Lappeenranta University of Technology

School of Energy Systems

Master's Thesis

Wavelet Transform and k-NN Method in Sensor Data Analysis

Tuukka Falkenberg

Supervisors Prof. Pertti Silventoinen and D. Sc. Mikko Kuisma

Abstract

Author: Tuukka Falkenberg

Title: **Wavelet Transform and k-NN Method in Sensor Data Analysis**

Master's thesis. Lappeenranta University of Technology, School of Energy Systems. 2016.

35 pages, 14 pictures, 1 table.

Supervisors: Professor Pertti Silventoinen, D. Sc. Mikko Kuisma.

Keywords: Signal analysis, machine learning, wavelet transform, k-Nearest Neighbors, sensor data

Industrial and commercial processes are packed with sensors and devices which produce signal data from the processes. These signals are used to monitor, control, and improve the processes or parts of them, usually in quite simple and limited ways. However, they can also provide much more information and value when analyzed more thoroughly with advanced signal analysis tools. This thesis studies the use of the wavelet transform, which is a type of time-frequency transformation, and a data classification and regression method known as the k-Nearest Neighbors method, as such signal analysis tools. The mathematical theory of the wavelet transform and k-Nearest Neighbors method is presented briefly, and their suitability in sensor signal analysis is discussed.

The objective of this thesis is to develop a flexible, adjustable, and highly automated method for signal analysis. An algorithm which utilizes the wavelet transform and k-NN method as a machine-learning-based tool is presented, and the principle and development possibilities of the tool are explained. A prototype algorithm based on the presented tool is tested with sensor data from a highway traffic sensor, and its functionality is examined with an underlying emphasis on the possibility of developing the algorithm further into an automated signal analysis tool that requires minimum user involvement.

Tiivistelmä

Tekijä: Tuukka Falkenberg

Työn nimi: **Aallokemuunnos ja k:n lähimmän naapurin menetelmä anturidatan analyysissä**

Diplomityö. Lappeenranta University of Technology, School of Energy Systems. 2016.

35 sivua, 14 kuvaa, 1 taulukko.

Tarkastajat: Professori Pertti Silventoinen, TkT Mikko Kuisma.

Avainsanat: Signaalianalyysi, koneoppiminen, aallokemuunnos, k:n lähimmän naapurin menetelmä, anturidata

Teollisissa ja kaupallisissa prosesseissa on huomattava määrä antureita ja muita laitteita, jotka tuottavat signaalidataa näistä prosesseista. Tuotettuja signaaleja käytetään prosessien tai niiden osien tarkkailemiseen, ohjaamiseen ja kehittämiseen, yleensä hyvin yksinkertaisilla ja rajallisilla tavoilla. Niiden avulla voisi saada kuitenkin huomattavasti enemmän tietoa ja arvoa tutkimalla niitä tarkemmin monipuolisempien signaalianalyysityökalujen avulla. Tässä diplomityössä tutkitaan aallokemuunnoksen, tietyn tyyppisen aika-taajuusmuunnoksen, ja luokittelu- ja regressioanalyysissä käytettävää k:n lähimmän naapurin menetelmän käyttöä työkaluina anturidatan analyysissä.

Työn tavoitteena on kehittää joustava ja helposti muokattava sekä mahdollisimman pitkälle automatisoitu menetelmä anturidatan analyysiin. Aallokemuunnoksen ja k:n lähimmän naapurin menetelmän matemaattinen teoria käydään lyhyesti läpi, ja niiden soveltuvuutta signaalianalyysiin arvioidaan. Niillä toteutettu koneoppimiseen perustuva analyysityökalu esitetään, ja sen toimintaperiaatetta ja kehitysmahdollisuuksia tutkitaan. Esitettyyn analyysityökaluun perustuvaa prototyyppialgoritmia testataan käyttämällä koedatana erään moottoritien rampin liikennemääräanturista kerättyä dataa, ja algoritmin toimivuutta tarkastellaan huomioiden samalla mahdollisuuksia kehittää algoritmista automatisoitu signaalianalyysityökalu joka vaatisi mahdollisimman vähän käyttäjän toimia.

Foreword

The opportunity to write this thesis became quite unexpectedly, and has been an interesting and exciting journey to remember. It would not have been possible without the guidance and inspiration from my supervisors, professor Pertti Silventoinen and D. Sc. Mikko Kuisma, whom I am extremely grateful to. I also appreciate tremendously my colleagues and employer for giving the possibility to work on this thesis freely, and providing helpful feedback along the way.

My family and friends have proven to be invaluable with their help and support. Thanks to mom, dad, and my brother, as well as the band mates and all my other dear friends for your patience and cordiality.

Table of contents

List of used symbols and abbreviations

ω	Angular velocity
ψ	Wavelet function
W	Wavelet transform

CMF	Conjugate Mirror Filter
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
DTWT	Discrete-Time Wavelet Transform
FIR	Finite Impulse Response
FT	Fourier Transform
IIR	Infinite Impulse Response
IoT	Internet of Things
K-NN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
STFT	Short-Time Fourier Transform
QMF	Quadrature Mirror Filter

1. Introduction

1.1 Sensor data in industrial and commercial processes

Collecting and analyzing process data usually has the primary goal of reducing costs or providing more value for the process operator or client. This additional value can be reached with different methods, for example by discovering deviations and/or correlation of different process variables that allow better process control, maintenance, or development possibilities. However, traditionally knowledge and insight about the processes in many industries are reliant on human memory and intuition, which make them more susceptible to miscalculations, human error and misunderstandings in addition to the inherent complexity and risks of managing such tacit knowledge. Human observation also has definite limits, which can impede the efficient analysis and development of data processing and process control.

As the booming electronics and digitalization has taken over the industrial and commercial sectors, things like smart devices, the Internet of Things, blockchains and other similar digital technology trends are becoming more and more prominent, not only as part of the newest devices and technologies but also as updates and retrofits to older applications. According to a report by IDC, the amount of IoT devices was already 15 billion in 2015, and is expected to grow to 200 billion by 2020 (MacGillivray & Turner 2015). Most of these devices are expected to be found in the business and manufacturing sectors, with estimated economic impact of 1,2 to 3,7 * 10^{12} dollars (McKinsey 2015). This leads to a dramatic increase in the importance and utilization of the IoT and other digital technologies in the future, and they will enable easy and cost-efficient implementation of a wide variety of process measurement and signal analysis possibilities.

These technologies also allow to gather very large amounts of data from almost any and all processes, but in order to find any meaningful information in large amounts of complex, information-sparse or seemingly irrelevant data, a more refined approach must be used instead of a person sifting through the data. One such approach is the application of

machine intelligence, or machine learning, which have recently been the subject of countless studies as well as business and industry development projects. A machine learning approach practically bypasses the inherent human factors in an analysis process and opens the possibility of automated, computerized numerical and logical analysis which greatly surpasses the capabilities of humans and reduces their workload. For signal analysis, this means that it is no longer necessary to understand the content of the measured data; if a correlation between the data and the objective of the analysis exists, it can be discovered with the right approach.

1.2 Motivation of this work

The objective of this thesis is to create and test a signal analysis tool for discovering variations in the measured signal data of practically any sensor. Suitable mathematical methods for the analysis tool are selected based on research literature of similar signal analysis cases, and a computational method for the signal analysis of sensor data is proposed and examined. The method focuses on detecting meaningful differences or variations in measured signal data, which imply changes in the process or the machinery associated with it. An algorithm to test this principle is constructed by utilizing a two-step data-analysis with a time-frequency analysis and a machine learning part, in order to make the algorithm adaptive to many different signals, and functional in different environments and processes with minimum user input. This algorithm is tested with sensor data measured from a highway ramp, and the results of the tests are examined to determine how effectively it works. Based on the results of the tests and studied research literature, the possibilities of developing the algorithm further is discussed. Several important aspects of the algorithm and its future development are left out of the scope of this work, and are only mentioned in relevant chapters.

2. Data analysis methods

The essence of computational data analysis lies in the possibility of performing extensive calculations upon data in a reasonable period of time. Due to advancements in digital processing capabilities of modern computers, very thorough and complex mathematical analysis can be carried out with fairly low computational effort. Depending on what kind of data is being observed, different tools to analyze and process it can be used. The tools described in this chapter are mathematically flexible, and can be used for multi-dimensional data as well as 1-dimensional signal analysis. However, this work focuses only on 1-dimensional data for its obvious connection to analog sensor devices and their signal output.

Frequency analysis is the backbone of all digital signal analysis. Observing and manipulating the frequency content of a signal can reveal information that would be very difficult or impossible to find by analyzing the signal in the time domain. However, for signals that change with respect to time, a static frequency-plane analysis might not be enough. Depending on the signal, the time at which certain events or frequencies occur might be as important or even more so than the frequency itself. One solution to improve upon this static frequency analysis and reduce the problem with time-variant signals is to utilize time-frequency analysis.

