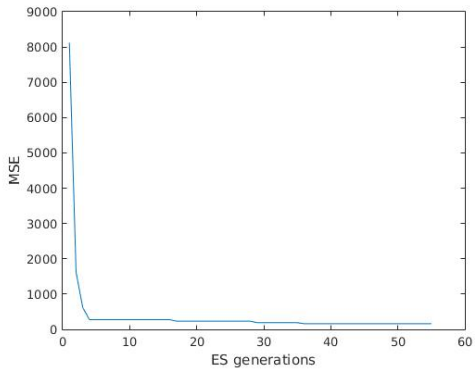
(f) Best model found for testset c_6 using SRP feature set (ES-RELM)

Figure 7.12: Predicted wear for 3 cutters of 3-fold CV for the normal (left) and regularized case (right)

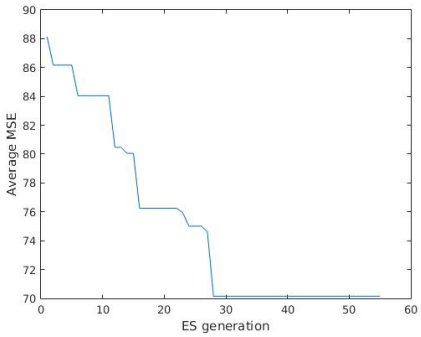
For the case where PCA is performed at the whole feature set of 91 features, the ES-ELM without regularization yields around the same results as for the case where FCBF and SRP-PCA are performed. However, when regularization is used, ES-RELM performs much better. The first experiment gives an MSE of 80 (Figure 7.13a), which is further optimized to 75 (Figure 7.13b). The results containing the average MSE of the 3-fold CV for the 2 feature sets with ES-ELM and ES-RELM are summarized in the following table. The parameters of the elastic net for the ES-RELM experiments are $\lambda = 1.68$ and $alpha = 0.84$ for the FCBF + SRP-PCA case and $\lambda = 0.26$ and $alpha = 0.66$ for the PCA case.

Experimental Setup	FCBF+SRP-PCA	PCA
ES-ELM	412.26 (5.38)	464.44 (5.31)
ES-RELM	221.42 (5.31)	70.14 (16.43)

Table 7.1: Average testset MSE for the 3-fold CV (average trainset MSE error in parenthesis)

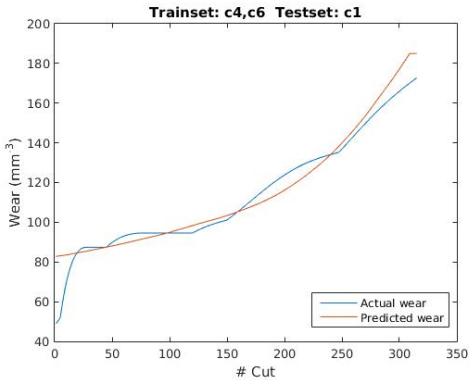


(a) ES graph optimization

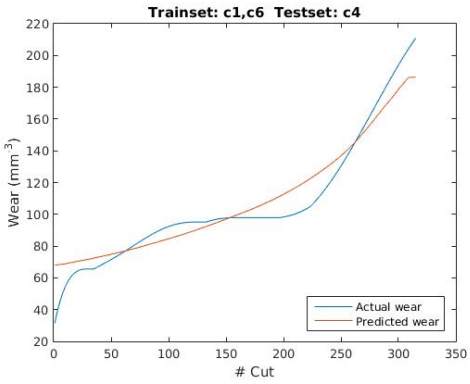


(b) ES graph starting from a previously found optimal candidate solution

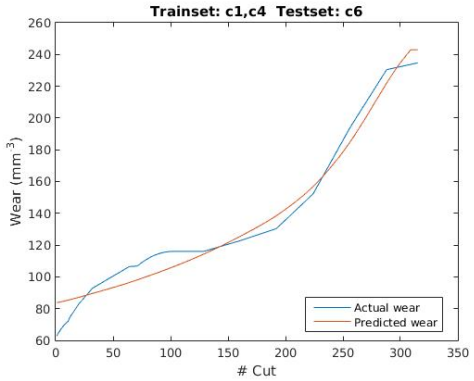
Figure 7.13: ES-RELM optimization



(a) Predicted performance for cutter c_1



(b) Predicted performance for cutter c_4



(c) Predicted performance for cutter c_6

Figure 7.14: Best Predicted performance for 3-fold CV when PCA is performed on all 91 features

Chapter 8

Conclusions & Discussion

In conclusion, it can be said that the main bottleneck of the framework lies in the feature extraction and feature selection part. Initially, the time-domain features extracted from the raw time-series using Continuous Wavelet Transform seem to be the most informative, although all features are noisy and contain outliers. Transforming all features by using the cumulative values at a given cut, therefore summing up all previous extracted features from the previous cuts, seems to be crucial and is recommended as an important and useful step for tool condition monitoring of CNC milling machine cutters, for the reasons explained in the previous chapter. The number of different scales for wavelet analysis is 9, which means that analysis is perfrormed at 9 different bands. Since in the scope of this thesis the interest is on the average wavelet power at each band, and there is high correlation between different frequency bands, choosing a different number of bands is in general not crucial to the feature extraction process and the final results, provided of course that this number is not too low.

