DEGREE PROJECT, IN SYSTEMS ENGINEERING , SECOND LEVEL
STOCKHOLM, *SWEDEN 2015*

Anomaly Detection in Diagnostics Data with Natural Fluctuations

JESPER SUNDBERG

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCI SCHOOL OF ENGINEERING SCIENCES

Anomaly Detection in Diagnostics Data with Natural Fluctuations

J E S P E R S U N D B E R G

Master's Thesis in Systems Engineering (30 ECTS credits)
Master Programme in Mathematics (120 credits)
Royal Institute of Technology year 2015
Supervisor at KTH was Per Engvist
Examiner was Per Engvist

TRITA-MAT-E 2015:46
ISRN-KTH/MAT/E--15/46--SE

Royal Institute of Technology
School of Engineering Sciences

KTH SCI
SE-100 44 Stockholm, Sweden

URL: www.kth.se/sci

Abstract

In this thesis, the red hot topic anomaly detection is studied, which is a subtopic in machine learning. The company, Procera Networks, supports several broadband companies with IT-solutions and would like to detected errors in these systems automatically. This thesis investigates and devises methods and algorithms for detecting interesting events in diagnostics data. Events of interest include: short-term deviations (a deviating point), long-term deviations (a distinct trend) and other unexpected deviations. Three models are analyzed, namely Linear Predictive Coding, Sparse Linear Prediction and Wavelet Transformation. The final outcome is determined by the gap to certain thresholds. These thresholds are customized to fit the model as well as possible.

Keywords: machine learning, anomaly detection, fault detection, Linear Predictive Coding, Sparse Linear Prediction, Wavelet Transformation

Sammanfattning

I den här rapporten kommer det glödheta området anomalidetektering studeras, vilket tillhör ämnet Machine Learning. Företaget där arbetet utfördes på heter Procera Networks och jobbar med IT-lösningar inom bredband till andra företag. Procera önskar att kunna upptäcka fel hos kunderna i dessa system automatiskt. I det här projektet kommer olika metoder för att hitta intressanta företeelser i datatraffiken att genomföras och forskas kring. De mest intressanta företeelserna är framfärallt snabba avvikelser (avvikande punkt) och färändringar äver tid (trender) men också andra oväntade mänster. Tre modeller har analyserats, nämligen Linear Predictive Coding, Sparse Linear Prediction och Wavelet Transform. Det slutgiltiga resultatet från modellerna är grundat på en speciell träskel som är skapad fär att ge ett så bra resultat som mäjligt till den undersäkta modellen.

Nyckelord: maskinlärning, anomalidetektering, feldetektering, Linear Predictive Coding, Sparse Linear Prediction, Wavelet-transformation

Acknowledgements

I would like to express my gratitude and sincere appreciation to Anders Waldenborg for all the help I got to accomplish this project and also for all the support I got in Python and the subject anomaly detection.

I am very grateful to Per Enqvist, lecturer, at the department of mathematics at KTH and would like to give my thanks for sharing his expertise and for vital guidance that I received throughout this process.

I want to thank Sandra Dahlgren and the entire Procera Network for having faith in me and giving me the opportunity to do this project.

I also want to thank the excellent writer Jonathan Hörnhagen for linguistic and grammatical improvements.

Abbreviations and Variables

iff	if and only if
e.g.	exemplī grātiā ("for example")
i.e.	id est ("that is")
db	Daubechies (a member of the wavelet family)
ECG	Electrocardiography
D	Input data set
D_{ref}	Reference data set
D_{test}	Test data set
d_i	Elements in D
k	Number of elements in D
i	Used as index for the data D, often: 1, 2, ..., k
Y	Output signal from the models
y_i	Elements in Y
N	Order of the model
j	Used in iterations for the order N, often: 1, 2, ..., N
C	Coefficient vector used in LPC, often: [1 c_2 ... c_{N+1}]
c_i	Elements in C
A	Coefficient vector used in LPC, often: [α_1 α_2 ... α_N]
α_i	Elements in A
W	Output signal from the moving window sum
w_i	Elements in W
AS	The anomaly score
CWT	Continuous wavelet transform
THV	Threshold value
THI	Threshold indicator

Contents

List of Figures

Chapter 1

Introduction

The first part of this chapter gives a brief explanation of the concept anomaly detection and some fundamental areas in mathematics, followed by some words about Procera Networks and a discussion about the main task of this paper.

1.1 What Is Anomaly Detection?

In data mining, anomaly detection is an automatic test that, in a data set, is to find data that in some sense does not fit to the rest of the data, i.e. is anomalous. The nature of an anomaly is entirely defined by the context, for the versatility of the data set can be very high. Basically, anomaly detection can be translated into one question; is it normal or not? The answer is always that the observed data is either normal or anomalous (depending on the context however, it is also possible to get something in between those two). Anomaly detection as a subject is highly related to fault detection and some might say that it is the same thing. For ease of understanding, two examples are examined, see Figure 1.1 and Table 1.1.

Both these two scenarios show similarities in the sense that they involve time dependent sequences of data points. They are different in other senses,

Figure 1.1: A real-valued discrete time series with a subsequence in the middle that shows tendency of being anomalous.

such as the type of data points which are real-valued and symbolic respectively. The structure of the data set is also different, one long sequence contra several sequences. In the sense of anomaly detection, the nature of the anomalies is what differs the most, for in the first example, it is a matter of finding what data points are anomalous, whilst in the second example, it is a matter of finding which sequence are anomalous compared to the others.

Consider the scenario in Figure 1.1, this could be interpreted as a factory machine producing items with corresponding measurements, after some time the machine encountered a problem and did not produce as consistent as it should (anomaly). However, this problem seems to be solved after some time and the machine continued in its normal behaviour.

U_1	login	passwd	messages	pictures	...	homepage	logout
U_2	login	passwd	pictures	homepage	...	homepage	logout
U_3	login	passwd	messages	messages	...	pictures	logout
U_4	login	passwd	homepage	pictures	...	page	logout
U_5	login	passwd	login	passwd	...	login	passwd

Table 1.1: User one through five at the left hand side of the table and the corresponding user commands at the right hand side. The last user shows tendency of being anomalous

An interpretation could also be made for the scenario in Table 1.1, namely, an intruder protection for a website with login requirements. All IP-adresses visiting the site seems fine except for the last that could be a bot trying to hack someones account (anomaly).

Anomaly detection is not only applicable in digital areas, it is also useful when saving lives. For it is used in ECG devices which records the signal from the heart beat rate and when it detects an anomaly in the signal the ECG immediately activates.

All this is a subtopic of machine learning[7].

1.2 Background

In southern Sweden, namely Malmö, a medium sized company (approximately 200 employees) is located[1]. They are called Procera Networks Inc and are working with IT-infrastructure improvements and customer insight care. The headquarter is located in Los Angeles, CA, USA. Particularly, Procera is managing Internet data traffic for their customer which uses Internet from different sources, e.g. programs (computer) or applications (smart phones). A collection name for these sources will throughout this report be 'signature' (e.g. Skype). All data being transferred at the customers of Procera are being stored in a database. This database is updated every hour. Every customer uses Internet individually from different signatures and in different amounts, therefore there is a value stored for every signature for every hour. There are over 20,000 signa-

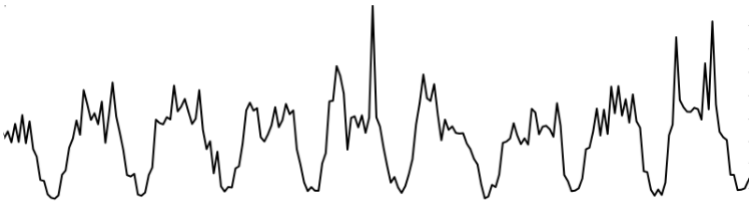

Figure 1.2: A typical time series, representing data traffic for nine days. A periodical property is visible.

tures. This results in a huge amount of data in many discrete time series. For easier understanding, an example of an ordinary time series is shown in Figure 1.2.

1.3 Objective

The aim of the thesis is to automatically detect anomalies with as high accuracy as possible. This is divided into two parts.

The first part is to build a number of suitable algorithms and then evaluate these to determine which one is the most optimal in this context.

The second part is to specialize in the final algorithm, i.e. to adjust all the parameters of that algorithm so that it will be suitable for its particular purpose. These adjustments could of course be done in advance. But the value of this particular assortment is that low performing algorithms can be rejected on the spot. Otherwise, the algorithms that are close in terms of the outcome must then be evaluated further (in detail).

1.4 Purpose

Ideally, the decisive anomaly detection framework is integrated at Procera Network with ease, at their Python-based database. The reason why Procera wants to use anomaly detection is to automatically get an overview of all the signatures at once and see if any of them are behaving improperly. In case of misbehaviour, a indication would be sent notifying that something is wrong. Some explanations for this sudden change of behaviour could be, e.g. due to maintenance errors, user overload or software updates not being compatible with a particular signature. Whatever reason there is behind the change can be argued to be of less interest in this context, but knowing about it is very valuable. Procera would then be able to put themselves one step ahead, working closer as well as faster with the customer. Also it creates the possibility to take care of the problem before a user eventually do. A successful accomplishment of the objective will facilitate the manual work at Procera tremendously.

1.5 Problem Discussion

The constant struggle with data of this user-based category is that they all behave differently. Signatures are used in different ways in different amounts at different times of the day. In the data, searching for specific patterns being connected with anomalies is not enough. This is because all problems causing anomalies have a possibility to occur. Some of these have never happened before, or are still unknown, therefore they are not sought for because they cannot be found in historical data.

Some signatures are used in the same way, e.g. two games, but one of the games has a much larger player base. This results in similar curves when only considering shape, but very different when considering size. The curve for one game is a rescaled version of the other. This should, therefore, not be treated as an anomaly.

On the other hand, some signatures are not used in the same way, e.g. a web-browser and a torrent client[1]. The browser transfers most of its data a short time after every user command while the torrent client needs only a few commands to achieve a large activity for a long period of time. Since the torrent client is not user-dependent it has the opportunity to behave more linear than the average behaviour, i.e. show less periodic property. This too should not be treated as an anomaly.

The conclusion of this is that the nature amongst the signatures differs, thus, they must be treated differently. An anomaly in one signature can be normal in another. Consequently, another task arises, namely, what should be considered as an anomaly. This is discussed in detail in Chapter 3

1.6 Problem Specification

Specifications are given by Procera and exists to keep the customers satisfied.

1. Anomalies are to be detected at least within a week. The reason for this is the greater risk of it being detected by the customer.
2. The output of the algorithm is a score which is related to each signature.
3. The anomaly detection algorithm is implemented in Python.
4. The algorithm needs to be faster than 60 minutes, for a new calculation will begin at this time.

[1] A torrent client enables the user to join a BitTorrent file distribution system which is a system that helps participants (all aiming for the same file) in the system to find each other and form efficient distribution groups, called swarms. In this swarm, every participant receive partitions of the desired file from several other participants simultaneously. The benefit is that there are several sources instead of one, as in peer-to-peer file sharing (P2P).

1.7 Research Questions

R1. What method is most suitable for detecting anomalies?

R2. What kind of deviations should be considered as anomalies?

R3. Which parameters are best to maintain and which to disregard?

R4. How will the outcome be scored?

R5. What level of certainty does the models provide?

Chapter 2

Literature Review

The literature review will provide the reader an understanding of necessary concepts and fundamental mathematics. The context of this chapter is also a foundation making it possible to answer the research question.

2.1 Supervision

In machine learning, there are primarily three subcategories of the topic anomaly detection, named and described below.

2.1.1 Supervised Learning

When a reference scheme is given for both the normal behaviour and the anomalous behaviour, the environment is said to be supervised. The anomaly detection task then comes down to a comparison whether the observed data is most likely to be more similar to the normal -or to the anomalous reference scheme.

Definition 2.1. Given data set D, normal reference data set(s) N and anomaly reference data set(s) A, *supervised machine learning* is defined to determine whether $d_i \subseteq D$ are more similar by nature to N or to A.

Benefits with this type of supervision is that it is kind of straight forward to solve and no further complications can occur. The disadvantage is that these reference schemes could be troublesome to receive, in some cases even impossible.

2.1.2 Unsupervised Learning

If there are no references at all to evaluate the observed data with, it is said to be unsupervised, therefore, the machine has to learn itself what states should be considered as normal and what states should be considered as anomalous.

Definition 2.2. Given data set D only and no reference set, *unsupervised machine learning* is defined to determine whether $d_i \subseteq D$ are more likely to be normal or anomalous in contrast to D, using D itself as reference.

The benefits with this type is that it can be applied on any data set and no groundwork on the data is required. The disadvantages is that the output can always be questioned, is it really anomalous or is it just a coincidence? What if the data that is considered anomalous really is normal and the opposite for the normal data. There is for the machine no correct answer to that, this has to be evaluated afterwards by professionals.

2.1.3 Semi-supervised Learning

As expected, semi-supervised is a in between of the two previous types. It is when one reference scheme is available whilst the other scheme is not. There are two cases where this can occur. Either there is a reference data set for the normal state and no reference for the anomalous set, or vice versa (reference for anomaly and no reference for normal state).

Definition 2.3. Given data set D and normal reference data set(s) N, *semi-supervised machine learning with normal reference* is defined as to determine whether $d_i \subseteq D$ are, by nature, "similar enough" to N. The term similar enough means that it should be within some threshold that is preferably chosen in an explainable way.