2.1 Time-frequency analysis

To obtain information about the frequency content of a measured or generated signal, a Fourier transform is by far the most commonly used mathematical analysis tool. It can be described as a transform of a signal from time domain into frequency domain. However, the biggest deficiency of the traditional Fourier transform is, that the transform does not give information about the time at which a certain frequency is observed. Instead, it only reveals the amplitudes and phase shifts of the frequency components present in the whole signal. To overcome this limitation and improve the analysis capabilities, so called time-frequency analysis methods can be utilized.

Time-frequency analysis consists in essence of any mathematical operation where a signal is transformed from a time-amplitude -domain into a time-frequency -domain. In other words, the signal is transformed and partially localized both in frequency spectrum as well as time, as opposed to being only frequency-localized like in the basic Fourier transform. In this way the information about when certain events happen can be recovered in addition to the frequencies caused by these events.

One well-established and very widely used time-frequency analysis methods is the Short-Time Fourier Transformation. The STFT is based on the Fourier transform, and expands upon it by splitting the signal into small sections with a process called windowing, and performing the Fourier transform independently to the pieces of the signal. This results in time-localized frequency information about the signal. To increase the usability of the STFT, the signal parts can be overlapped, resulting in a more detailed, continuous representation of the time-frequency content. Even with these capabilities, the STFT transform does have its limits. The STFT has a rigid time-frequency window, which means that there is always a fixed, static compromise between the time and frequency resolution at all frequencies and times. (Chui 1992) Because of this, the STFT has always limited capabilities of distinguishing events in both time and frequency planes. This leads to the conclusion that the analytic capabilities of the STFT with signals containing both high and low frequencies and/or transients will be far from ideal.

2.1.1 Wavelets

To improve upon the capabilities of the STFT, a more recent time-frequency analysis operation known as the wavelet transform has been developed and researched in the past decades. The mathematical principles of wavelet transforms can be applied to many different fields of study and different source data, but in this thesis the application of wavelets is limited to a 1-dimensional continuous wavelet transform, suitable for the measurement data of, for example, an analog sensor device. The basic idea of a 1-dimensional wavelet transform is that instead of decomposing a (1-d) signal into a sum of sinusoids, as in the Fourier transform, a different basis function called a wavelet function is used. This wavelet function is translated across the signal for time localization and dilated

so that small and large-scale correlation of the signal with the wavelet function is obtained. The scales used in the wavelet transform correspond to frequency bands in the signal. This results in a time-frequency representation of the original signal. The most important advantage of the wavelet transform compared to STFT is that instead of a fixed tradeoff between time and frequency resolution, wavelets allow a high frequency resolution at low frequencies and high time resolution at high frequencies. This allows a substantially better analysis of complex signals that contain a multitude of frequencies or transients. (Mallat 2009)

2.1.2 Continuous wavelet transform

A continuous wavelet transform can be seen as the wavelet-based equivalent of the Short-Time Fourier Transform. In fact, the two can be thought of as partially identical operations when a Morlet wavelet is used in the CWT, as the Morlet wavelet's basis function is a sine wave multiplied with a Gaussian window. In comparison, the Fourier transform uses a continuous sine wave as the basis function and the STFT adds a window function that is used to "cut" the signal into time-localized pieces that are then transformed with the FT to gain a time-frequency representation. The difference of STFT and CWT is that the dilation and translation of the wavelet function allows the time and frequency resolutions of the CWT to change according to the frequency and time step in which it is being calculated.

The core element of a continuous wavelet transform is a wavelet function. A wavelet function can be defined as a function $\psi(t)$ which satisfies two requirements; the function must integrate to zero:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{1}$$

This is usually, but not necessarily, a characteristic of an oscillating function. The second requirement is that the wavelet function must be square integrable:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \tag{2}$$

Which can be interpreted as the function having finite energy. These conditions are sufficient to define a wavelet function for a continuous wavelet transform, but in order for function $\psi(t)$ to be a *mother wavelet,* which opens the possibility of a mathematical operation called an inverse CWT, it must satisfy the admissibility condition:

$$C \equiv \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega, \qquad 0 < C < \infty \tag{3}$$

For a function *f(t)* that fulfills at least the criteria defined in equations (1) and (2), the continuous wavelet transform is defined as:

$$W(a,b) \equiv \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-b}{a}\right) dt \tag{4}$$

Where a is the scale factor and b is the shift factor, or dilation and translation factors, respectively. The result of this definition is that the wavelet transform can be essentially regarded as a collection of the inner products (or cross-correlation) of the signal *f(t)* and the translated and dilated wavelet $\psi_{a,b}(t)$ for all a and b. (Rao & Bopardikar)

By calculating a Fourier transform of the wavelet function, it becomes clear that the wavelet function is essentially a bandpass filter (Mallat 2009). This leads to the idea that the wavelet transform can be viewed as a collection of the outputs of the signal *f(t)* filtered through bandpass filters with a changing frequency band as a function of the scale factor a. The Fourier transform of a wavelet function can be expressed as:

$$F\left[\psi\left({}^{t}/_{a}\right)\right] = |a|\Psi(a\omega) \quad (5)$$

Thus the center frequency and the -3 dB bandwidth of a wavelet dilated with any scale factor *a* is $1/|a|$ times the center frequency and -3 dB bandwidth of the mother wavelet. In other words, the Q-factor of the wavelet bandpass filter is constant. By analyzing the wavelet function further, it can be derived that the RMS duration and bandwidth of the wavelet function are linked:

$$\Delta t_{\psi}(a)\Delta\omega_{\psi}(a) = c_{\psi} \quad (6)$$

This means that the product of the duration and bandwidth of the wavelet function is constant. For small values of the scale factor *a,* the wavelet function has a small RMS duration which results in good time resolution, e.g. the CWT produces an accurate localization of signals in the time domain. However, the RMS bandwidth of the wavelet function is large, and thus the CWT does not separate frequencies close to each other for small values of *a*. For large values of *a,* the opposite holds true, and the CWT has good frequency resolution because of the low bandwidth and can thus separate small variations in the frequency domain, but is not localized precisely in the time domain.

The ability of the CWT to distinguish two different frequencies or events in time can be illustrated by using so-called time-frequency cells shown in figure 1. Events inside one cell cannot be distinguished from each other, and the shape of the cell depends only on the scale factor *a*. As mentioned above, the product of duration and bandwidth, which equals the area of a time-frequency cell, is constant. (Rao & Bopardikar)

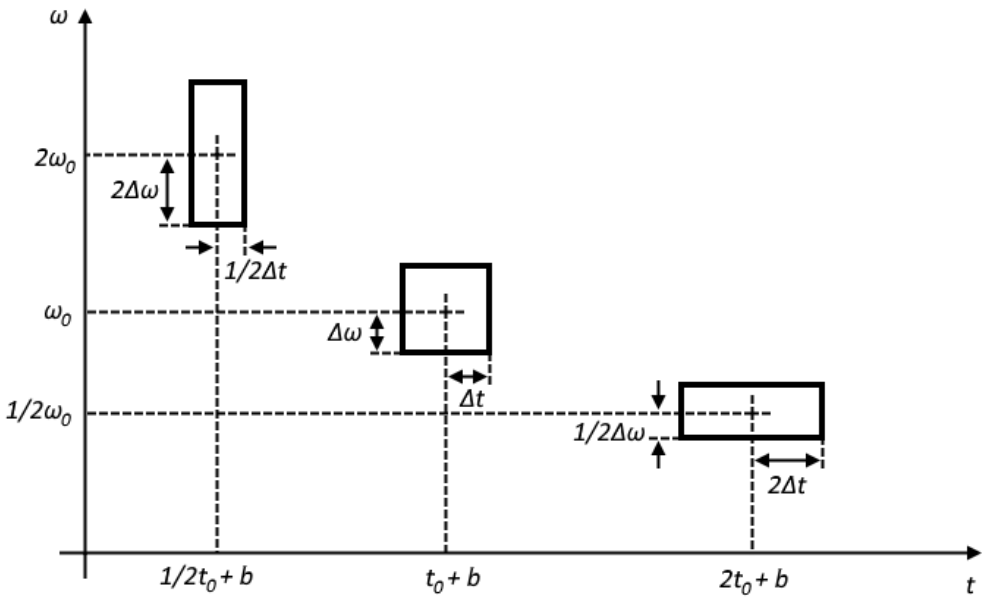

Figure 1. Time-frequency cells of a wavelet function, also known as Heisenberg boxes.