Furthermore, it is confirmed that EAs and in specific the proposed ES can be useful as far as efficient learning of the parameters of an ELM is concerned. ES is able to quickly find a sufficiently good solution and is also able to escape local optima. Regularization also seems to be crucial for finding a stable model and ES-RELM experiments significantly outperform the ES-ELM experiments without regularization. Although the training error is less for the ES-ELM experiments, the test error is much lower for ES-RELM, which confirms that the generalization capabilities of the model are improved de to regularization.

It is also noteworthy that the feature set where PCA is performed on

the whole feature set outperforms the one where FCBF and SRP-PCA are applied. This could be due to the fact that SRP-PCA projects to the first dimension that maximizes variance, while PCA projects to the first 4 dimensions with maximum variance. Since there is high redudancy among the features, even after FCBF, projecting each feature subset into the first PCA dimension will still result in features that are highly correlated with each other (Figure 7.10). On the other hand, performing PCA in the whole feature set and projecting to 4 dimensions results in 4 features that are indepedent to each other, even if the original feature set contains redundant features. As far as the number of features is concerned, projecting to a greater number of features is in essence pointless because they explain practically zero variance of the data. Also, since the number of feautures is not that big, performing PCA is not that computationally expensive.

It can be argued that the proposed ES-RELM variation of ELM is not really an ELM but just a type of Single Layer Forward Network, since for the first part the input weights and bias , and for the second part an elastic net is used to define the connectiong weights instead of determining them analytically by taking the pseudoinverse like in the ELM or ES-ELM case, which is not feasible since regularization is used.

8.1 Future Research

In this section a few suggestions will be proposed, believing that they might potentially be proven useful for future research as far as tool condition monitoring of CNC milling machine cutters is concerned or possibly other machine learning problems that deal with feature extraction from time-series.

First, since the main bottleneck of the framework lies in the feature extraction part, it is crucial to be able to extract or engineer features that can provide information about the targets in more detail. One idea in this direction would be, apart from average wavelet power, to extract other measures by using CWT, like pair-wise comparisons of time-series in different cuts that can show first order (linear) or second order (non-linear) phase or amplitude correlation. For example, pair-wise comparison between a signal and the signals corresponding to previous and next cuts can be done.

Moreover, as far as SRP-PCA is concerned, it could be modified to project each SRP subset to 2 or 3 dimensions instead of just one, as a way to produce independent, non-redundant features.

As far as the ES-RELM part is concerned, two main augmentations could be tried. First, it could be tried to run the ELM for more than one time and average the outputs of each run, with different input weights and bias or regularization parameters each time. Of course, this would mean that the optimization search space would grow, but the proposed modified ES is known to perform well in multidimensional space.

Taking into account the significant performance boost due to the regularization used by the elastic, an even better performance could be achieved by trying other regularization techniques such as the normalized elastic net, composite absolute penalty and Owen's hybrid penalty reviewed by Miche et. al [39]. Moreover, another idea worth trying regarding regularization is to add an extra layer to the ELM. The output of the ELM could pass from an extra layer of nodes with sigmoid activation function and then regularization could be performed again via another elasticnet. This only adds two extra parameters for the ES optimization scheme, the regularization parameters of the extra elastic net.

Last but not least, using a wrapper approach for feature selection instead of a filter one could also improve the performance, although it would significantly increase the required computational time.

Bibliography

[1] Butala P. Sluga A. and Peklenik J. A conceptual framework for collaborative design and operations of manufacturing work systems. *IRP Annals Manufacturing Technology*, 54, 2005.

[2] Ranganathan M. Faloutsos C. and Manolopulos Y. Fast subsequence matching in time-series databases. *ACM SIGMOD Record*, 23, 1994.

[3] Weiss G. Mining with rarity: a unifying framework. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter*, 6, 2004.

[4] Leonardi S. Lankford J. Lin J., Keogh E. and Nystrom D. Visually mining and monitoring massive time series. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 460–469, 2004.

[5] Weigend A. and Gershenfeld N. *ime Series Prediction: forecasting the future and understanding the past*. Addison Wesley, 1994.

[6] Lin J. and Keogh E. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, 8(2).

[7] Bakshi B. and Stephanopoulos G. Representation of process trends in induction of real-time patterns from operating data for diagnosis and supervisory control. *Computers & Chemical Engineering*, 18(4).