This is probably the most common type for it is often the case that there exists a desired normal state, which is known, and all other outcomes are classified as anomalies. These anomalies are, at the start of the process, unknown. An example for this could be a factory machine producing items with a certain measure with an additional threshold, all items measured outside this threshold are considered anomalous. The reason for an item to be measured abnormal, could be anything, it could be too hot in the factory, wear on the machine tools or a software that is out of date. Benefits and disadvantages with this type are a combination of the benefits and disadvantages from supervised and unsupervised learning.

The opposite semi-supervised type, where anomalies are sought for, is more common when there are several normal possibilities but only a few number of anomalous states.

Definition 2.4. Given data set D and anomalous reference data set(s) A, *semi-supervised machine learning with anomalous reference* is defined as to determine whether $d_i \subseteq D$ are, by nature, "similar enough" to A. The term similar enough means that it should be within some threshold that is preferably chosen in an explainable way.

An example of this type is the intruder example in Table 1.1 where every user is classified as normal except those who try to log in over and over but still fail.

Furthermore, in computer science, anomaly detection can be classified into two other subcategories, that are independent of the supervision, named and explained below.

- **Online method** is an algorithm that is processing data in a time serial fashion, i.e. it is fed input data successively and solves the problem in every step. This type of method is applied successfully on systems that are currently up and running (i.e. online).
- **Offline method** is an algorithm that requires the full data set when it begins. If that requirement is fulfilled it produces an output answer from that data right then and there. No further actions are made in offline methods.

2.2 Norm

Norm is a mathematical term, defined as the length or size of a vector (in a vector space) or a matrix. It is written out as $||X||$, where X is the vector that is to be normed. If the norm of several vectors is sought, the lengths are summarized[2]. The power of the norm is often mentioned before the norm, e.g. L2-norm. The general definition of the $L_n - norm$ is:

Definition 2.5. $||X||_n = \sqrt[n]{\sum_i x_i^n}$

Even though the formula is very clear and every $L_n - norm$ looks much like other norms, the value of n becomes very important, for the properties are very different for different n's and their applications differ too. The most common norms will be discussed briefly in subsections 2.2.1 and 2.2.2.

2.2.1 L2-norm

The most common of all norms is the L2-norm or the Euclidean norm. It is obtained when $n = 2$ in the general definition (Def 2.5) and the result is shown below:

Definition 2.6. $||X||_2 = \sqrt{\sum_i x_i^2}$

Simply, it is the square root of all the vectors squared. Note that higher values of the elements in X have more impact on the outcome than the small values (due to the squares).

L2-norm is recommended for Gaussian problems and smooth problems. If the problem is spiky, it will try extensively to fit the problem. Sometimes it tries too hard and the effort backfires and nothing useful comes out.

2.2.2 L1-norm

This norm handles the values by their absolute value and not squared. Again, from the general definition, with $n = 1$, the L1-norm definition is:

Definition 2.7. $||X||_1 = \sum_i x_i$

So it is the sum of all values in X. Note that all elements in X have equal impact on the outcome (in terms of order).

L1-norm is often more effective than L2-norm for spiky problems, because it is more simple.

2.2.3 L0-norm

An interesting norm is the L0-norm which, technically, is defined as: $||X||_0 = \sqrt[0]{\sum_i x_i^0}$ But since all $x \neq 0$ will turn into ones, the L0-norm can be treated as a counter that counts all nonzero elements in a vector (X).

2.2.4 L-inf-norm

An other interesting norm is the L-∞-norm, that by the general definition looks like:

$||X||_\infty = \sqrt[\infty]{\sum_i x_i^\infty}$

This looks tricky but by the property of the infinite exponent, the largest element in X will become much greater than every other element, hence, $\sum_i x_i^\infty = max(|x_i^\infty|)$, and finally:

$||X||_\infty = max(|x_i|)$.

2.3 Filter

In signal processing, filter is a function that has the task to remove various unwanted components or features from the input signal. Commonly for discrete-time samples, it is desired to extract, and then be able to observe the actual process only. This is done by removing the noise.

Noise is a collection name for all the high-frequencies in a signal. The definition of high frequencies is, of course, dependent on the context and individual for each case. But they all have one characteristic in common; their frequency is too high to have any relevance.

Definition 2.8. Noise is defined as a compilation of frequencies that are too high to be of any relevance to the context.

2.3.1 Forward-backward Filter

A forward filter uses previous data while backward filter uses upcoming data and therefore results in a shift in time. Forward-backward filter, on the other hand, is a combination of a forward filter and a backward filter. The combined filter applies a function twice, once forward and once backward (hence the name). The reason for applying the function in both directions is to obtain a response with zeroed phase-shift, i.e. there is no time distortion and no time delay in the

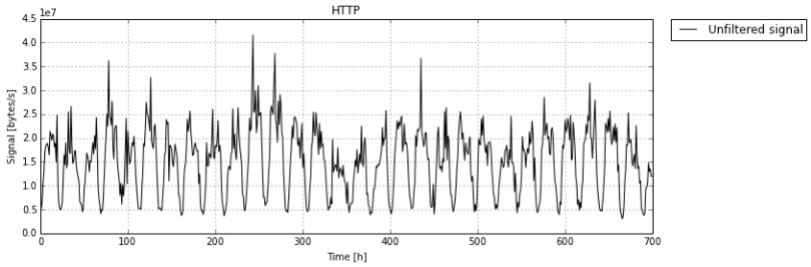

Figure 2.1: A signal that is unfiltered.

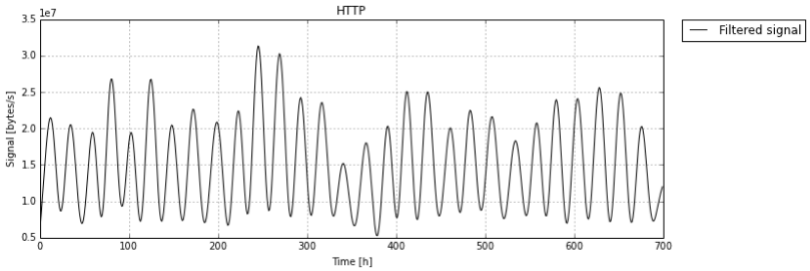

Figure 2.2: A signal that is filtered.

filtered signal. Depending on the intention, either high-pass[1] or low-pass filter can be used, the latter is explained in detail below.

2.3.2 Low-pass Filter

A low-pass filter is a filter that passes low-frequency signals and attenuates signals with frequencies higher than the cutoff frequency, i.e. this is a frequency that acts as a threshold mark and will only try to remove frequencies lower than this mark.

When removing noise, the low-pass filter is a good choice of filter.

The cutoff frequency is usually determined by the user. This filter applies a forward function and a backward function at once, the result is a linear phase. The filter uses both forward and backward filtering to achieve a zeroed phase-shift, i.e. the signal is not moved in time, resulting in no phase distortion and no delay. The Butterworth filter has this property[3], see Figure 2.1 compared with Figure 2.2.

[1]High-pass filter is not explained more than that it is the exact opposite of the low-pass filter. The reason for this is that this type of filter is not used in this paper.

2.4 Dot Product

Dot product or inner product is directly related to the cosine of the angle between two vectors in Euclidean space. It holds for all dimensions and is often written as $\mathbf{A} \cdot \mathbf{B}$. It is defined as the sum of the product of the corresponding elements of the two vectors, see Definition 2.9

Definition 2.9. $\mathbf{A} \cdot \mathbf{B} = \sum_{i}^{n} A_i B_i$, where n is the dimension

The two vectors are perpendicular iff the sum of the dot product is equal to zero, this is the most significant case and it can be used in many areas.

2.5 Fourier Transform

Occasionally, it is the case that the observed event gives no peculiar information. Looking at the exact same event from another point of view can reveal valuable information that would otherwise be hidden in the original view. In many cases, vital information is hidden in the frequency content of the signal. Fourier transformation is a mathematical tool that transforms a time series to a frequency domain, i.e. it translates the signal. The outcome consists of the frequencies that are underlying in the signal. These are located in the frequency spectrum, which is a spectrum containing all of the frequency components of the signal. Hence, the frequency spectrum of a signal shows the frequencies present in the signal.

Hereby, the very definition of the Fourier transform, $F(\omega)$, will follow.

Definition 2.10. $F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$

It can transform signals into the frequency domain but it is also possible to go the opposite direction, namely, the inverse Fourier transformation. The corresponding definition is given below.

Definition 2.11. $f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega$

It is a particular property with the Fourier transform that needs to be lifted, namely, that it has a trade off between time resolution and frequency resolution. What this means for the user technically is on the one hand, having good knowledge about what frequencies that exists within the signal, but on the other having less knowledge about what frequencies that exists within the signal but more knowledge about when they occurred.

2.6 Wavelet Transform

Wavelet transform, as Fourier transform in Section 2.5, is a mathematical transformation between time domain and frequency domain. It is a widely used method and its popularity is derived from the disposal of the time/frequency

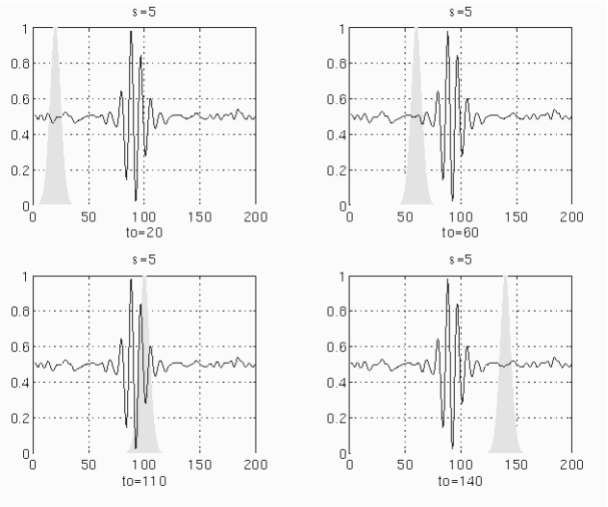

Figure 2.3: A narrow wave in a wavelet transformation at different times. [6] (Polikar, R., 2014. *The Wavelet Tutorial*. pp.38)

solution trade off. As a matter of fact, wavelet can display both good time resolution and good frequency resolution. In many applications it is of great relief not having to consider this extra task when optimizing the time resolution. This localization is one of the major reasons why wavelets are used instead of Fourier transforms. The name of the method is sort of self explanatory, for wavelet functions can be considered as a wave that travels through the signal. It travels with one input variable, the translation variable, and the width of the wave is determined by another input variable, the width variable, that is set by the user of the algorithm. The impact of the width variable is illustrated in Figure 2.3 and 2.4 where a small and a larger value is used. Also, the traveling motion of the wave can be seen.

2.6.1 Mathematical Approach

A vector $\mathbf{v}$ can be written as a linear combination of the basis vectors $\mathbf{B}$ in that space where α contains the corresponding coefficients, see Equation (2.1). The number of basis vectors is always equal to the dimension of the space, also the number of elements in every basis vector.

$$\mathbf{v} = \sum_{\mathbf{k}} \alpha^{\mathbf{k}} \cdot \mathbf{B}_{\mathbf{k}} \tag{2.1}$$

The vectors in Equation (2.1) can be generalized to functions by replacing the basis vectors $\mathbf{B}_{\mathbf{k}}$ to $\phi^{\mathbf{k}}$ and $\mathbf{v}$ to a function $f(t)$, see Equation (2.2) (if sines and cosines are chosen here, the Fourier transform function is obtained).

$$f(t) = \sum_{\mathbf{k}} \alpha^{\mathbf{k}} \cdot \phi^{\mathbf{k}} \tag{2.2}$$

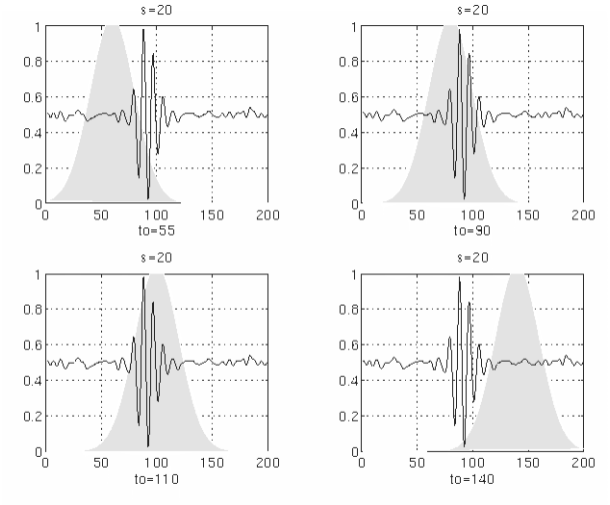

Figure 2.4: A wide wave in a wavelet transformation at different times. [IBID] (Polikar, R., 2014. *The Wavelet Tutorial*. pp.39)

Let $f(t)$ and $g(t)$ be two functions, such that $f(t) \in L^2(\mathbb{R})$ and $g(t) \in L^2(\mathbb{R})$, i.e. $f(t)$ and $g(t)$ belongs to the set of square integrable functions. Inserting this in the definition for the inner product (see Definition 2.9), for the continuous case with functions that are square integrable on the interval [a,b], gives Equation (2.3).

$$\langle f(t), g(t) \rangle = \int_a^b f(t) \cdot g(t) dt \tag{2.3}$$

Here, if $f(t)$ is assigned to the input signal and $g(t)$ is assigned to the basis function(s), or in other words, the wavelets, the continuous wavelet transformation function is obtained, see Equation (2.4).

$$CWT_x^{\psi}(\tau, s) = \int x(t) \cdot \psi_{\tau,s}(t) dt, \text{where } x(t) \text{ is the input signal and } \psi_{\tau,s}(t) \text{ are the wavelets, which are given in Equation (2.5) below.} \tag{2.4}$$

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \tag{2.5}$$

The interpretation of this is that the output is the similarity in the sense of frequency between the input signal and the wavelet.