The smaller the area of the time-frequency cells is, the better resolution the wavelet transform yields for both time and frequency. However, there is a limitation to how small the area can be, expressed by the Heisenberg *uncertainty principle:*

$$\Delta t_{\psi}(a)\Delta\omega_{\psi}(a) \geq \frac{1}{2} \tag{7}$$

This is the theoretical limit of the resolution of the wavelet transform, associated with the Gaussian function. However, as the Gaussian function does not satisfy the requirements of a wavelet, the resolution of a wavelet transform is in practice always lower. (Rao & Bopardikar)

As shown above, the dilation of the wavelet changes the frequency spectrum of the wavelet function and the translation of the wavelet reveals the signals' correlation with respect to time. The wavelet transform thus produces a time-frequency energy-density representation of the function *f(t).* The key advantage of the WT compared to STFT is that the time and frequency resolution are variable, which allows the wavelet transform to

reveal more specific and intuitive information about a signal compared to the STFT in many cases. (Mallat 2009)

2.1.3 Discrete wavelet transform

One drawback of the continuous wavelet transform is that it quickly becomes mathematically very demanding even with very simple functions. The region of support for the transform *W(a,b)* can be defined as the set of pairs of the scale and shift factors *(a,b)* for which *W(a,b) ≠ 0*. For the CWT, there is no limit to the region of support and it can thus be the entire plane defined by $\mathbb{R}^2$, which implies that the transform would need infinite number of calculations to complete. However, the CWT has inherently some redundancy which means that the entire support of *W(a,b)* is not needed to recover *f(t)*. (Rao & Bopardikar) A discrete wavelet transform exploits this attribute, by discretizing the scale and shift parameters. It is important to note that the function *f(t)* can still be a continuous function, and the discretization refers only to the transform's parameters *a* and *b*. The DWT can be thought of as analogous to the Fourier series, for both represent a continuous-time signal in a discrete frequency domain. DWT can be expressed as a function with discrete variables *k* and *l* in the form

$$f(t) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} d(k,l) 2^{-\frac{k}{2}} \psi(2^{-k}t - l) \tag{8}$$

Where the values *d(k,l)* are the coefficients of the wavelet transform at $a = 2^k$ and $b = 2^k * l$. As such, the sampling of the DWT is dyadic, with the next level of dilations producing a time-frequency cell twice the size of the previous level in one plane, and half the size in the other plane. This means that the DWT forms a time-frequency graph as shown in figure 2. The time-frequency cells contain the coefficients of the wavelet transform, and cells higher in frequency have double the time resolution but half of the frequency resolution of the adjacent cells at a lower frequency.

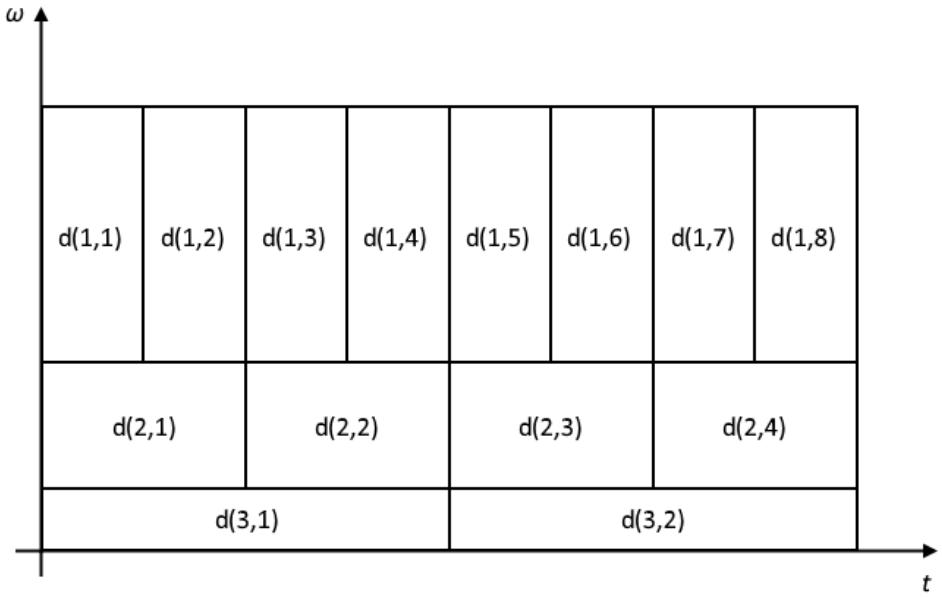

Figure 2. Time-frequency table generated by the DWT.

The DWT provides a non-redundant presentation of the signal in the time-frequency plane. However, a major drawback of the DWT is that it is time-shift sensitive. Shifting the signal backwards or forwards in time will cause the DWT coefficients to change drastically, and the DWT of a certain signal will yield significantly varying coefficients depending on its location in the time plane.

2.1.4 Filter banks and wavelet packets

One way of representing the wavelet function is that it forms a bandpass filter, which is then dilated and convoluted with the signal *f(t)* in the wavelet transform. In the case of a filter, the dilation refers to reducing the filter cut-off or passband frequency. Logically, this implies that it is possible to create a WT of a signal by implementing a digital filter bank composed of suitable filters. These filters must satisfy certain conditions in order for the transform to be mathematically precise, which allows the reconstruction of the original signal from the transform coefficients. Such filters are called perfect reconstruction filters. (Mallat 2009)

Quadrature mirror filters (QMF) exhibit this property, but they require the use of the simple Haar filter or IIR filters (Mallat 2009), which complicates fast processing in digital systems. A similar conjugate mirror filter (CMF) bank, depicted in figure 3, is a filter bank which can be used to describe the wavelet transform, and it allows the use of more complex and FIR filters. The difference between a QMF and a CMF are different filter coefficients, but the filter bank structure is identical for both. The name of these filter banks refers to the fact that the filters *h[n]* and *g[n]* have a mirrored frequency response, one being a low-pass filter and the other being a high-pass filter with the same cutoff frequency. The cut-off frequency of these filters is at the *quadrature point* of the frequency response, which equals $\pi/2$. (Weeks 2006) In the following paragraphs, the *h[n]* refers to a low-pass filter and *g[n]* to a high-pass filter.

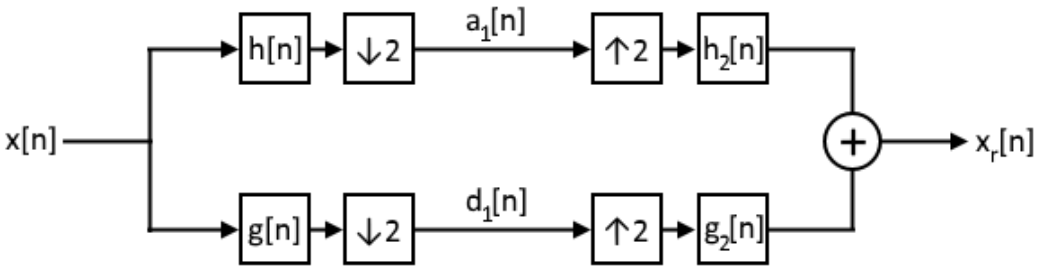

Figure 3. A conjugate mirror filter bank, which allows decomposition and perfect reconstruction of the input signal *x[n]*.

In the CMF, the signal *x[n]* is filtered through the high- and low-pass filters *h[n]* and *g[n]* and then decimated to produce coefficients, which are divided into detail coefficients, marked $d_i[n]$, and approximation coefficients $a_i[n]$ at the corresponding level *i.* On the right side of the filter bank, the original signal is reconstructed by inserting zeros as every other value to the detail and approximation coefficients and then filtering the resulting signals before adding them to create a perfect reconstruction of the original signal, $x_r[n]$. The filters $h_2[n]$ and $g_2[n]$ are dual filters, identical to the *h* and *g* filters. The transfer functions of the conjugate mirror filters are defined by the filter condition:

$$|h(\omega)|^2 + |h(\omega + \pi)|^2 = 2 \tag{9}$$

By taking advantage of the properties of CMFs, filter banks of arbitrary size can be constructed. An example of a three-level wavelet filter bank is shown in figure 4, describing the wavelet packet tree structure of a DWT.

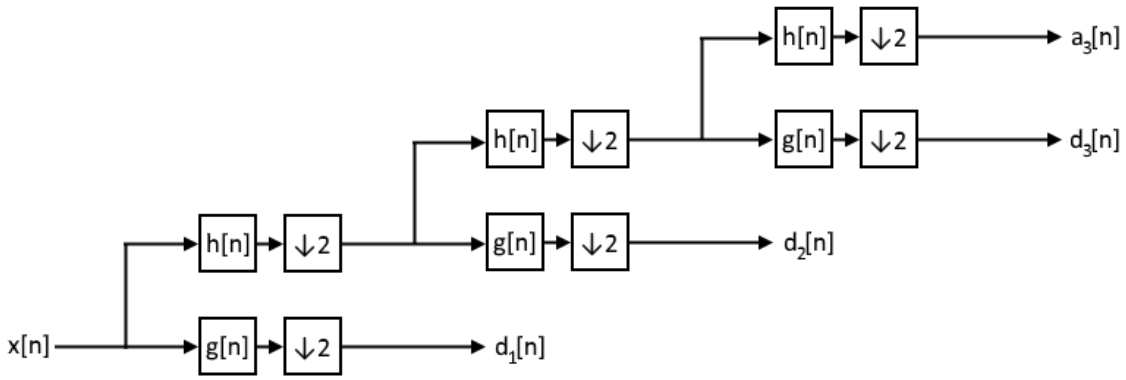

Figure 4. A wavelet filter bank, or a wavelet packet (decomposition) tree of a discrete wavelet transform with three levels of coefficients.