[8] Keogh E., Chu S., Hart D., and Pazzani M. *Segmenting time series: A survey and novel approach*. World Scientific Publishing, 1994.

[9] Aladesaye M. Application of predictive maintenance to industry including cepstrum analysis of a gearbox. *Massey University Auckland, New Zealand*, 2008.

[10] Jeffery D. *Principles of Machine Operation and Maintenance*. Butterworth Heinemann Ltd, Oxford, Great Britain, 1991.

[11] Proth J.M. Chu C. and Wolff P. Predictive maintenance:the one-unit replacement model. int. j. production economics. *Int. J. Production Economics*, pages 285–295, 1998.

[12] Pintelona L. Pinjala S.K. and Vereecke A. An empirical investigation on the relationship between business and maintenance strategies. *Int. J. Production Economics*, pages 214–229, 2006.

[13] Swanson L. Linking maintenance strategies to performance. int. j. production economics. *Int. J. Production Economics*, pages 237–244, 2001.

[14] Fu M.C. Yao X., Fernandez-Gaucherand E. and Marcus S.I. Optimal preventive maintenance scheduling in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 17, 2004.

[15] Jardine A. K.S., Lin D., and Banjevic D. *A review on machinery diagnositcs and prognostics implementing condition-based maintenance*. Elsevier Ltd., 2006.

[16] Li X., Lim B.S., Zhou J.H., Huang S., Phua S.J, Shaw K.C., and Er M.J. Fuzzy neural network modelling for tool wear estimation in dry milling operation. *Annual Conference of the Prognostics and Health Management Society*, 2009.

[17] Chen Z.Y., He Y.Y, Chu F.L, and Huang J.Y. Evolutionary strategy for classification problems and its application in fault diagnostics. *Engineering Applications of Artificial Intelligence*, 16:31–38.

[18] Yan G.T. and Ma G.F. Fault diagnosis of diesel engine combustion system based on neural networks. In *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics*, volume 5, pages 3111–3114, Shanghai, China, 2004.

[19] Huang Y.C. nd Huang C.M. Evolving wavelet networks for power transformer condition monitoring. *IEEE Transactions on Power Delivery*, 17(2):412–416.

[20] D. Percival and A. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge University Press, 2nd edition, 1994.

[21] D. Percival and A. Walden. *The Illustrated Wavelet Transform Handbook*. Institute of Physics Publishing, Bristol and Philadelphia, 2002.

[22] Kiymik MK, Akin M., and Subasi A. Automatic recognition of alertness level by using wavelet transform and artificial neural network. *Journal of Neuroscience Methods*, 139, 2004.

[23] Torrence C. and P. Glibert. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79(1).

[24] Pyle D. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, Los Altos, California, 1999.

[25] Fitzgerald W.J., Ruanaidh J.J.K.O., and Yates J.A. *Generalised Changepoint Detection*. University of Cambridge Engineering Department, 1994.

[26] Prochazka A., Kukal J., , and Vysata O. Wavelet transform use for feature extraction and eeg signal segments classification. *3rd international symposium on communications, control, and signal processing*, pages 719–722, 2008.

[27] Phinoyomark A., Limsakul C., and Phukpattaranont P. Application of wavelet analysis in emg feature extraction for pattern classification. *Measurement Science Review*, 11(2), 2011.

[28] Kilby J. and Prasad K. Continuous wavelet analysis and classification of surface electromyography signals. *International Journal of Computer and Electrical Engineering*, 5(1), 2013.

[29] Hall M. Correlation-based feature selection for discrete and numeric class machine learning. *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 359–366, 2005.

[30] Das S. Filters, wrappers and a boosting-based hybrid for feature selection. *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[31] Yu L. and Lui H. Feature selection for high-dimensional data: A fast correlation-based filter solution. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

[32] Zhao R. and Mao K. Semi-random projection for dimensionality reduction and extreme learning machine in high-dimensional space. *IEEE Computational Intelligence Magazine*, 10(3), 2015.

[33] Bianchi L., Dorigo M., Gambardela L.M., and Gutjahr W.J. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: an international journal*, 8(2), 2009.

[34] Blum C. and Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3), 2003.

[35] Beyer HG and Schwefel HP. Evolution strategies - a comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[36] Kramer O., Ting C.-K, and Buning H.K. A new mutation operator for evolution strategies for constrained problems. *IEEE Congress on Evolutionary Computation*, 2005.

[37] Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 1994.

[38] Zou H. and Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2), 2005.

[39] Miche Y., Mark van Heeswijk, Bas P., Simula O., and Lendasse A. Tropelm: A double-regularized elm using lars and tikhonov regularization. *Neurocomputing*, 74, 2011.

[40] Sick B. Review on-line and indirect tool wear monitoring in turning with artificial neural networks: A review of more than a decade of research. *Mechanical Systems and Signal Processing*, 16(4):487–546, 2002.