Wavelet is self-dual, i.e. it can transform a signal into the frequency domain and back to the signal amplitude domain with the inverse of the CWT, however, it is excluded here.

2.7 Fundamental Statistics

Some necessary statistic knowledge are here being presented.

2.7.1 Null Hypothesis

In statistics, and in experiments specifically, a null hypothesis is a statement that says whether the studied object has any impact on the outcome or not. If the studied object has impact, the null hypothesis is rejected (and not rejected otherwise), see Definition 2.12.

Definition 2.12. A null hypothesis is a statement that indicates no relation between two objects. The null hypothesis is, after experiment(s), either rejected or disproved.

Scepticism must however always be directed towards the result, for in statistics, randomness is always an obstacle. Therefore, the result must not only reach the given barrier but also exceed this barrier with a value (a value that is often set by 5 % or 1 % significance level, i.e. the test will most likely succeed in 1 out of 20 or one 1 of 100 times of the time respectively) so that the probability of an extreme occurrence is low. If the results meet the target but not the extra value, there is a probability that the null hypothesis could be either true or false. This results in a weak conclusion, for the evidence is not enough.

In medical terms, e.g. if a new drug intends to reduce the risk of getting cancer, a possible null hypothesis would be: "this drug does not reduce the risk of getting cancer".

2.7.2 Type I Error

Errors of the first kind are errors that indicates an event being confirmed when it in fact did never occur.

Definition 2.18. If a null hypothesis is rejected falsely, it is said to be a Type I Error.

Consider the case where a doctor says to his patient that she has cancer, but in fact she is healthy, that is a Type I Error. This type is also known as false positive error.

2.7.3 Type II Error

In some sense, Type II Errors are the opposite to Type I. The difference is now that the null hypothesis is accepted instead of being rejected.

Definition 2.19. If a null hypothesis is accepted falsely, it is said to be a Type II Error.

Consider the doctor scenario again, but this time he says to the patient that she is healthy, but in fact she has cancer, then he has committed a Type II Error. This type is also known as false negative error.

In matter of significance, what error type is most important to avoid, it always depends on the context. For the doctor, of course, the Type II Error is much worse, for then, no treatment will be carried out.

2.8 Toeplitz Matrix

A Toeplitz matrix, named after the German mathematician Otto Toeplitz, is a matrix in which each descending diagonal is constant, i.e. $a_{i,j} = a_{i+1,j+1}$. It is easy to show with an example, see T_n in Equation (2.6) where T_n is the Toeplitz matrix with order N.

$$T_n = \begin{pmatrix} a_1 & b_2 & b_3 & \cdots & b_N \\ a_2 & a_1 & b_2 & \cdots & b_{N-1} \\ a_3 & a_2 & a_1 & \cdots & b_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_N & a_{N-1} & a_{N-2} & \cdots & a_1 \end{pmatrix}, \quad \begin{array}{l} \text{where } b_i = a_i \text{ in a} \\ \text{symmetric Toeplitz} \\ \text{matrix.} \end{array} \tag{2.6}$$

Chapter 3

Methodology

In order to create virtual models of real problems, interpretations must be made. It is done here in this chapter, along with explanation of the three models; Linear Predictive Coding, Sparse Linear Prediction and Wavelet Transform.

3.1 Mathematical Interpretation

As described in Chapter 1, anomaly detection is a vague term. Vague objectives have a higher possibility to lead to vague conclusions. To prevent this, inventions for more precise definitions of the goals will hereby follow, starting with the main objective (detecting anomalies in the signature database).

Task 1. *Anomaly detection.* Given data set D, find subset A $\subseteq$ D where A is anomalous.

It is possible for A to be an empty set, for in that case the observed signature works as desired.

An experiment like this is most surely expected to not obtain a perfect result without either error type. Therefore, an additional task is to keep the level of Type I & II Errors (see Def. 2.18 and Def. 2.19) as low as possible.

Task 2. *Parameter optimization.* Improve the parameters in the algorithm until changes have no significant impact on the outcome in the sense of Type I Error and Type II Error.

An interesting proceeding task is to consider which error type that causes the most damage. Also, to see if there is any possibility to improve one of them separately, leading to the third task.

Task 3. *Error Type analysis.* Inspect and find out if one of the Type Error is more hazardous than the other.

Hereinafter, a detailed version of the two states in the signature are explained.

3.1.1 State Classification Assertion

When determining states in a time series, it must be kept in consideration what type of supervision there is in the ambient. Different algorithms are working under different supervision.

When a reference data set is available, parameters for the algorithm will be extracted from this and also be used as an underlying foundation for the algorithm evaluation. The input data will be observed and evaluated from the reference set to see if it is of similar nature or not.

The more supervision that is accessible, the easier it is to implement the algorithm and generally it will result in a more accurate result. The reason for this is that it is easier to recognize a state if the behaviour of this state is known. All that is needed is to compare the input data with the reference and make a judgement whether or not they are similar in the sense of context. On the other hand, the downside is that more preparatory work is needed. Sometimes it is difficult to obtain these references and it requires a lot of work, or even worse, sometimes they are impossible to obtain. An example of this is if a company just invented a brand new machine that are to produce a very rare item, then no data of the machine's performance yet exist.

When a reference data set is not available, the input data itself will be used as ground in evaluation. The input data will be observed and evaluated to see if there is any subset of the input data set that is of foreign nature.

Bottom line is; supervision is a trade-off between the amount of preparatory work and performance.

3.1.2 Normal State

If a reference exists for the normal state (supervised and semi-supervised with normal reference), this will be used as template for the definitions of a normal state. On the other hand, if there is no reference for the normal state, there are two possible scenarios. Either there is an available reference for the anomaly state or it could be that the anomaly state is not accessible, or nonexistent. In the first case (semi-supervised with anomaly reference) the anomaly state should be determined primarily. What is not considered as an anomaly should simply be classified as normal by elimination, i.e. the anomaly is the first layer and what does not stay in this filter is classified as normal. However, in the latter scenario (unsupervised), the machine has nothing to learn from but from the actual data set, i.e. it has to work online.

Definition 3.1. A state is classified as normal when it exists no significant, and statistically proved, deviation in terms of amplitude, periodicity and trend.

In supervised and semi-supervised with normal reference, as the normal state is set as reference, it is of great interest that the observed data set, that is to be used as reference, is really behaving as a normal signature.

3.1.3 Anomaly State

The anomaly state is the antagonist to the normal state, i.e. it is the opposite state. It is interpreted in a similar fashion. A state is classified as anomalous if it is no longer behaving normal. Thus, a normal state is necessary for another state to be anomalous, for otherwise it has nothing to distinguish itself from. If the anomaly state is the primary state and is to be analyzed first, a definition is necessary.

Definition 3.2. A state is classified as anomalous when it is no longer normal, i.e. a significant and statistically proved deviation can be presented in any of the following: amplitude, periodicity or trend. Also, a change in the noise can be argued to be included as an anomaly.

Further, if there is a reference set available for the anomaly state, the machine will only be able to detect that particular anomaly that the machine is taught to detect. The problem with this, in this context, is that all kinds of anomalies are to be detected. Therefore, it is very difficult to find all anomalies and teach them to the machine. The conclusion of this is that supervision will be used but only with the normal, and known, state (Semi-supervised learning with normal reference).

It can be the case that some types of anomalies (e.g. trends) are most distinguished by one model and other kinds (e.g. temporary spikes) are easier to detect by looking at the problem from another aspect. That is one of the reasons why several models are surveyed in this paper.

Deviations will always be present, and of course, some of these must be accepted for they are natural fluctuations. Therefore, the acceptance level of the various deviations have to be determined empirically. This will be solved by putting a threshold on the output, that has to be exceeded before classifying the item as an anomaly. This threshold will be decided by the user to achieve the objectives in a reasonable matter. This is discussed for each goal in each method, see Section 3.3 through 3.5.

3.2 Input Data

The input data is a discrete time series with hourly bias, i.e. every 60 minutes. All data traffic that has been used during that hour is summarized and stored in a data D_{test} variables are the average upload and the average download speed during one hour. Throughout this paper the sum of average upload and download speed is the only input variable. They could be examined separately but that is only included in the future work, see Section 5.8. The input data set is hereinafter referred to as D and elements in D are called d_i. Also, the reference part of D is called D_{ref} and the test part is called D_{test}. Furthermore, there is a special signature called "Unknown". This signature contains all the data that is transferred by signatures that is yet not known to Procera.

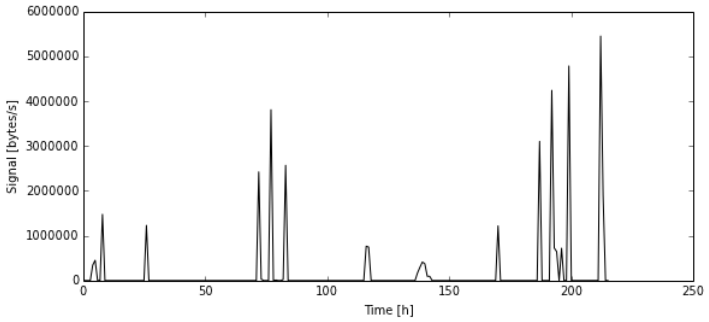

Figure 3.1: A signature with low volume.

3.2.1 High Versus Low Volume Signatures

Signatures with low data traffic rate behaves very different compared to high volume signatures, and even compared to other low volume signatures as well. The reason for this is that they have low usage (few active users) that the average speed is located at zero or very often close to zero, see Figure 3.1.

The issue that arises because of this is that statistics cannot play its full role. In a low volume signature, one user or one event can affect the time series a lot. The actual distribution is hidden behind the natural deviations, which all become very strong in a low volume signature. On the other hand, in larger scaled signatures, one deviation will not affect the shape of the time series that much, it will most likely be unnoticed. Bottom line is; it is easier to detect the true distribution in a signature the larger the usage is. In fact, it is almost impossible to understand what the underlying distribution is in Figure 3.1. Consequently, a threshold, in terms of transferred data, is created that must be exceeded for the signature to be examined by the algorithms (explained in Section 3.3 through 3.5). It might be the case that a signature varies around this value, hence, it is examined every other week, but not the next week and so on. To prevent this, the threshold for a signature to not be examined is set to a lower value. These two threshold values are determined empirically, see Section 3.2.2.

3.2.2 Analysis Threshold

There are way too many signature to analyze them all by inspection, but empirical studies show that the majority of the signatures with mean value greater than 10^5 has a shape that is similar to what is desired, as in Figure 1.2. Hence, signatures with mean value below 10^5 are more similar to what is more difficult to analyze, as in Figure 3.1. The threshold for a signature to move from being a "low volume" signature to a "high volume" signature is set to twice that number, i.e. $2 * 10^5$. The threshold for a signature to move from being a high volume signature to a low volume signature is set to half that number, i.e. $5 * 10^4$. Bottom line; only signatures with mean value above 10^5 will be

examined. If a non-analyzed signature (small) is growing, it needs to exceed $2 * 10^5$ to be classified as a large volume signature and large signatures need to drop below $5 * 10^4$ to be classified as a low volume signature.

3.3 Linear Predictive Coding Framework

A potential approach to detect anomalies is to use Linear Predictive Coding (LPC). This is a mathematical model using a forward linear predictor to predict values in a discrete-time signal based on previous data. It creates a data set Y by estimation, using parameters that are obtained when the minimization of the relaxed error $(d_i - y_i)^2$ is minimized, where d_i is the elements in the input signal and y_i is the elements in the estimated LPC signal. LPC model uses L2-norm, i.e. $||d_i - y_i||_2$.

Definition 3.3. LPC framework is defined by the following. Given data set D, use information from D to obtain coefficients α that are able to build a forward prediction data set Y. Determine if each $y_i \subseteq Y$ is anomalous by considering how much the nature of Y diverges from the original set D, where $i = 1, 2, ..., k$ and k is the number of elements in the analyzed data.

In its context, the algorithm is supervised in the form of semi-supervision with reference for the normal state. Hence, the machine learns how the normal state is defined and will indicate that an anomaly has occurred when the machine considers that the signal has left the normal state and entered the anomaly state[4]. This algorithm is implemented in a way explained in the following steps:

1. **Filter the signal** with intention to remove the noise in the signal. This is done in Python programming using SciPy's function '.filtfilt()', which is a forward-backward filter. This results in a much more clear signal, which could be expected to represent the actual process signal.