In a DWT, the approximation coefficients produced by the low-pass filters *h[n]* are decimated and passed to the next filter level, whereas the detail coefficients of the high pass filters *g[n]* are decimated and then returned as the detail coefficients of each level. However, in general, the low-pass filter coefficients can be also filtered through next level of high- and low-pass filters, and the wavelet packet tree can be of arbitrary structure (Mallat 2009). A full-depth binary tree is shown in figure 5.

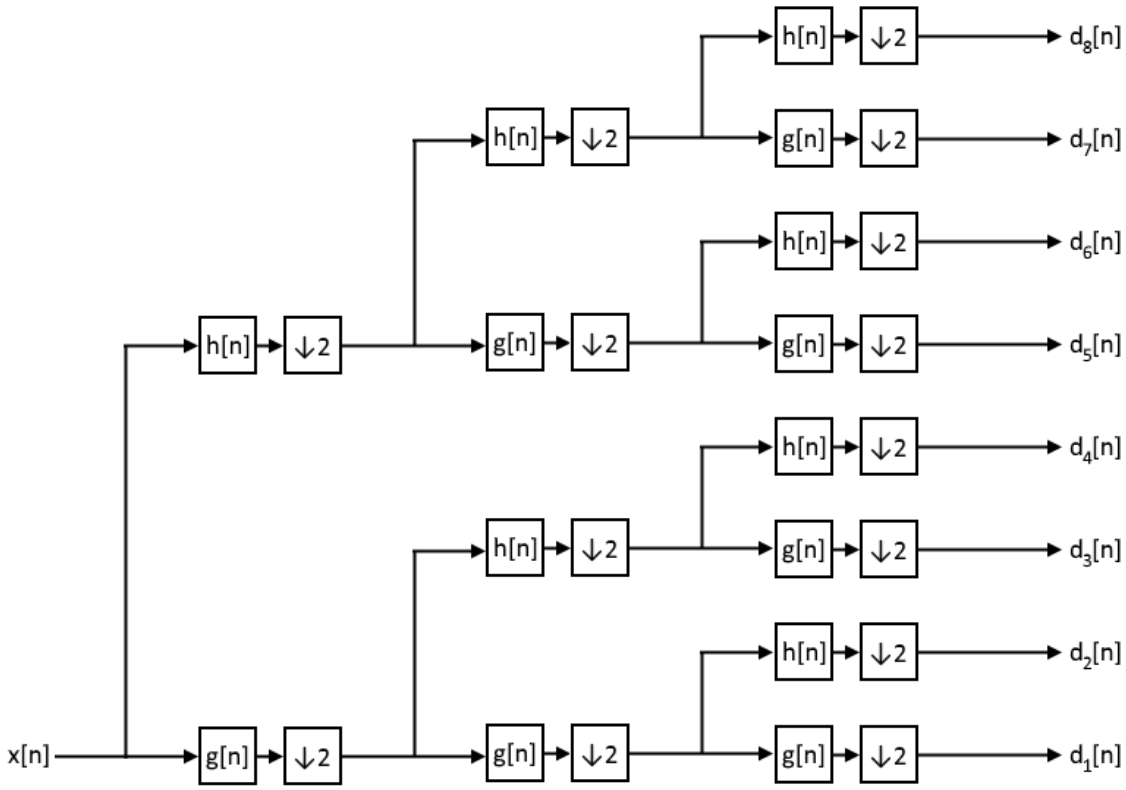

Figure 5. A full-depth wavelet packet decomposition tree.

Due to the decimation, the wavelet packet decomposition tree provides a non-redundant representation, regardless of the tree structure used (Mallat 2009). The tree structure affects only the shape of the time-frequency cells resulting from the filtering. With a full-depth wavelet packet decomposition tree as shown in figure 5, the time-frequency cells resulting from the transform will have identical dimensions in all frequency bands, similar to the STFT.

Filter banks allow an efficient way of performing a wavelet transform on time-discrete data, especially in modern computing applications. This provides a practical approach to time-frequency analysis, with good adaptation possibilities to different input signals.

2.2 Machine learning and regression analysis

Machine learning is one of the newest and most widely researched techniques in modern computing applications, for example in the condition monitoring of mechanical or electrical

devices. The core of machine learning is a machine or system that can process data without direct human interaction, or without following only static instructions that are predefined. Instead, it produces decisions or predictions using models generated completely or partially by the machine itself based on the input data. These models can contain anything from simple regression analysis to complex artificial neural networks or evolutionary methods.

There are countless different subfields of machine learning, as well as machine learning and pattern recognition algorithms, depending on the application and mathematical background. Some methods applied in research related to different industrial processes and applications include Support Vector Machines (Boldt & Ribeiro 2014), Case-Based Reasoning (Deng et al. 2015) and k-Nearest Neighbors method (Bouguerne et al. 2011). All these methods are based on some input data which is used to create a model by the algorithm. This model is then used to classify new input data into pre-defined or algorithm-generated categories. By categorizing the input data, for example, to failure-free and faulty cases, a machine learning -based condition monitoring system can be implemented without the need to understand or account for the data itself. Machine learning methods also allow the efficient use of complex, data-rich and information-sparse data by reducing the need for human interaction and comprehension of the data (Wuest et al. 2016).

2.2.1 K-Nearest Neighbors classification

One of the simplest and oldest algorithms commonly regarded as machine learning is the k-Nearest Neighbors algorithm, used for both classification and regression of data. The k-NN algorithm is based on calculating a distance or similarity between two or more instances of data. This distance can be used to find the nearest points (“neighbors”) of the data, which can reveal a connection between the data points. The number of nearest neighbors to consider for the classifier operation is a key parameter for the k-NN method. It can change the results of the algorithm drastically, and should be chosen on case-by-case basis with respect to the used data.

The data in a k-NN classifier is usually divided into two sets: training data and input data. The training data set is used to make the classification models or baseline of the regression analysis, and the input data is compared to the training data to determine the distance

between the data sets. Based on this distance, a classification is made depending on how many nearest neighbors, or instances of training data, are used for the classification. (Dougherty 2013) This process is illustrated in figure 6.

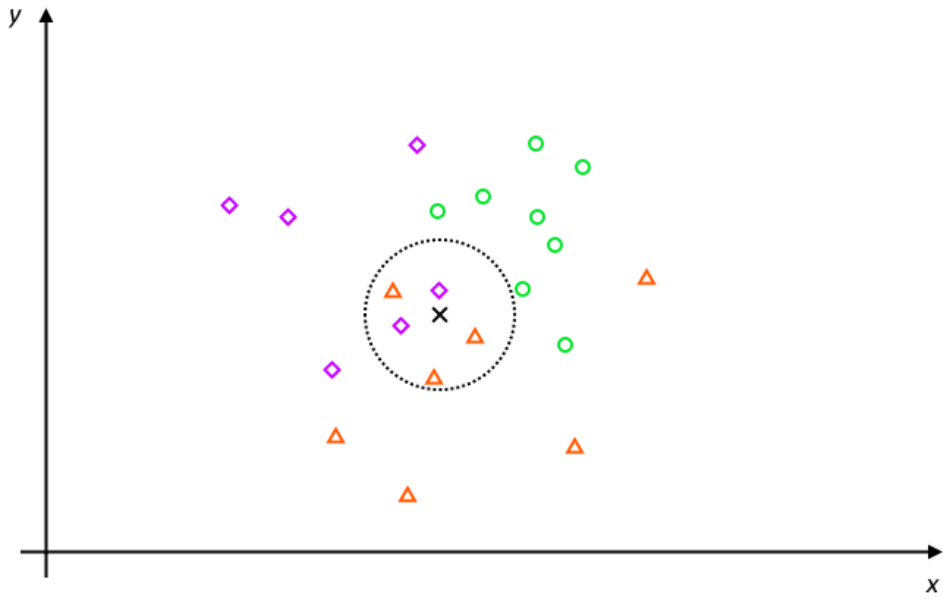

Figure 6. Illustration of the k-NN classification. The black X represents input data, and it's 5 nearest neighbors are located inside the dotted circle.

A k-NN classifier will return a classification label of the input data in addition to the distance, by comparing it to the labeled training data. This provides an intuitive way to categorize data, if knowledge about the classification category of the training data is available, and it becomes easy to construct a k-NN classifier even with no insight on the data and its content. For example, knowing whether some measured data sets are from a normal operating state of a system or from an abnormal, e.g. faulty system state, will enable the k-NN classifier to be trained. After training, the classifier can be used to discriminate new input data to either of the categories, revealing the systems operating state.

Unlike some methods of data identification and classification, there are no limits to the number of different classification categories for the k-NN method. This means that for signal analysis purposes, it is possible to construct a classifier system which differentiates

all distinguishable signal states from each other. However, the obvious prerequisite for the classification is that the signal data must correlate with the system states.

The k-NN algorithm can also be used for regression analysis, in which the input data is not classified, but instead the distance of the input data compared to the training data is measured and returned as a result. This distance can then be used to make a comparison between the similarity of the data sets or for other analysis. A k-NN classifier algorithm can also use inverse distance weighing to emphasize data which is more similar. In this case the k-NN classification is done by weighing the labels based on their distance from the input data, and choosing the label with the largest resulting weight number. More similar data will thus have a larger effect on the classification, improving the efficiency of the classification in some cases.