2. **Calculate LPC coefficients** C, with the signal and together with the LPC order N as input and the LPC coefficients as output. The number of coefficients is equal to the order plus one $(N + 1)$. Find the $N + 1$ coefficients of an N order linear filter:

   $y_i = -c_1 * d_{i-1} - ... - c_{N+1} * d_{i-(N+1)}$, where i is the index in D

   Such that the sum of the squared-error $e_i = y_i - d_i$ is minimized[4].

   LPC determines the coefficients of a forward linear predictor by minimizing the prediction error in the least squares sense. The C-vector will take the form $C = [1\ c_2\ ...\ c_{N+1}]$.

3. **Calculate the alphas.** This is done by using a Toeplitz matrix. In this context it is also symmetric, but this is not a property of the Toeplitz matrix, it just happened to be symmetric for this task. Hence, the entire matrix can be filled in with only the first row or column given. Furthermore,

it is of size $N \times N$, where N is the order (or equivalently, $N = length(C)$, where C is the vector from the previous step). An important thing is that it is the first element of C that is removed.

These two A system that has the α's as output, see Equation system (3.1).

$$\begin{pmatrix} C_1 & C_2 & C_3 & \cdots & C_N \\ C_2 & C_1 & C_2 & \cdots & C_{N-1} \\ C_3 & C_2 & C_1 & \cdots & C_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N & C_{N-1} & C_{N-2} & \cdots & C_1 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} C_2 \\ C_3 \\ C_4 \\ \vdots \\ C_{N+1} \end{pmatrix} \tag{3.1}$$

Note that compared to the c_i's, one less α is obtained, i.e. N. The C-vector is normalized with respect to the first element, i.e. $c_1 = 1$ (which is why it is of no interest to use this element). It is cut from the beginning. Which is not the case for the Toeplitz matrix, where the last row and column are eliminated to adopt the same size, i.e. $\mathbb{R}^N$. The α's are gathered together, $A = [\alpha_1\ \alpha_2\ ...\ \alpha_N]$.

4. **Create the LPC transformed signal** Y, in order to represent the input signal, see Equation (3.2). It can be considered to be located in a parallel space in time.

$$y_i = \sum_{j=1}^{N} a_j \cdot d_{i-j}, \quad \begin{array}{l}\text{where } N \text{ is the order and } i \text{ is the index} \\ \text{for the elements in } Y.\end{array} \tag{3.2}$$

5. **Analyze the difference** between the input signal and the Y-signal, but to be able to do this fairly, both signals must be normalized first.

6. **Distribute score** to the difference signal, s.t. a score is given to every item in the data set. Points that has a score outside of a threshold are considered anomalous. This is explained in detail in Section 3.5.4.

3.3.1 Parameters

Some input values are having more impact on the outcome than others, therefore, they are tested for different values. The parameters that are having less impact are static, e.g. the filter length is always 8 hours[1], see Table 3.1.

The order of the LPC method is limited to 1 - 5 days. There are two reasons for this. First, the period of the signatures are 24 hours so the method works better if the order match the period[2]. Secondly, the higher the order is the more complex the method becomes. If the system is too complex it has a possibility of becoming unstable.

[1] Empirical result
[2] Empirical result

Parameter	Value
Forward-backward filter length	8 Hours
Order	1 - 5 days
Window length	1 - 7 days
Threshold value	$(E[D_{ref}] + SD[D_{ref}]) * k^*$

*Where D_{ref} is the reference data set and k is a factor (function) dependent of order and window length.

Table 3.1: The different input parameters with their corresponding test values.

Upon request from Procera, anomalies were to be detected at least within a week, therefore, the window length does not exceed one week. The deviations in the model are added up together for a period back in time, in a window-like manner. The window length decides how many elements that are included. Hence, it is also a measure of how far back in time the window stretches, so if results are desired within a week, the window length cannot exceed one week.

Theoretically, higher order results in a more robust signal, which is good in the sense that it is able to mimic the actual signal better. However, if it is too good to mimic the signal, it mimics the deviations as well. This is avoided because deviations will then be undetected.

Higher window length reduces the probability of a natural fluctuation to be visible, which is a good thing. However, long windows will also smooth out temporary deviations and therefore higher possibility to pass undetected.

The input values are (1) order, (2) window length, (3) anomaly type and (4) evaluating 16 different artificial anomalies. This results in a huge amount of combinations. In every evaluation, the filtered input signal is displayed together with the calculated LPC-response, also the difference between these two are included. The difference function is acting as a deviation measurement. This is possible because the LPC-transformed signal is normalized, so it is only the difference in nature that appears in the difference function. Furthermore, a window summation is created for the difference so that natural fluctuations are having less impact on the outcome. A small number of plots are shown with corresponding window sum, they can be seen in Section 4.1.1.

Extra attention is paid to the evaluation of the point anomaly and the changing trend. The reason for this is that the point anomaly is an excellent example of a fast deviation and the changing trend of a slow deviation. It is of interest if these two categories can be classified by the model too so that it can avoid additional manual work. Therefore, these two anomalies are fully evaluated with two different orders, 3 and 5, together with window lengths 2 and 7 correspondingly. Regarding the other 14 anomalies, only the LPC results are displayed (to keep this thesis at a reasonable length), see Section 4.1.

3.4 Sparse Linear Prediction Framework

There is a saying, that if two methods produce equally good results, the easiest method should be used. Basically, The Sparse Linear Prediction (Sparse LP) model is similar to LPC but it has a higher chance for increased simplicity. Sparsity is a measure of simplicity. The model is similar to the LPC in the sense that they both minimizes the errors e_i^2 and they produce coefficients, α, that are used for forward prediction and N is the order of the model. But they differ in one major way, Sparse LP has an additional input parameter, γ, which is used as a sparsity parameter. What this does is that it puts a penalty on having only non-zero elements within the alpha's, correspondingly to what gamma is used[5]. Hopefully, this results in a more elegant alpha vector containing only non-zero elements at the most "valuable" locations. In this context, valuable means a location that has more impact on the outcome than other elements.

Definition 3.4. Given data set D, use information from D to obtain coefficients α, such that the error $||d_i - y_i||_1$ is minimized, where d is the input signal and y is the estimated signal. Non-zero elements in α will be penalized depending on input parameter γ, use as sparse α as possible without ruining the result. A forward prediction data set Y is constructed from these α's. Determine if each $y_i \subseteq Y$ is anomalous by consider how much the nature of Y diverges from the original set D.

Furthermore, Sparse LP could be argued to be considered as a L0-norm model, for it puts a lot effort to produce sparse outcomes.

A step-by-step description could be made for this method too, but it would look similar to the LPC model, the only difference in the method is at the second step (see Step 2) where the extra input parameter (γ) is implemented to decide how much non-zero elements are to be punished. Another task comes with this model, namely to trim γ so that the results of the model are similar to the LPC model but with less complex α.

3.5 Wavelet Framework

It is mentioned in Section 2.5 that observing an event from different aspects can result in interesting outcomes. Wavelet transform is a time-frequency transform, meaning that it transforms a time signal (with an amplitude) into its frequencies at the same time interval[6]. Looking at the frequency spectrum of a signal will perhaps bring revelations that the two previous models did not accomplish.

Intuitively, frequencies has something to do with the change in rate of a process. If a variable in the process changes rapidly, it is said that it has a high frequency. If a variable in the process does not change rapidly, i.e. it changes smoothly, it is of low frequency. If this variable does not change at all, it is said to have zero frequency. This is the main concept of this method and how it will fit in the context. It is the unexpected changes in rate of the signal that are sought for.

The wavelet model does only have two inputs, the choice of wavelet family and the input signal. Furthermore, wavelet analysis requires no preparation and in this context it is an unsupervised machine learning model. However, a reference part is needed anyway to create a threshold, but this is explained further in Section 3.5.2

3.5.1 Wavelet Transform

The explicit application for wavelet transformation is to extract the momentary frequencies from a signal. The wavelet transform is a bit more complicated transform to perform in theory than the LPC and SLP model. Briefly, a description of the concept of a wavelet is that a wavelike function (hence, the name) travels through the signal and calculates the momentary frequency in every step.

In Python programming, there are many built-in functions and add-ons that are able to perform these transformations. The available wavelet families are listed below in Table 3.2.

Wavelet name	Filter length	Abbreviation
Daubechies	2-40	db
Symlets	4-40	sym
Haar	2	haar
Coiflets	6-30	coif
Biorthogonal	2-18	bior
Reverse biorthogonal	2-18	rbio
Discrete Meyer (FIR Approximation)	62	dmey

Table 3.2: Built-in wavelet families in Python programming.

In total, there are 75 different wavelets that are available, including combination of filter length and types. The approach to all this is to greatly reduce this number by confirming that some has almost equal outcome, or less useful outcome than others. For example, after a thorough investigation has been done, conclusions can be drawn like: Daubechies wavelets only needs to be used with wavelength 2, 10 and 20, for then all interesting outcomes are covered. Even more reduction is performed in this paper, for only Daubechies with the longest filter length is used. The other are added to future work in Section 5.8.

3.5.2 Thresholds

To find an anomaly in the outcomes of the models are easy to interpret by inspection, it is simply to pin-point the deviation right away. But for a computer it is a bit more complicated, especially since it must be able to do so for all kinds of signals with different magnitude. Thresholds are one way to solve this problem.

Judging the outcome by absolute value is not fair, for when comparing deviations between different signatures, some signatures uses much less data in general than perhaps a more popular signature. In that case, a small deviation for a large signature is equal to a large deviation for a small signature. This does not satisfy the demand. Hence, the theory is rejected.

Judging the outcome by percentage is not fair either, because at the outliers at lower values of the signal (often by night), every deviation will appear to be larger than at higher values at the signal (often by day). Hence, this theory is rejected too.

For semi-supervised anomaly detection, a reference data set is available. From the reference set, extract information such as mean value and standard deviation. These values are not able to tell anything about deviations but they provide an indication of the magnitude of the examined signal. When the magnitude is known, let every deviation be compared with it, and then, finally, find out how the ratio looks like. For the LPC model an empirical definition is invented and is hereby explained along with a definition.

The threshold value equation is designed by inspecting several results from the LPC model, and then the threshold value is placed at a desired level. From this, a table of several threshold values that are connected with its corresponding parameters is created. A function is then created with objective to fit the table. The variable name for the threshold value is THV and the formula can be seen in Definition 3.5.

Definition 3.5. $THV_{LPC} = (\frac{E[D_{ref}]}{30} + SD[D_{ref}]) * 0.06 * (1 + \frac{24}{N_{LPC}}) * (1 + \frac{168}{WL})$, where D_{ref} is the reference data set, E and SD is the mean value and the standard deviation respectively, N_{LPC} is the LPC order and WL is the window length.

An increase of any of the variables result in a larger threshold value. However, the order and the window length do not have as much impact as the expected value and the standard deviation.

For wavelet (that is unsupervised) the threshold value is the maximum value of the outcome frequencies during the reference set times 1.5, see Definition 3.6.

Definition 3.6. $THV_{wavelet} = max[D_{ref}] \times 1.5$, where D_{ref} is the reference data set and max is the maximum value.

3.5.3 Window Sum

A moving window sum function, W, is applied to the output signals. How this is implemented is explained in detail in Definition 3.7.

Definition 3.7. $w_i = \sum_{i=N+M}^{k} \frac{y_i}{k-(N+M)}$, where y_i are the elements in the output signal Y, k is the length of input data set D, N is the order of the model and M is the window length.

Elements w_i are stored in the data set W. So W is a vector of length $k - (N + M)$, namely, $W = [w_{N+M}\ w_{N+M+1}\ ...\ w_k]$.

3.5.4 Anomaly Score

A score, AS, is given to every signature for every time step after the window sum, W, is created. It is considered as an indication of how deviating the deviation is. It is defined as following:

Definition 3.8. if $|d_i| > THV$, then $AS = \sum_{i=a_f}^{a_l} \frac{w_i^2}{THV \cdot (a_l - a_f)}$, where w_i is the elements in W, $i \in \mathbb{N}$, a_f and a_l are the first and the last anomalous element in the anomalous sequence and k is the length of the input data. Note that i continues from a_l if there are several anomalous sequences in the same data set (the second sum starts when the second anomalous sequence starts).

Note that anomaly scores below one will not result in anything (it is not an anomaly) and for a sequence that enters the anomalous area briefly has anomaly score close to one, $AS > 1$, whilst extremely deviating sequences has much higher anomaly score, $AS >> 1$.

3.6 Trial of Methods

The methods are compared to each other in terms of what method is most suitable to detect anomalies that are included in this context. The most successful method is the method that has the highest rate of detecting anomalies and also has the least amount of Type I & II Errors.

3.7 The Test

In order to see how well the models perform, a number of made-up errors are added in some signature signal with purpose to be detected by the model. The test is employed in the following way, namely that a relatively normal signal (i.e. no anomalies) is chosen and the first part is representing the reference data set, where the coefficients are collected, and the second part is the manipulated part where the artificial anomalies are implemented and is used as a performance test. The second part is the evaluation part where the anomaly score is assigned.