2.2.2 K-NN distance metrics

The k-Nearest Neighbors is a numerical method, and it requires a metric which is calculated for the input data and used to decide the distance, or similarity, to the training data. There are several substantially different ways to calculate the distance between data sets in k-NN algorithms, for example a few of the most common ones include Euclidean, cityblock, correlation, cosine, Hamming, Mahalanobis and Chebychev distance metrics.

In general, the k-NN does not require any specific type of distance algorithm, and any mathematical operation which returns a single value from a set of data can be used. The Euclidean and cityblock distances are common metrics for numerical data, and they are easy to grasp as they are closely tied to casual distances in 3-dimensional and 2-dimensional spaces. The Euclidean distance is defined as the shortest distance between two points, *a* and *b* in an n-dimensional Euclidean space:

$$d_{euclidean}(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \qquad (10)$$

And the cityblock distance, sometimes referred to as taxicab or Manhattan distance, is defined as the sum of absolute difference of coordinates:

$$d_{cityblock}(a,b) = \sum_{i=0}^{n} |a_i - b_i| \tag{11}$$

The cityblock distance resembles the distance required to move in a city with an orthogonal layout of streets, hence the name. In its general form it applies to all n-dimensional spaces, like the Euclidean distance.

Correlation, a statistical method for finding dependence in data, can also be applied to the k-NN distance calculations. There are many different types of correlation, but in the k-NN method perhaps the most useful type is linear correlation, which is used to find or test a linear statistical relationship between variables or sets of data. The most common definition for linear correlation is the Pearson product-moment correlation coefficient:

$$p(x,y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{12}$$

Where *cov(X,Y)* denotes the covariance of *X* and *Y*, and *E* is the expected value operator. μ and σ are the expected value and standard deviation, respectively. The Pearson's correlation returns values from -1 to 1, where values close to 1 mean there is a strong linear correlation between the variables, 0 means there is no correlation, and -1 means an inverse linear correlation.

The k-NN algorithm can return substantially different calculated distances for the input data depending on the distance metric used. The metric must be chosen carefully, and the results should be interpreted with knowledge on what the chosen distance represents. For example, the Hamming distance is commonly used in information technology for error detection and correction in texts, because it returns a natural number describing the

amount of individual differences, e.g. different characters, in the data. Chebychev distance is a useful metric for calculating the required operating time of machines that can move independently along several axes, like an overhead crane which moves into position in a logistic warehouse, and machine vibration signal analysis and classification has been tested to work well with the Euclidean and cityblock distances (Seshadrinath et al. 2013).

2.2.3 Dimensionality of data in the k-NN method

Increasing the dimensionality of the data sets can reduce the functionality of the k-NN algorithm, sometimes referred to as the curse of dimensionality, by reducing the separation between the nearest and farthest neighbors in the data set. How much this affects the k-NN distance depends on the qualities of the data sets, and in some high-dimensional cases does not cause problems as only irrelevant additional dimensions reduce the k-NN differentiation capability (Houle et. al. 2010). In general, increasing the amount of dimensions improves the differentiation capabilities up to a certain point, after which it starts to degrade the differentiation (Dougherty 2013). This is called the peaking phenomenon, shown in Figure 7.

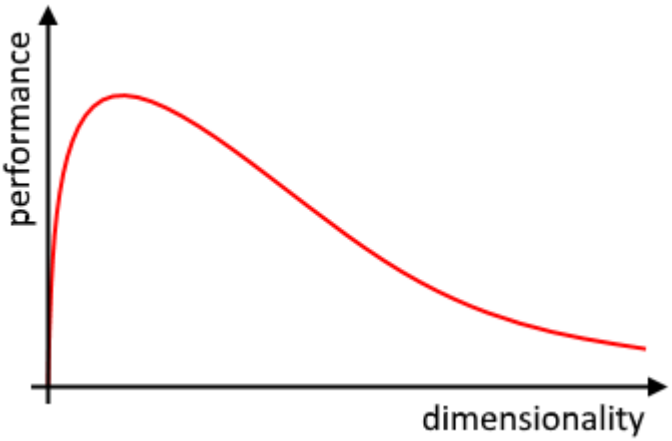

Figure 7. The peaking phenomenon in k-NN classification causes the classification performance to decrease after a certain number of dimensionality.

If k-NN classification problems arise from high-dimensionality, they can be circumvented by performing dimensionality reduction before the k-NN algorithm is used on the data. Dimensionality reduction consists mainly of two methods: feature selection and feature

extraction. The goal of feature selection is to find the data points that provide most information, and use them in the classification process, discarding the rest and thus reducing the dimensionality. Similarly, feature extraction is based on finding data points that are combinations of the original data, and using these data points in the classification. These dimensionality reduction methods can be implemented with many different methods, for example subset selection, principal component analysis (PCA) or linear discriminant analysis (LDA). (Dougherty 2013)

3. Signal analysis of sensor data

In order to discover details or events happening in an industrial process, a signal data recording of some relevant parameter must be obtained and analyzed. Such signals can be practically anything from temperature, sound, pressure or vibration to luminance, magnetic field strength or amount of electromagnetic interference in a circuit. Traditional sensor equipment for manufacturing industries have relied on measuring mechanical quantities like pressure, force, power, or flow rate, but the modern IoT devices and sensor equipment bring forth numerous possibilities of measuring more specific and complex quantities.

As the measured quantities and processes from which they are measured become more complex, they become more difficult to analyze intuitively. This means that only looking at the signal itself will not suffice to gather or uncover meaningful information, and more advanced mathematical methods must be used to gain knowledge on the signal. Another problem is that the increased ease of data collection and number of signals to be recorded and analyzed results in massive amounts of data, which might not contain any relevant information. Small changes become easily masked by the big data, which implies that more efficient and sensitive methods of signal and data analysis need to be applied.

Different applications and methods of signal analysis in industrial and commercial processes have been the subject of much academic research. For example, prominent methods for detecting bearing and rotor failures or rotor eccentricity of electric motors and bearing and gear failure of gearboxes include motor stator current and torque analysis (Blödt et al. 2006) as well as vibration and acoustic analysis (Kia et al. 2012). These methods focus on detecting deviations in the measured signal, which are assumed to correlate with mechanical failures, and they utilize varying mathematical analysis tools from frequency analysis to pattern recognition. The wavelet transform and k-NN method are fine examples of such analysis tools, and they provide many possibilities to develop a signal processing algorithm for sensor data analysis.

3.1 Analysis algorithm composition

An algorithm which uses the wavelet transform and k-NN method to perform regression and classification analysis on signal data is described in this section. The algorithm can be implemented in any suitable environment which allows the use of the wavelet transform and k-NN methods, and can be used for practically any cyclic or periodic sensor data. A diagram illustrating the signal analysis process is shown in figure 8.

The algorithm input is a measured signal from an observed system, which is sampled to a discrete form. If the signal's sample length or measurement timing is variable, it is cut and resampled to a pre-determined sample length and timing to provide precise and comparable measurements. The sampled signal is then converted with the wavelet transform. Both CWT or DWT can be used, but as the DWT is time-shift sensitive, it requires signal data which is localized precisely in the time-domain. The CWT allows a slightly more flexible approach, because time-shifted signals will simply yield similar coefficients which are also time-shifted, instead of coefficients with different amplitude.

When used with finite length data, edge effects can affect the coefficients and signal representation accuracy produced by the WT. These effects are caused by the discontinuities in the beginning and end of the signal. Different ways to mitigate this problem include circular convolution, signal reflection and wavelet extrapolation, which can help reduce the edge effects to an acceptable level (Williams & Amaratunga 1997).