3.7.1 Error Type Test

The models are also investigated in detail to find the number of Error Types by running it on several signatures. This test is performed when all signatures are in normal state to detect Type I Error and also performed when each signature having an anomaly, this to detect Type II Error. In the latter part of the test only two artificial anomalies are used, namely, point up {1} and trend up {2}. To be able to do this on a large amount of signatures, a function for these two anomalies are created. For the point anomaly the function takes two late points and adds $8 * E[D_{ref}]$ to them. For the trend anomaly the function takes the

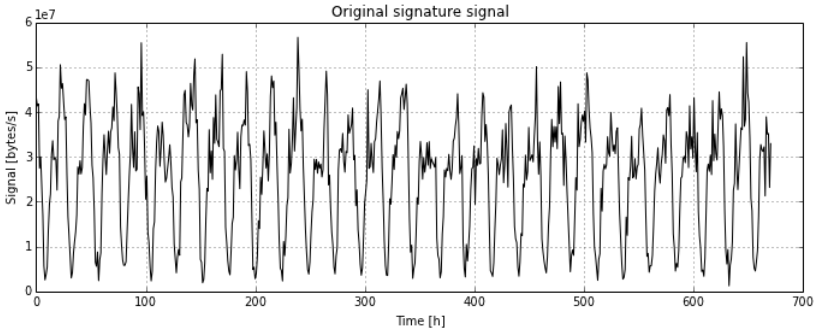

Figure 3.2: This is the original signal that is used in every artificial anomaly set.

mean value of the reference data set and adds it successively to the test part of the signal.

In this Error Type test, another variable is added (Threshold multiplier). What this multiplier does is that it changes the threshold value to a larger or smaller value and the reason for this is to investigate the impact of the threshold value.

3.8 Artificial Anomalies

The original signal that is chosen is a four week long signal and it is selected with care, so that the deviations are equally small during all four weeks. The reason for this is to get as fair outcome as possible, and also make the artificial anomalies easy to visualize. The first two weeks are reference data and the last two weeks are the working space (test data). The chosen signal that is to be modified is in the top ten of the signals with most traffic and can be shown in Figure 3.2.

Hereafter, a huge amount of figures of artificial anomalies will follow, to facilitate for the reader, they are grouped below and connected with a reference number. The artificial anomalies come pairwise, one anomaly directed down and one directed up. Also, they are grouped in two categories, namely local and global anomalies. Anomaly [1], [2], [5], [6], [7] & [8] are local anomalies. Anomaly [3] & [4] are global anomalies. The signature in [9] is not categorized, for no artificial anomalies are implemented, but it is still included in this project for it behaves differently compared to the other.

[1] Point anomaly (Fig. 3.3)

[2] Trend-jump (Fig. 3.5)

[3] A trend changing direction (Fig. 3.7)

[4] Change in usage (Fig. 3.9)

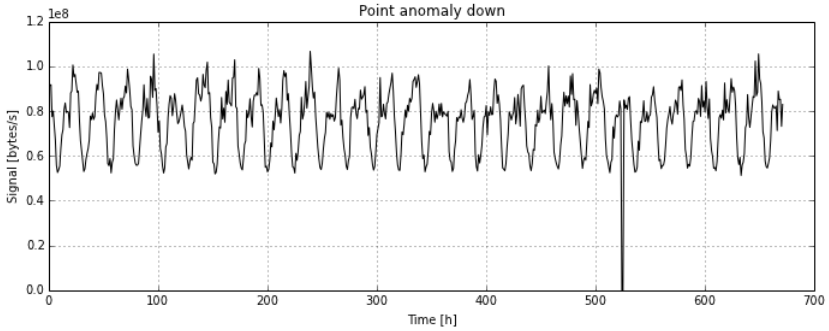

Figure 3.3: A downward-pointing point anomaly.

[5] Sudden change in periodicity (Fig. 3.11)

[6] Sudden change in frequency (Fig. 3.13)

[7] Anomaly located in the reference data (Fig. 3.15)

[8] Multiple anomalies located in the test data (Fig. 3.16)

[9] A special signature that does not correlate to the other signatures (Fig. 3.17)

In Figure 3.3 a point anomaly is added {1}. To clarify, the anomaly is not to be confused with an extra bad night, the anomalous point is located in a peak. Potential phenomenon: a signature having a server shut down.

In Figure 3.4 another point anomaly {1} is inserted but in the reverse direction. The anomalous point is located in a lower subsequence. Potential phenomenon: a signature having an unexpected bug requiring a lot of data traffic.

In Figure 3.5 a jump in terms of trend {2} is displayed. The trend is (in the context) constant except at the center where it "jumps" and continues in a much lower average value. Potential phenomenon: a signature, suddenly, having trouble working at one platform (e.g. iOS[3] or Android[4].) due to poor software updates. Its antagonist can be seen in Figure 3.6.

The purpose of manipulating the signal in such manner as in Figure 3.7 and 3.8 is to discover the effects of slow permutations.

Figure 3.9 and 3.10 also represents slow, long-term changes but a bit more realistic in the sense that the minimum and maximum value for every cycle (day) is changed proportionally and not additive.

What is illustrated in Figure 3.11 and 3.12 is weaker and stronger periods, correspondingly. The purpose of these are to examine what happens if, e.g. a signature becomes more affected of the time of the day.

[3] A platform developed by the company Apple.
[4] A platform for smart phones, developed by Google.

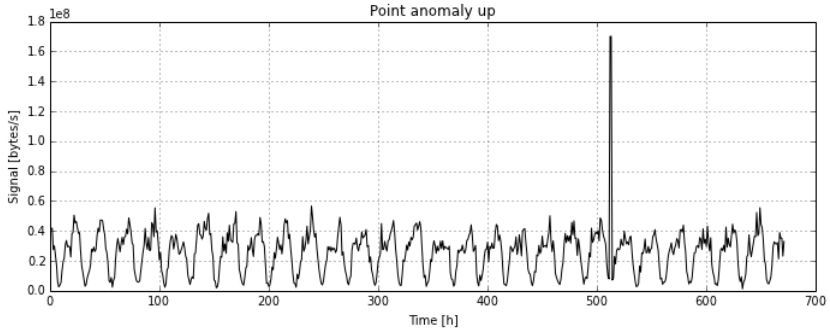

Figure 3.4: An upward-pointing point anomaly.

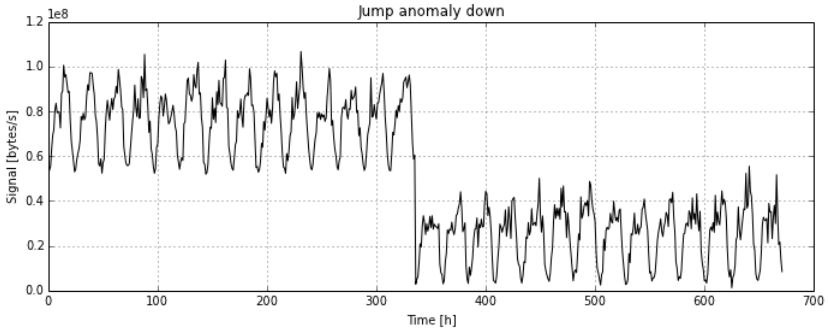

Figure 3.5: A jump in terms of trend, the trend is constant except at the center where it "jumps" and continues to be constant but a lower magnitude. Potential phenomenon: a signature that is using data traffic flow in the background, suddenly, the background data stop working.

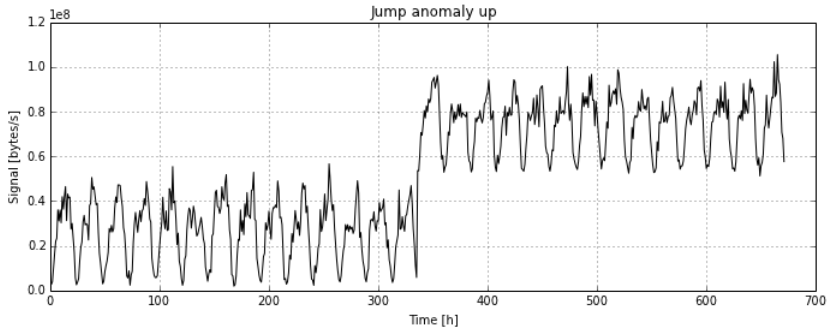

Figure 3.6: A jump anomaly in terms of trend.

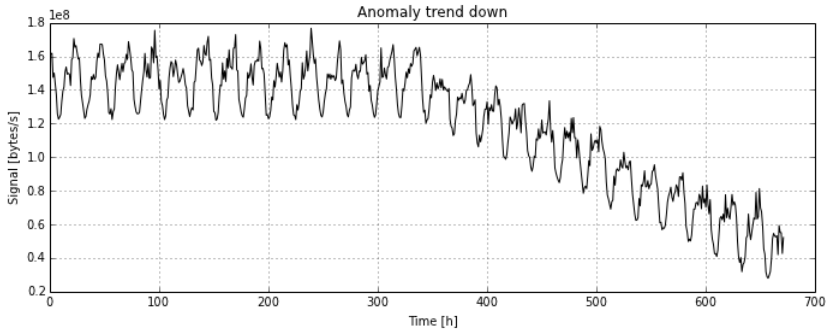

Figure 3.7: A change in terms of trend, the trend is, at the center, changed dramatically downwards. Potential phenomenon: a signature suddenly starts to lose users slowly.

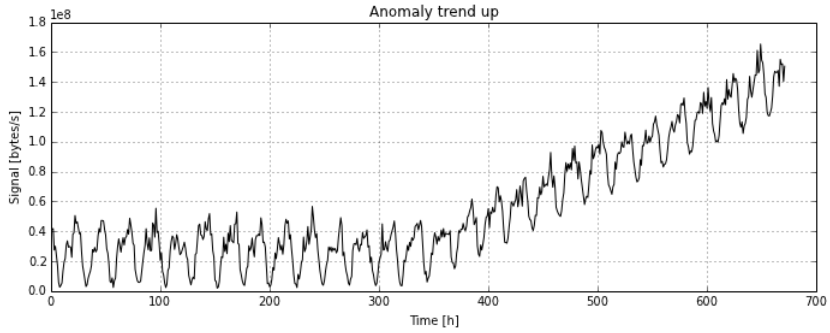

Figure 3.8: A change in terms of trend, the trend is, at the center, changed dramatically upwards. Note that the difference between the highest value and the lowest value for every day is still constant. Potential phenomenon: the signature slowly starts to use data traffic in the background.

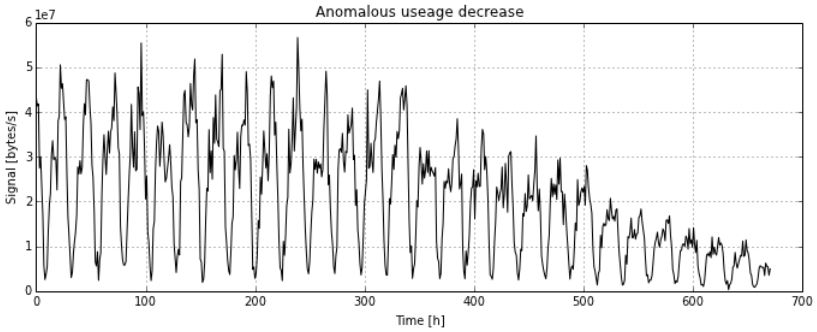

Figure 3.9: A change in terms of employment. The highest and lowest value are both, after some time, greatly decreased by percentage. Potential phenomenon: the signature loses the interest of the users due to different reasons.

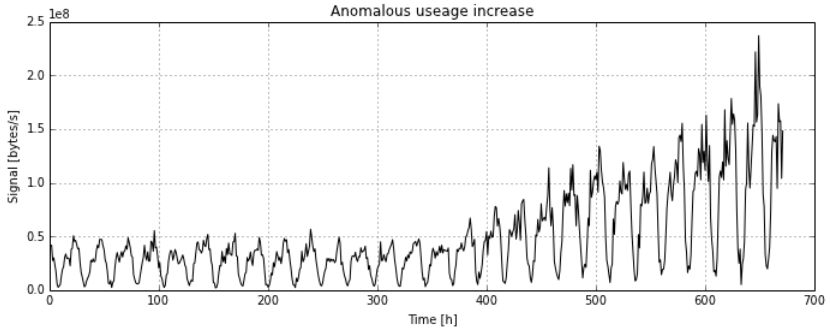

Figure 3.10: A change in terms of employment. The highest and lowest value are both, after some time, greatly increased by percentage. Potential phenomenon: a new commercial increasing the amount of users and data traffic. Requesting more servers.

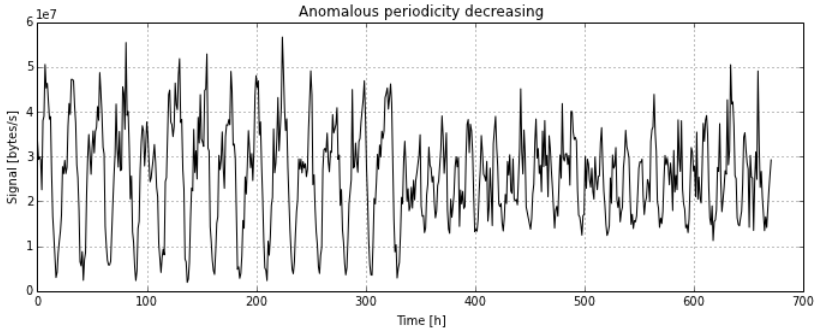

Figure 3.11: A change in terms of periodicity. The periodic oscillations decreases after half of the time series. Potential phenomenon: the signature uses more data in the background and less data while active.