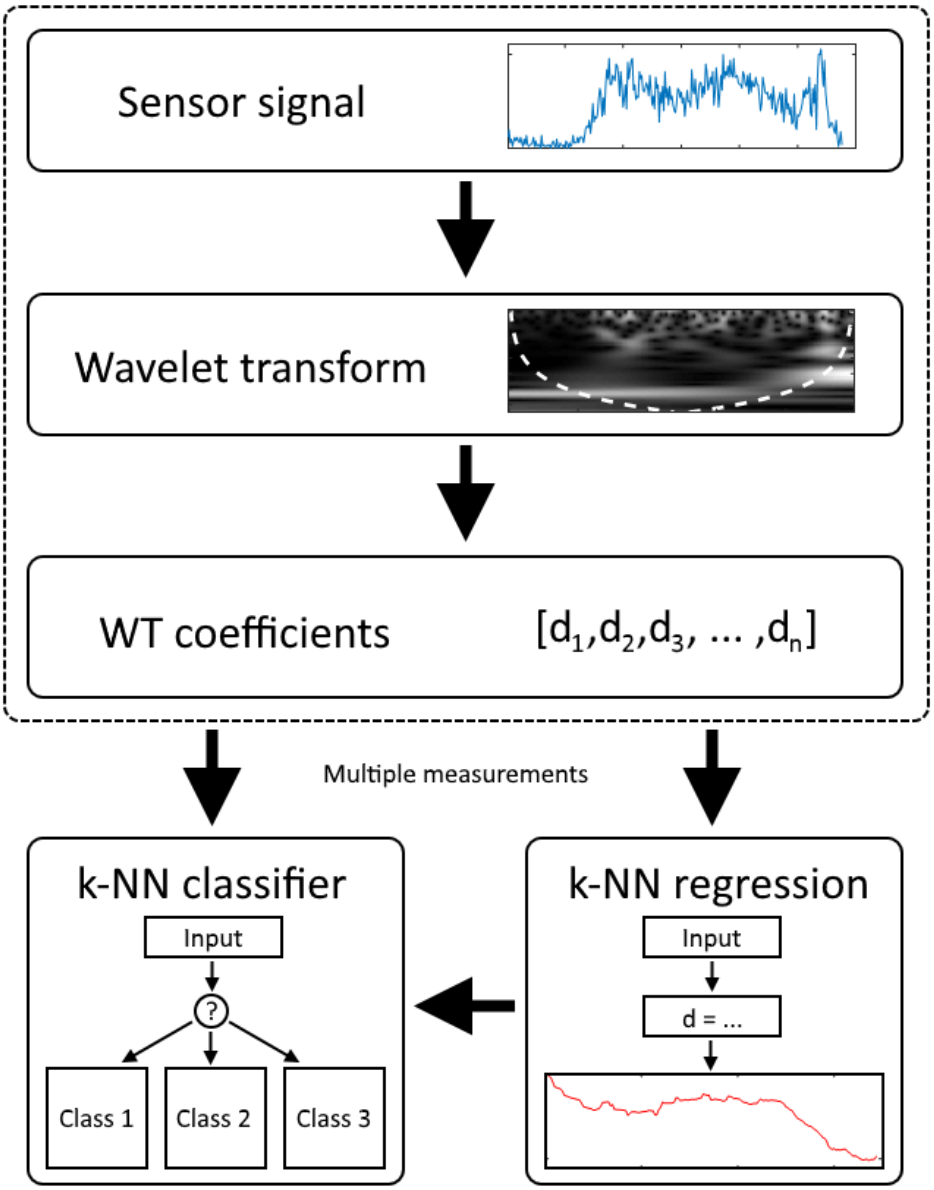

Figure 8. Process diagram of the signal analysis algorithm

The WT forms a time-frequency representation of a signal, which makes it possible to detect transients and abnormal frequencies. In this form the signal can also be easily manipulated to include or exclude certain frequencies, similar to filtering with low-, high- or bandpass filters, but the filtering requires only one step instead of possibly several consecutive filters. Noise or other irrelevant frequencies can be cut out from the signal, and the signal can be split to several parts if needed.

The second step of the algorithm is the regression and classification analysis, which requires multiple individual measurements to be recorded and processed as described in the previous paragraphs. The WT coefficients are used to build a database of sampled signals, which allows to construct a k-NN classifier and perform regression analysis on the signal. If needed, dimensionality reduction can be applied before the k-NN method to speed up computing and improve the algorithm effectiveness by reducing redundant dimensions of the data.

Regression analysis provides a tool to detect changes in the signal by comparing the latest signal measurement with the earlier measurements from the database. When the content of the signal, for example, a dominant frequency changes or attenuates, it changes the WT coefficients, which in turn increases the regression distance. The results of the regression analysis can be used in building the k-NN classifier, by labeling the data to different classes based on the regression distance value. Also, if the system state is known for the signal source, the k-NN classifier can be constructed directly without the regression analysis. After training the k-NN classifier, it is then used to label new signal data directly, providing a highly automated way of classifying input data.

4. Tests on traffic loop sensor data

This chapter describes the test results obtained by applying the signal analysis algorithm discussed in the previous chapter to sensor data from a highway ramp loop sensor (PeMS 2006). The sensor measures the number of passing cars with a five-minute time resolution. The data set contains measurements from 25 weeks, or 175 days, with a total of 50400 samples. There are 2903 missing samples scattered throughout the data set, represented by a value of -1, from times when the sensor has been offline.

For signal analysis purposes, the data set is divided into daily signal measurements with 288 samples per day. This provides a suitable signal length for the time-frequency analysis and allows an intuitive comparison of individual days. The measurements are organized by day, starting from those measured on Mondays and ending on those measured on Sundays in order to provide a basis for the regression analysis. An example of the sensor data measured on a Monday is shown in figure 9.

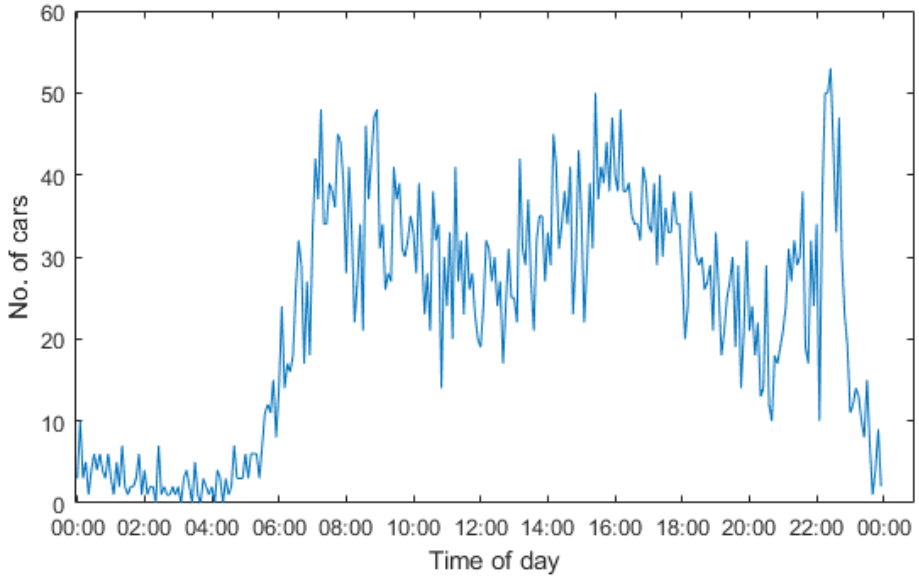

Figure 9. Number of cars driving through a highway ramp during a Monday.

The objective of the first test in section 3.1 is to see how well the algorithm can indicate changes in the signal data, which are caused by changes in the traffic patterns on weekends. The results of the regression analysis could be then used to construct a k-NN classifier for classification of subsequent data in a more developed algorithm. Section 3.2

examines how changing the algorithm's key parameters affects the results and efficiency of the regression analysis. All numerical processing is done using the MATLAB R2016b software with signal analysis and machine learning toolboxes.

4.1 Detection of weekends with k-NN regression

The algorithm calculates the k-NN distance for the measurements and returns it as an output variable. This variable indicates the similarity of the input data, or latest measurement, compared to the earlier measurements used as training data. Measurements from Mondays and Tuesdays are used as training data, and the measurements from the rest of the week are used as input data and compared to the training data one measurement at a time, and the distance of each measurement is plotted to a graph. As the measurements are 5-minute aggregates, the actual frequencies of events in the data are in the micro- and millihertz range. The k-NN regression distance is calculated with the Cityblock distance metric, and the mean of the 10 nearest neighbors' distances is shown for each measurement in figure 10.

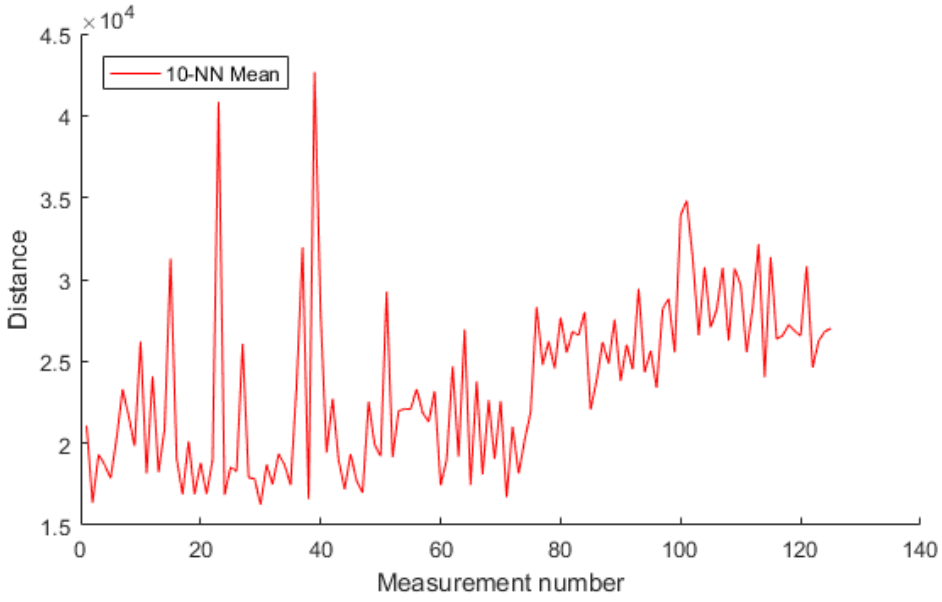

Figure 10. Mean 10-nearest neighbor distance of weekdays from Wednesday to Sunday compared to the training data from Mondays and Tuesdays.