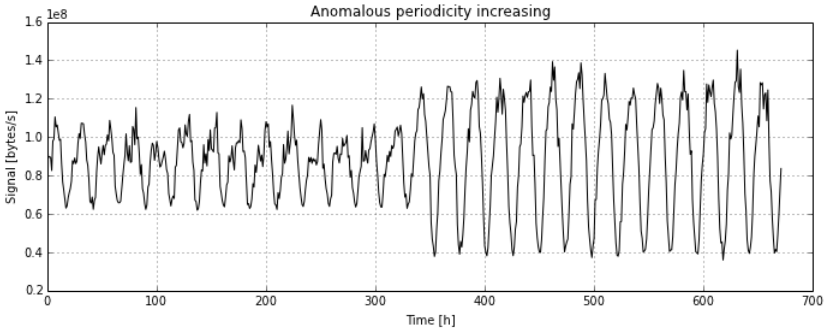

Figure 3.12: A change in terms of periodicity. The periodic oscillations increases after half of the time series. Potential phenomenon: the signature uses less data in the background and more data while active.

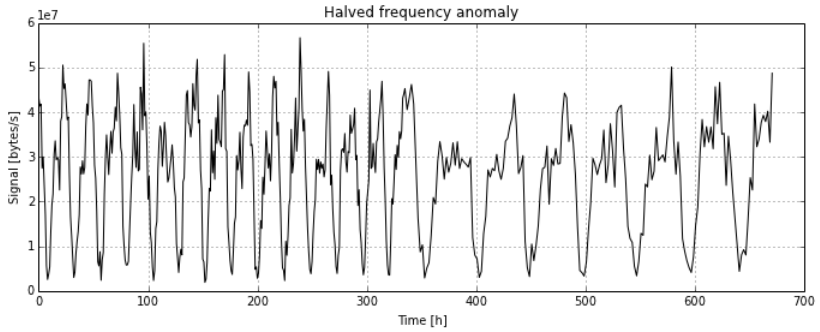

Figure 3.13: A change in terms of frequency. The frequency is halved to 48 hours after half of the time series. Potential phenomenon: the signature suddenly is only used during some days and is ignored the other days.

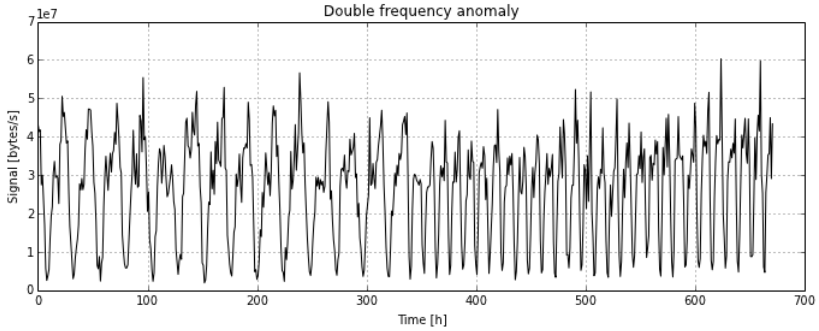

Figure 3.14: A change in terms of frequency. The frequency is doubled to 12 hours after half of the time series. Potential phenomenon: the signature is suddenly only used in mornings and evenings and not during noon.

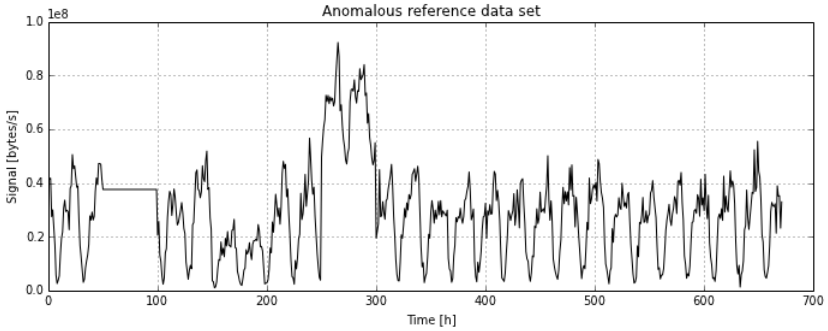

Figure 3.15: A signal with anomalous reference data set. This kind of anomaly is of interest for supervised and semi-supervised models, where the reference data is used as template.

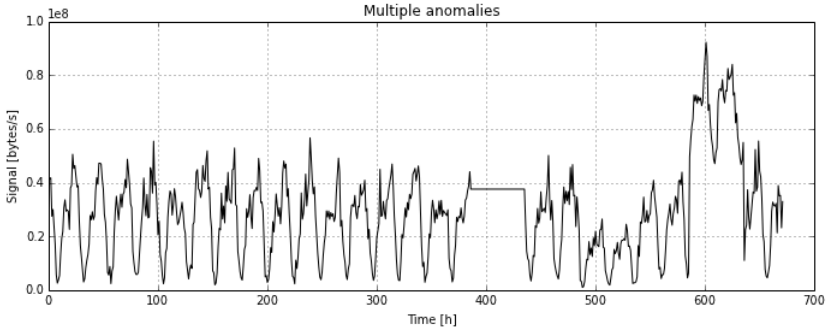

Figure 3.16: A signal that suffers from several different kinds of deviations. More explicitly, there is a static part that is considered anomalous in this context, a decrease and a sudden increase of data traffic.

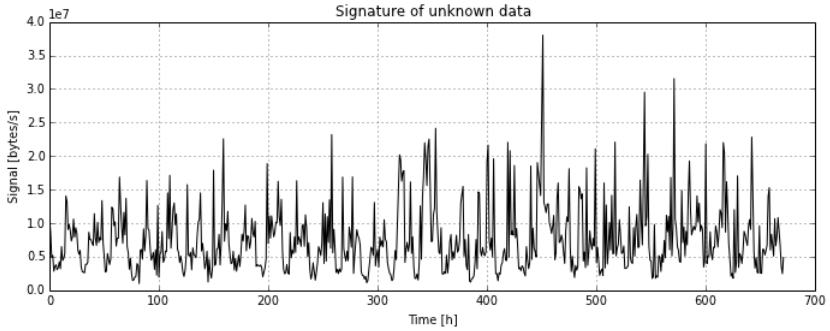

Figure 3.17: A signal that is collected from the signature called Unknown, which is a signal that is used as a catch-all for all unknown data that are found in the data traffic.

Chapter 4

Empirical Results

This chapter summarizes the data acquired from the three models. The chapter is divided into three sections that are dedicated to the results from the models. Extensive experimental results are provided in terms of plots to support the effectiveness of the proposed models. Because of the copious amounts of figures that could be supplied in this chapter, only a few cherry-picked figures are shown. Diagrams show complete overview of the results.

4.1 LPC Results

The outcome of the LPC-model is only the anomaly score, in which very many results can be observed quickly. However, in this paper, the outcome will also be provided in forms of plots for easier analysis. Two parameter combinations are examined in particular, namely, order 3 with window length 2 and also order 5 with window length 7. The reason for this is that the first is supposed to detect shorter and more temporary anomalies while the latter is implemented more for trend detection. The digital results for these are shown in Table 4.3 and 4.4. If there are more than one element in the anomaly score (Sec 3.8) it means that the signature is anomalous at one point, then back to normal and then anomalous again and so on. The filter length and the threshold value does also have an impact on the outcome, they are also surveyed, but separately. This is discussed further in Chapter 5.

The "Threshold indicator" indicates the difference between the largest deviation in the reference data set compared to the threshold, it is abbreviated as THI and is defined as in Definition 4.1. The purpose of the threshold indicator is to display how normal the reference data set is, where smaller values are normal (typically 0.5) and larger values are anomalous (above 1).

Definition 4.1. $THI = \frac{max(Y-D)}{THV}$, where Y is the LPC signal, D is the input signal (filtered) and THV is the threshold value.

The results from the Error Type test can be seen in Table 4.1 and 4.2 for

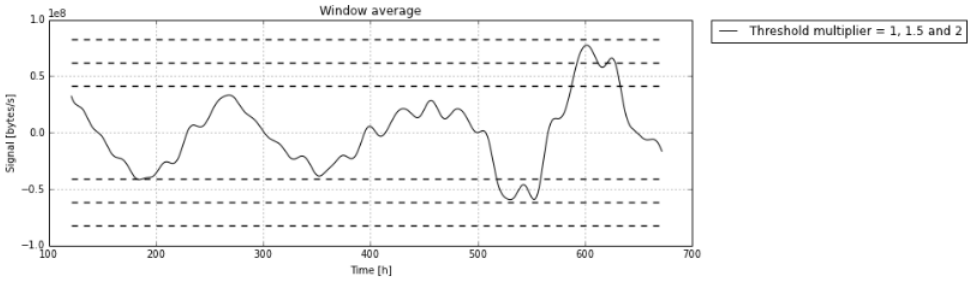

Figure 4.1: A window sum with three different threshold values.

short and long-term LPC respectively, where the threshold value is manipulated for different desires. A graphical demonstration for how this threshold multiplier affects the process can be seen in Figure 4.1.

Short-term Error Type LPC results

Anomaly type	Threshold multiplier	Average score	Error I	Error II
Point up	1	21	74 %	8 %
Point up	1.5	1.49	33 %	63 %
Point up	2	1.05	15 %	75 %
Trend up	1	13700	74 %	0 %
Trend up	1.5	6100	29 %	0 %
Trend up	2	3400	15 %	0 %
Trend up	10	145	0 %	0 %

Table 4.1: Overview for the results from the LPC-model with low order and low window length regarding Error Types. Threshold multiplier indicates how much the threshold is moved away from the nominal value.

Long-term Error Type LPC results

Anomaly type	Average score	Error I	Error I
Point up	2.7	20 %	27 %
Trend up	1550	20 %	0 %

Table 4.2: Results from the LPC-model with high order and long window length on several different signatures with artificial point -and trend anomaly up. Threshold multiplier indicates how much the threshold is moved away from the nominal value.

LPC outcome for order 3 days and window length 2 days

	For one signature (HTTP)	
Anomaly type	**Anomaly score**	**Threshold indicator**
$Point_u$	[1.65, 2.44]	0.26
$Point_d$	[466.99, 3.64, 2.56]	0.39
$Jump_u$	[1426.32, 3.52]	0.71
$Jump_d$	[1255.1, 2.47]	0.65
$Trend_u$	201.46	0.26
$Trend_d$	356	0.37
$Usage_u$	[72.8, 50.19]	0.26
$Usage_d$	0.73	0.26
$Periodicity_u$	0.53	0.18
$Periodicity_d$	0.76	0.46
$Frequency_u$	0.42	0.26
$Frequency_d$	0.91	0.26
Reference anomaly	30.7	26.87
Multiple anomalies	[3.89, 17.54, 115.53]	0.26
Normal data	0.42	0.26
Unknown data	[2.41, 2.06, 2.13]	0.51

Table 4.3: Overview for the results from the LPC-model with low order and low window length. The index in the anomaly type column indicates up or down pointing anomaly. Anomaly score below 1 means that there is no anomaly.

4.1.1 LPC-plots

All artificial anomalies are shown as input value (the normal data set is included here). However, the order of the LPC-transformation and the window length are chosen so that the most interesting result can be seen.

All the artificial anomalies are used as input in the LPC method and the outcomes are presented in this section (in terms of plots). They are visible in Figure 4.2 through 4.23. They are presented in the same order as they were created in Section 3.8.

LPC outcome for order 5 days and window length 7 days

	For one signature (HTTP)	
Anomaly type	**Anomaly score**	**Threshold indicator**
$Point_u$	0.61	0.20
$Point_d$	0.84	0.52
$Jump_u$	1206.45	0.26
$Jump_d$	2820.04	0.37
$Trend_u$	1527.42	0.20
$Trend_d$	7803.21	0.56
$Usage_u$	881.28	0.20
$Usage_d$	61.45	0.20
$Periodicity_u$	0.62	0.17
$Periodicity_d$	0.43	0.20
$Frequency_u$	0.69	0.19
$Frequency_d$	0.83	0.19
Reference anomaly	[36.23, 43.5]	1.62
Multiple anomalies	[10.26, 24.72]	0.20
Normal data	0.33	0.20
Unknown data	[3.55, 7.34, 26.89, 6.51]	0.39

Table 4.4: Overview for the results from the LPC-model with high order and high window length. The index in the anomaly type column indicates up or down pointing anomaly. Anomaly score below 1 means that there is no anomaly.

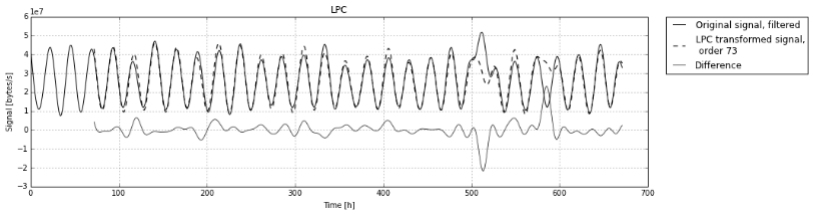

Figure 4.2: The outcome of the LPC method when a point suddenly increases a lot {1}. When the deviating point occurs, the generated signal continues with what it considered to be expected, but since this was not the case, the difference indicates a jump. The corresponding window sum of the LPC method can be seen in Figure 4.3.

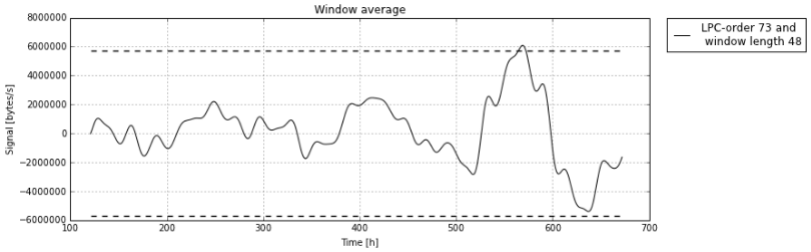

Figure 4.3: The corresponding window sum from the LPC method in Figure 4.2

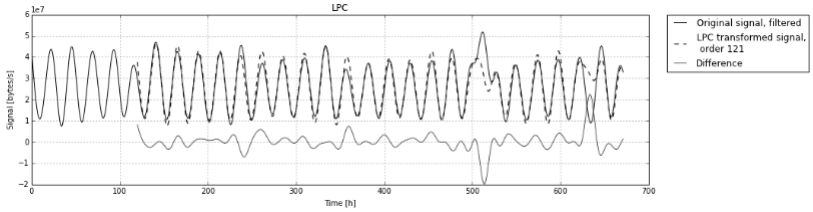

Figure 4.4: The outcome of the LPC method when a point suddenly increases a lot {1}. When the deviating point occurs, the generated signal continues with what it considered to be expected, but since this was not the case, the difference indicates a jump. The corresponding window sum of the LPC method can be seen in Figure 4.5.

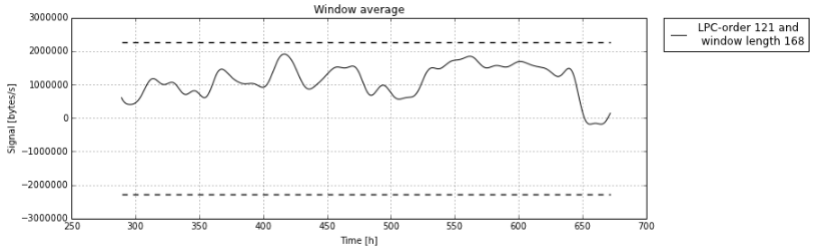

Figure 4.5: The corresponding window sum from the LPC method, with the same parameters, that is shown in Figure 4.4

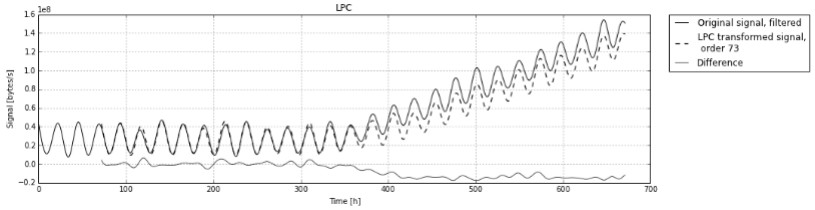

Figure 4.6: The outcome of the LPC method when a trend suddenly increases {3}. When the trend is increasing the generated signal follows, but slightly slower, hence, the difference is below zero. The corresponding window sum of the LPC method can be seen in Figure 4.7.

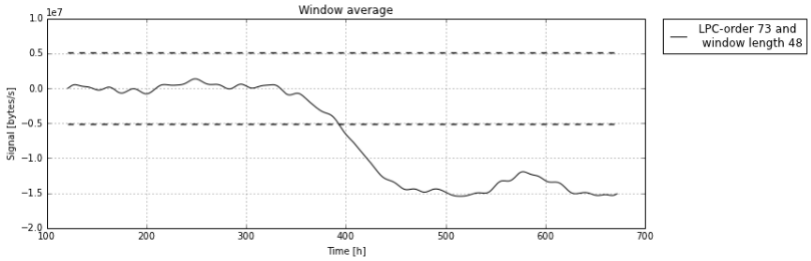

Figure 4.7: The corresponding window sum of the LPC method with the same parameters as in Figure 4.6. When the trend is increasing the generated signal follows, but lowered slightly, hence, the difference is below zero.

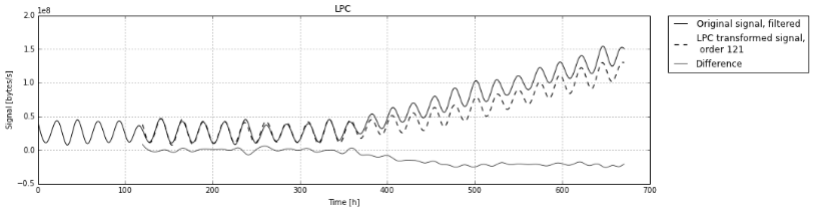

Figure 4.8: The outcome of the LPC method when a trend suddenly increases {3}. When the trend is increasing the generated signal follows but slightly slower, hence, the difference is below zero. The corresponding window sum of the LPC method can be seen in Figure 4.9.

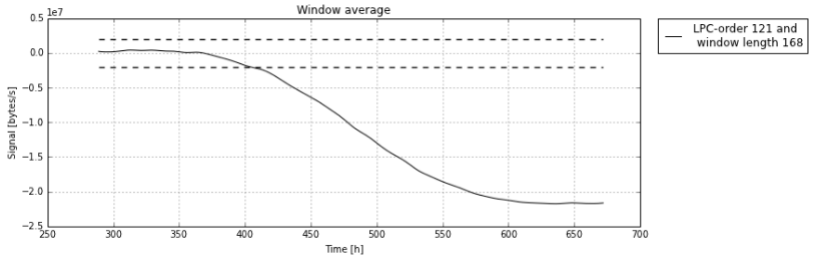

Figure 4.9: The corresponding window sum of the LPC method with the same parameters as in Figure 4.8. When the trend is increasing the generated signal follows but lowered slightly, hence, the difference is below zero.

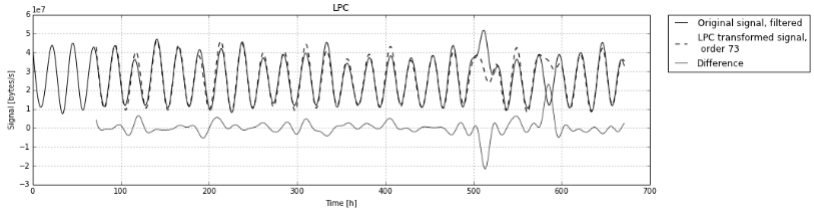

Figure 4.10: The outcome of the LPC method when an up-pointing point anomaly {1} suddenly appears for low LPC order and short window length. When the anomaly occurs the model does not expect this and it creates a deviation in the difference between the signals.

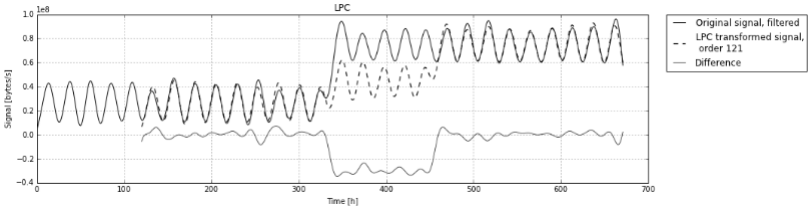

Figure 4.11: The outcome of the LPC method for jump trend anomaly {2} and for high order and long window length. When the trend is increasing the generated signal follows but slightly slower, hence, the difference is below zero.

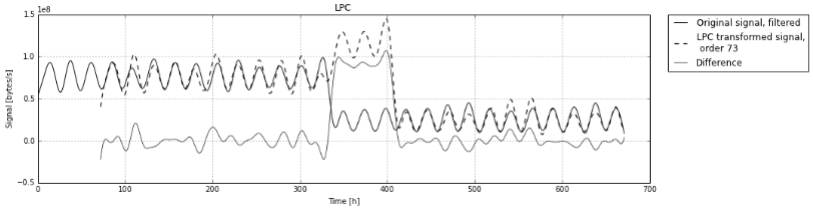

Figure 4.12: The outcome of the short-term LPC model when there is a jump anomaly {2} in the signal.

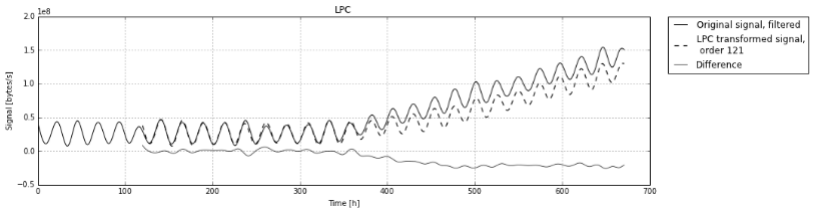

Figure 4.13: The outcome of the long-term LPC model when a slowly increasing trend {3} is implemented. When the trend is decreasing the generated signal follows but slightly slower, hence, the difference is slightly above zero during the trend.

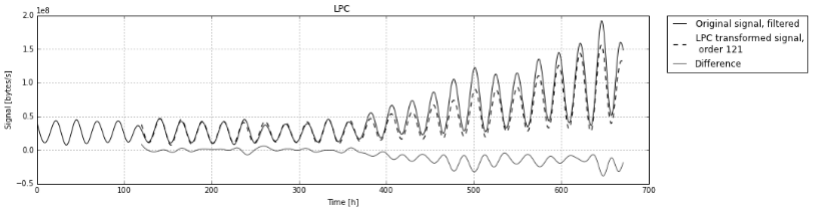

Figure 4.14: The outcome of the long-term LPC model when the usage suddenly starts to increase {4}. When the amplitude for each day is increasing day-by-day, more noise can be visible in the difference signal.

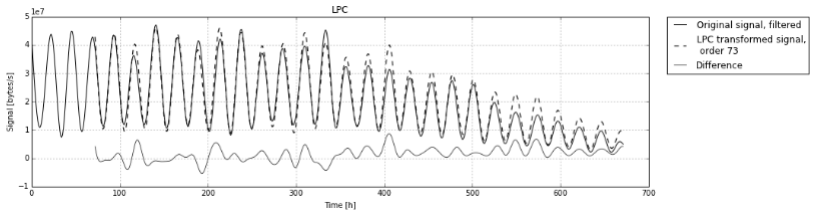

Figure 4.15: The outcome of the LPC method when the periodicity suddenly decreases {4}. When the amplitude for each day is decreasing day-by-day, the difference signal becomes even smaller and the threshold remains, therefore, the anomaly score will most likely be low.

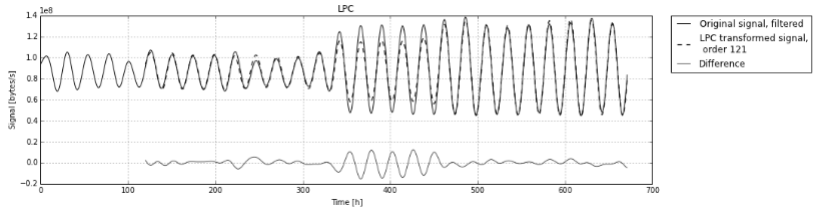

Figure 4.16: The outcome of the long-term LPC method when the periodicity suddenly increases {5}. When the amplitude of the periods are increasing the generated signal takes some time to adjust to this. Noise is visible during this adjustment.

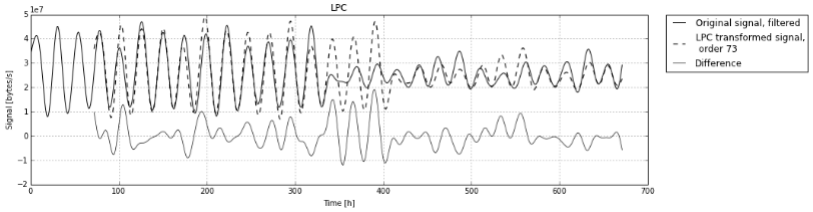

Figure 4.17: The outcome of the short-term LPC method when the periodicity suddenly decreases {5}.

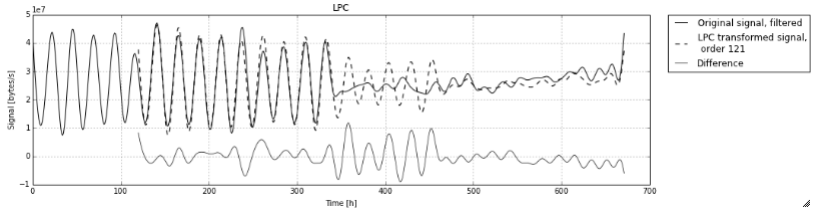

Figure 4.18: The outcome of the LPC method when the frequency suddenly increases {6}. The signal has a frequency that is too high to pass the filter.

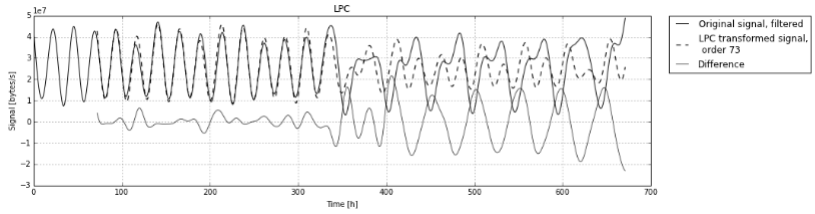

Figure 4.19: The outcome of the short-term LPC model when the frequency suddenly decreases {6}.