The data contains obvious outliers, mainly caused by missing sensor data, which the algorithm does not interpolate or discard. Some of the smaller outliers are caused by

other events that affect the traffic patterns, for example national holidays and special events near the loop sensor ramp. Measurements numbered from 0 to 75 are from weekdays from Wednesday to Friday, and numbers 76 to 125 are from Saturday and Sunday. The mean distance for the weekdays is ca. 0,77 times the mean distance for Saturdays and Sundays. Thus, the distance is significantly larger for the weekend measurements, which means that the regression part of the algorithm can be used to differentiate weekends and/or other special events from regular weekdays. Separating weekends from special events happening on weekdays is not examined in this thesis, however, for example suitable frequency filters in the wavelet transform could be used to achieve a division between these special cases.

4.2 Effect of algorithm parameters

Filtering certain frequencies out of the data used in k-NN calculations, using a larger or smaller number of nearest neighbors, or different k-NN distance metric will change the result of the algorithm. The parameters should be chosen depending on the input data and the objective of the analysis, especially if the initial results are not as expected.

As the objective of the algorithm is to discern measurements from weekends from those of weekdays, the ratio of average regression distance between the measurements on weekdays and weekends can be thought of as a crude indicator of the algorithm efficiency. However, it does not take into account the deviation of individual values, e.g. how the algorithm handles data outliers, and thus it cannot be used as the only performance indicator. The results of individual measurements should be studied to examine the algorithm efficiency thoroughly.

4.2.1 Frequency filtering

Filtering out frequency bands can reduce noise and other disturbances in the data. In these tests, the MATLAB CWT function produces 67 scales of coefficients for a signal length of 288 sample points, each scale corresponding to different frequencies in the signal. Filtering has a definite effect on the regression distance, shown in figure 11, with 0, 20, and 40 scales corresponding to the highest frequency components of the wavelet transform excluded

from the k-NN regression, effectively low-pass filtering the signal. The regression distance is calculated with the cityblock distance metric and the result is the mean of the 10 nearest neighbors' distance. The results in figure 11 have been normalized by dividing each respective curve with its mean value to make the figures comparable, as reducing the dimensionality of the input data reduces the total distance calculated in the k-NN regression.

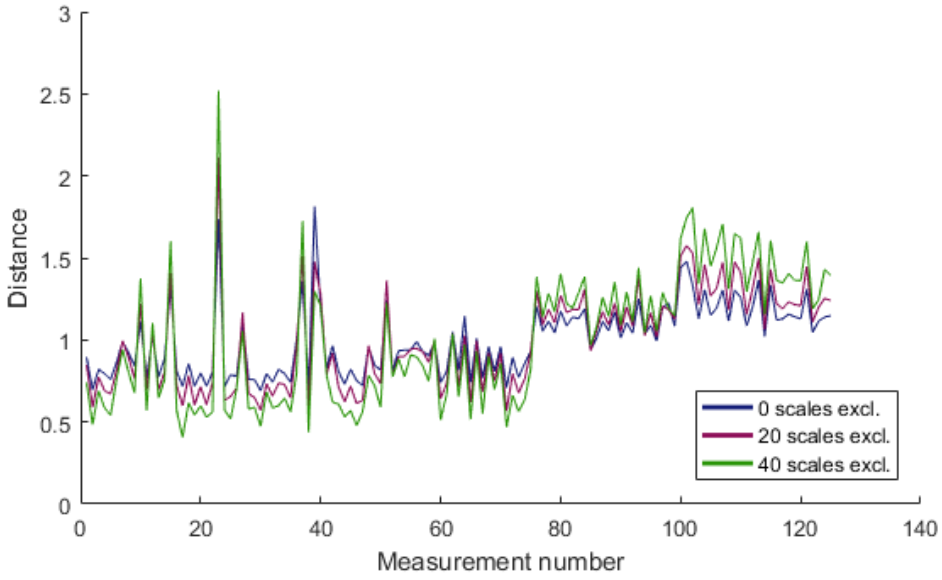

Figure 11. Effect of frequency filtering of the signal prior to k-NN regression.

High-pass filtering improves the regression analysis between the weekends and weekdays, but it also makes some of the outlier measurements stand out, which might not be a wanted effect depending on the case. In this case the outliers are not of special concern, and filtering out the high frequencies improves the total results, as the regression distance ratios improve substantially up to a certain point. The distance ratios achieved by filtering the high frequencies are shown in table 1.

Table 1. k-NN regression distance ratio of weekdays and weekends with respect to WT scales excluded.

Scales excluded	k-NN distance ratio	Scales excluded	k-NN distance ratio
0	0,779	35	0,620
5	0,760	40	0,587
10	0,739	45	0,533
15	0,717	50	0,478
20	0,695	55	0,474
25	0,672	60	0,499
30	0,645		

In practice, the filter bands can be chosen to observe specific frequencies of interest, making the algorithm more effective with data that contains noise or other non-informative frequencies.

4.2.2 Number of nearest neighbors

Changing the number of nearest neighbors to be evaluated with the k-NN regression is another key parameter of the algorithm. The regression distance is zero for identical measurements, which means that in case of erroneous data, for example receiving the same measurement once to both the training data set as well as the input data, the regression algorithm will return a distance of zero if only one nearest neighbor is used. The larger the number of nearest neighbors to be used in the comparison, the less chance there is for outliers and errors to affect the results significantly. On the other hand, too many nearest neighbors can make it more difficult to detect changes and categorize the data, especially if the data contains outliers which might begin to affect the results as the number of neighbors increases. The effect of changing the number of nearest neighbors on the regression distance is shown in figure 12 for 1, 10, and 20 nearest neighbors. Distance measure used in the regression is cityblock, and no WT scales are excluded.

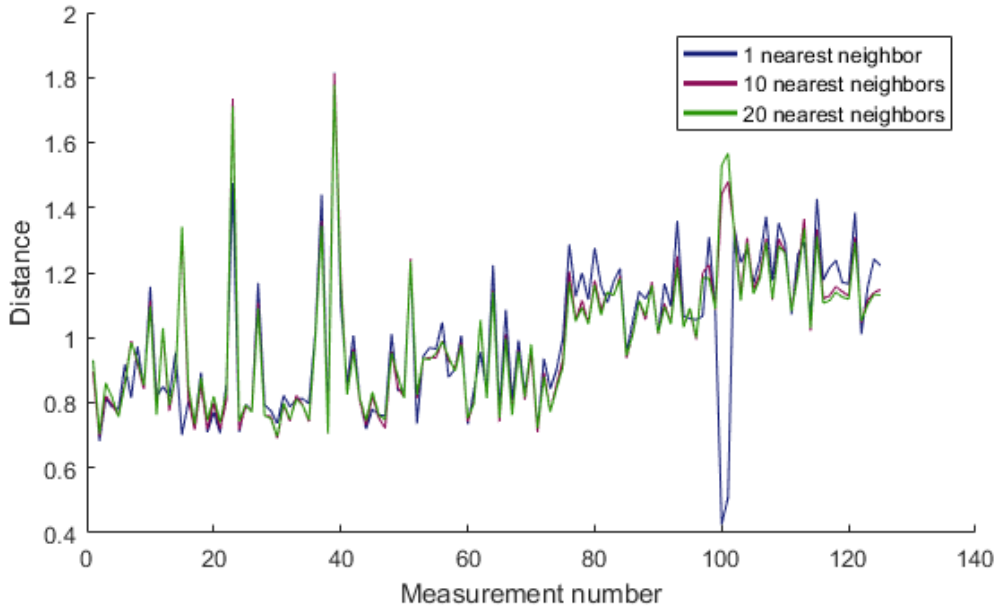

Figure 12. Effect of the number of nearest neighbors in k-NN regression

In this case the number of nearest neighbors does not affect the results much, except when the number is very small. The average regression distance ratios for 1, 10 and 20 nearest neighbors are 0,775, 0,772 and 0,779 respectively. However, a small number of nearest neighbors causes the regression distance to decrease dramatically for measurements for which there exists a few almost identical measurements, even if they are completely dissimilar to all other measurements, as shown by the samples 100 and 101 in figure 12. Increasing the number of neighbors makes the regression more reliable, especially with imperfect data such as the highway ramp sensor data.

4.2.3 K-NN distance metric

Changing the k-NN distance metric will affect how the distance between the data points is calculated. It has a substantial impact on the algorithm results, and metrics which return continuous values, such as the cityblock, Euclidean and correlation, are most useful in the analysis of sensor signal data. The regression distances for the sensor data using the three aforementioned metrics are shown in figure 13, with the mean of 10 nearest neighbors plotted with no WT scales excluded.

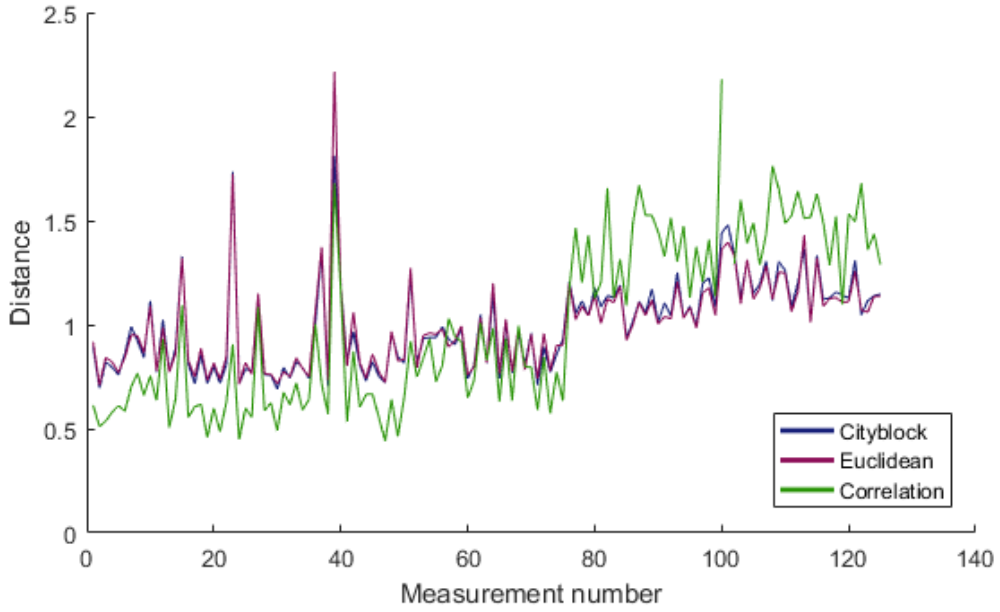

Figure 13. Effect of k-NN distance metrics on the algorithm output.

The Euclidean and cityblock provide very similar distances, with average distance ratios of 0,801 and 0,772 respectively. Correlation distance metric provides best division between weekdays and weekends with an average ratio of 0,506, but does not return a distance value for measurement 101 because the numerical variation in the input data is too small. This indicates a measurement with missing or erroneous samples, possibly measured on a day when the loop sensor has been offline and thus the data does not contain enough variance to calculate the correlation distance with. The cityblock and Euclidean distances do not exhibit this problem, as they are essentially sums of values, and would work on data with zero variance.

5. Discussion

The algorithm developed in this thesis indicates good capabilities of functioning as a highly automated signal analysis tool. The specific computational requirements were not studied, but if the algorithm requires too intensive calculations, the computational complexity of the algorithm can be managed by reducing the input signal data size e.g. by downsampling the signal, reducing the number of wavelet coefficients or applying dimensionality reduction on the input data before the k-NN regression or classification. There are also many possibilities of making the algorithm more effective at detecting signal variations at the cost of more processing required. Some of these possibilities are mentioned in this work, but are not studied as they are beyond the scope of this thesis.

There are several parameters which affect the outcome of the algorithm, and care must be taken to adjust these parameters depending on what kind of data is processed and what is the preferred outcome of the calculation. Binary input data might benefit from the use of a completely different distance metric, for example Hamming distance, instead of the metrics examined in this thesis. The use of k-NN classification to label input data directly into categories was not tested with the highway ramp sensor data, but it is one of the core elements in the algorithm's functionality, and should be studied in a future work, possibly with data that can be labeled to several categories to test how well different categories are discerned with the algorithm. The classification capabilities could also be improved with the use of more advanced machine learning algorithms, such as neural networks, support vector machines or decision trees.

Conclusions

Signal data analysis can be performed with many different mathematical tools, and in this thesis the use of the wavelet transform and k-Nearest Neighbors method in the analysis of sensor signal data was studied. These methods were chosen for their suitability in signal analysis and condition monitoring, based on research literature on the subjects. They are also suitable for a wide variety of data types with correct parameter selection. The wavelet transform provides a time-frequency representation of a signal, which opens up possibilities to separate interesting events from each other or unwanted noise, and the k-NN algorithm is a light machine-learning method which can be used for both regression and classification of practically any data.

The combination of these two methods to create a signal analysis algorithm was examined, and a prototype algorithm was constructed to test the capabilities of the chosen methods. With test data from a highway traffic loop sensor, the algorithm was used to successfully separate measurements made on weekends from those made on weekdays by utilizing a continuous wavelet transform and k-NN regression analysis. The partially bad quality of the data, with some missing measurement values and other uncontrolled events, generated some minor inaccuracies. The results of the algorithm could be improved with some changes and additions to the algorithm, or fine-tuning the key parameters of the wavelet transform and k-NN method. Changing the key parameters was examined in the test chapter, and choosing suitable values provided notable improvements on the algorithm results.

References

(Blödt et al. 2006) Blödt, M., Regnier, J., Chabert, M., Faucher, J. 2006. Fault Indicators for Stator Current Based Detection of Torque Oscillations in Induction Motors at Variable Speed Using Time-Frequency Analysis. [Online document]. Available at http://ieeexplore.ieee.org/document/1664709/

(Boldt & Ribeiro 2014) Boldt, F. d. A., Ribeiro, M. P. 2014. Performance Analysis of Extreme Learning Machine for Automatic Diagnosis of Electrical Submersible Pump Conditions. [Online document]. Available at http://ieeexplore.ieee.org/document/6945485/

(Bouguerne et al. 2011) Bouguerne, A., Lebaroud, A., Medoued, A., Boukadoum, A. 2011. Classification of Induction Machine Faults by K-Nearest Neighbor. [Online document]. Available at http://ieeexplore.ieee.org/document/6140191/

(Chui 1992) Chui, C. K. 1992. An Introduction to Wavelets. Texas A&M University. Academic Press.

(Deng et al. 2015) Deng, X., Luo, R., Li, J. 2015. Similarity Matching Algorithm of Equipment Fault Diagnosis Based on CBR. [Online document]. Available at http://ieeexplore.ieee.org/document/7339222/

(Dougherty 2013) Dougherty, G. 2013. Pattern Recognition and Classification. [Online document]. Springer. Available at: http://link.springer.com.ezproxy.cc.lut.fi/book/10.1007/978-1-4614-5323-9/page/1

(Houle et. al. 2010) Houle, M. E., Kriegel, H. P., Kröger, P., Schubert, E. & Zimek, A. 2010. Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? Springer.

(Hös & Baszó 2014) Hös, C., Baszó, C. 2014. Volumetric Pumps and Compressors. [Online document]. Budapest University of Technology and Economics. Available at http://www.hds.bme.hu/mota/eng/vpc/vpc.pdf

(Kia et al. 2012) Kia, S. H., Henao, H., Capolino, G.-A. 2012. A comparative study of acoustic, vibration and stator current signatures for gear tooth fault diagnosis. [Online document]. Available at http://ieeexplore.ieee.org/document/6350079/

(MacGillivray & Turner 2015)

MacGillivray, C., Turner, V. 2015. Worldwide Internet of Things Forecast, 2015-2020. [Online document]. Available at http://www.idc.com/getdoc.jsp?containerId=256397

(Mallat 2009) Mallat, S. 2009. A Wavelet Tour of Signal Processing. Elsevier/Academic Press.

(McKinsey 2015) McKinsey Global Institute. 2015. The Internet of Things: Mapping the Value Beyond the Hype. [Online document] Available at http://www.mckinsey.com/business-functions/business-technology/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world

(Mehra 2007) Mehra, D. 2007. Positive Displacement Pumps Market. [Online document]. Available at http://www.pumpsandsystems.com/topics/pumps/pumps/positive-displacement-pumps-market

(PeMS 2006) University of California, Berkeley. Highway Performance Measurement System. [Online document]. Available at http://archive.ics.uci.edu/ml/machine-learning-databases/event-detection/

(Rao & Bopardikar 1998) Rao, R. M., Bopardikar, A. S. 1998. Wavelet Transforms. Introduction to Theory and Applications. Addison-Wesley.

(Seshadrinath et al. 2013) Seshadrinath, J., Singh, B., Panigrahi, B. K. 2013. Investigation of Vibration Signatures for Multiple Fault Diagnosis in Variable Frequency Drives Using Complex Wavelets. [Online document]. Available at http://ieeexplore.ieee.org/document/6497640/

(Weeks 2006) Weeks, M. 2006. Digital Signal Processing Using Matlab and Wavelets. Jones & Bartlett Learning.

(Williams & Amaratunga 1997)

Williams, J. R., Amaratunga, K. 1997. A Discrete Wavelet Transform without edge effects using wavelet extrapolation. [Online document]. Available at: http://link.springer.com/article/10.1007/BF02649105

(Wuest et al. 2016) Wuest, T., Weimer, D., Irgens, C., Thoben, K.-D. 2016. Machine learning in manufacturing: advantages, challenges, and applications. [Online document]. Available at http://dx.doi.org/10.1080/21693277.2016.1192517