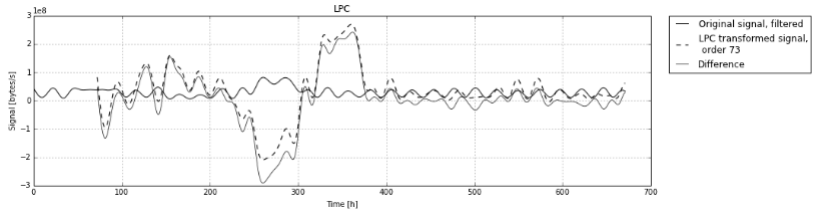

Figure 4.20: When the reference is anomalous {7}, the model behaves very unusual.

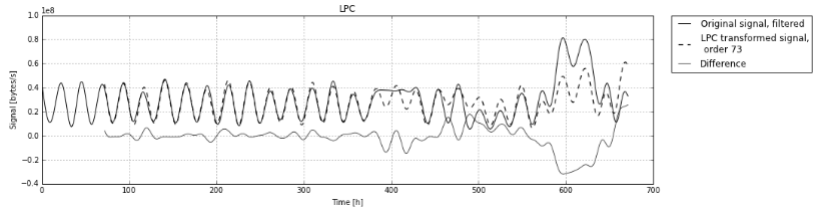

Figure 4.21: The outcome of the short-term LPC model when the multiple amount of anomalies {8} are located in the signal.

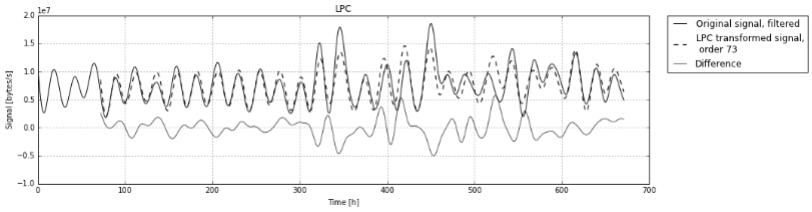

Figure 4.22: The outcome of the short-term LPC model for the signal: Unknown {9}. This signal has no anomalies artificially added.

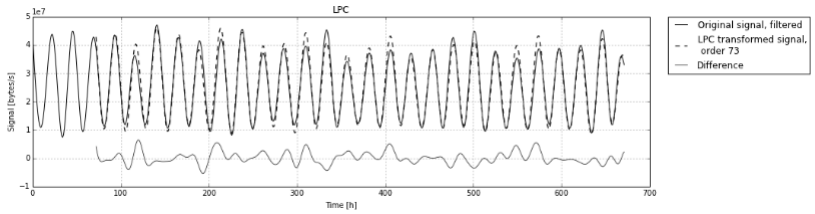

Figure 4.23: The outcome of the short-term LPC model for the original signal.

4.2 Sparse LP Results

The input values for SLP are: order, window length, anomaly type and the sparsity parameter γ, this results in even more plots than for the LPC model. However, these are omitted due to too much similarity to the LPC results. The reason for this is that for reference data sets of length of one week or less, the SLP model creates no zero-element in the α for this task. Furthermore, when γ reaches large values (e.g. 10^8), the system starts to converge to zero and all α becomes zero. This is for reference data set containing two weeks of data.

4.3 Wavelet Results

The wavelet outcome consists of an anomaly score too. A complete overview of the outcome is presented in Table 4.5 and the most interesting findings are visualized in terms of plots in Section 4.4.

The threshold value is different for the wavelet model, see Definition 3.6.

The wavelet model is investigated in detail for two different anomalies, namely *Point anomaly up* and *Trend up*. The applied test is the same as for the test in Chapter 4.1 (9 different signatures at 11 different locations in time for each signature). The results for Type I Error are equal for both the point and the trend anomaly. This is natural considering both being created from the same signal. One anomaly were detected during normal signatures which results in one Type I Error. Furthermore, for the point anomaly case, the artificial anomaly that is applied to every signature is the mean value of the signal times 8 in two succeeding points. The outcome for the point anomaly is that all 88 anomalies were detected with the wavelet model. The average anomaly score is 15.88.

The result for the trend anomaly is that 1/88 anomalies were detected. The only anomaly that is detected has score 1.28 (average score is 0.0178).

4.4 Wavelet Plots

The outcome of the wavelet model is presented in terms of plots, see Figure 4.24 through 4.39.

Wavelet outcome

	One signature (HTTP)	88 different signatures	
Anomaly type	**Anomaly score**	**Error I**	**Error II**
$Point_u$	8.48	1/88	0/88
$Point_d$	6.91		
$Jump_u$	1.44		
$Jump_d$	1.25		
$Trend_u$	-	1/88	87/88
$Trend_d$	-		
$Usage_u$	[1.32, 1.7]		
$Usage_d$	-		
$Periodicity_u$	-		
$Periodicity_d$	-		
$Frequency_u$	-		
$Frequency_d$	-		
Reference anomaly	1.57		
Multiple anomalies	1.99		
Normal data	-		
Unknown data	1.27		

Table 4.5: Overview for the results from the wavelet-model. In the anomaly score column, if the value is "-" it means that no anomaly were detected. If an anomaly is detected, the score is printed.

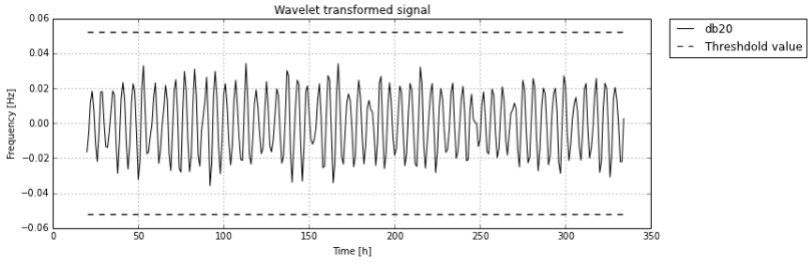

Figure 4.24: Wavelet transformed signal output for anomaly input: No anomaly

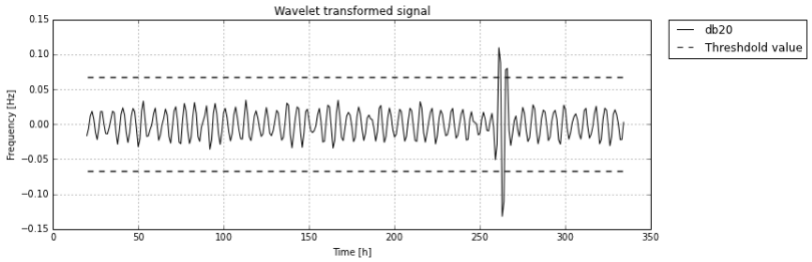

Figure 4.25: Wavelet transformed signal output for anomaly input: Point anomaly up

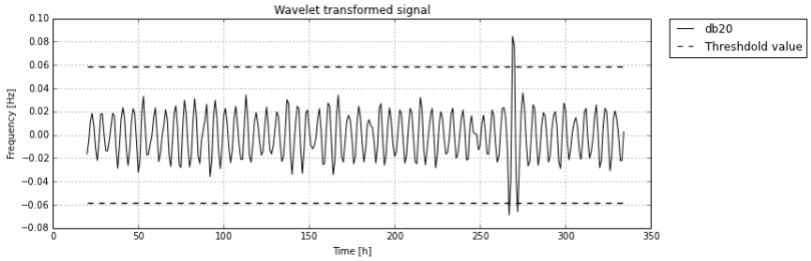

Figure 4.26: Wavelet transformed signal output for anomaly input: Point anomaly down

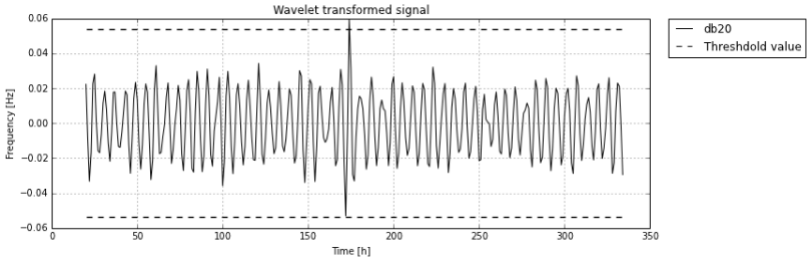

Figure 4.27: Wavelet transformed signal output for anomaly input: Trend-jump up

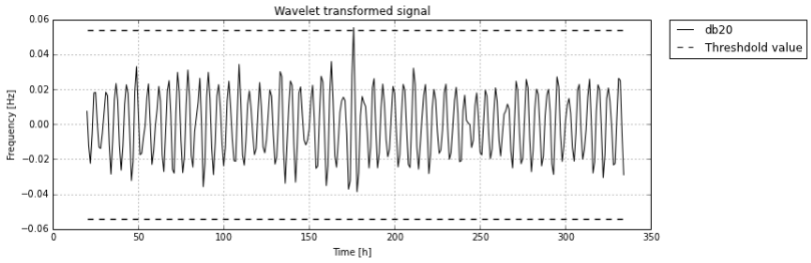

Figure 4.28: Wavelet transformed signal output for anomaly input: Trend-jump down

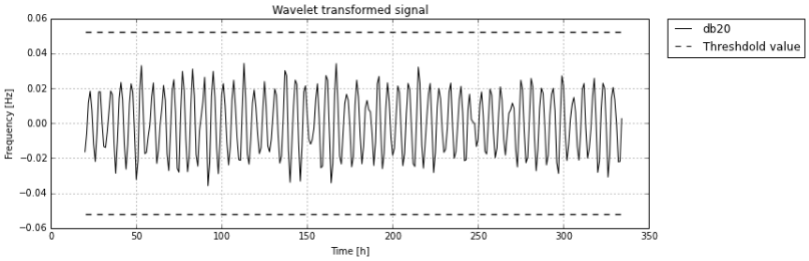

Figure 4.29: Wavelet transformed signal output for anomaly input: A trend changing direction up

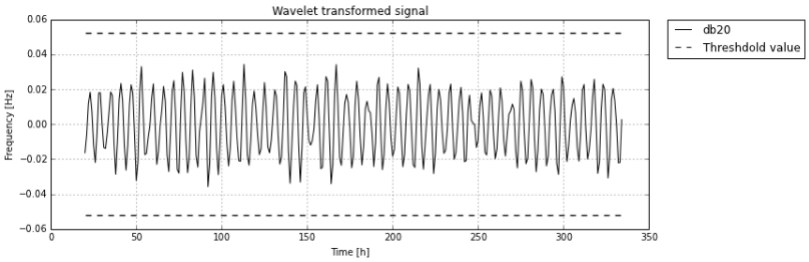

Figure 4.30: Wavelet transformed signal output for anomaly input: A trend changing direction down

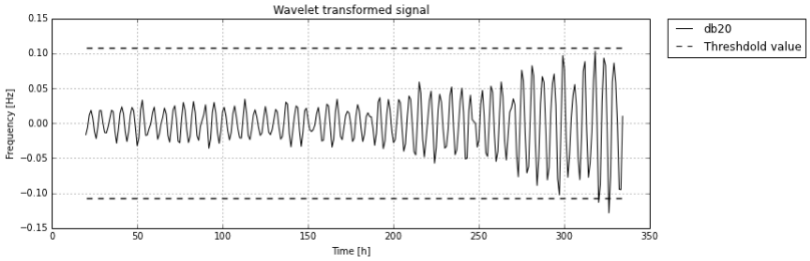

Figure 4.31: Wavelet transformed signal output for anomaly input: Change in usage up

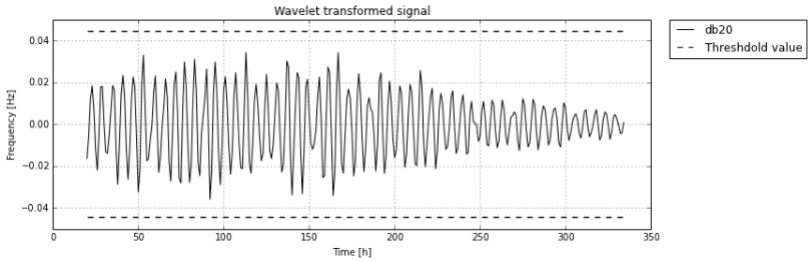

Figure 4.32: Wavelet transformed signal output for anomaly input: Change in usage down

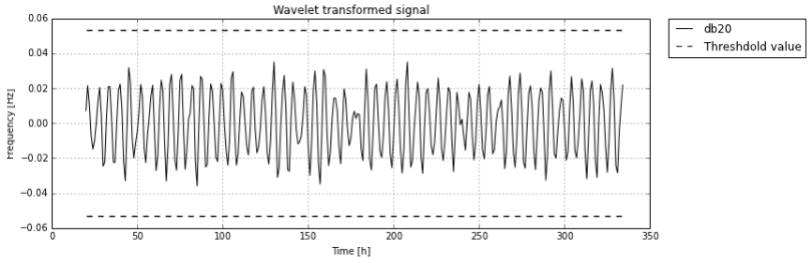

Figure 4.33: Wavelet transformed signal output for anomaly input: Sudden change in periodicity up

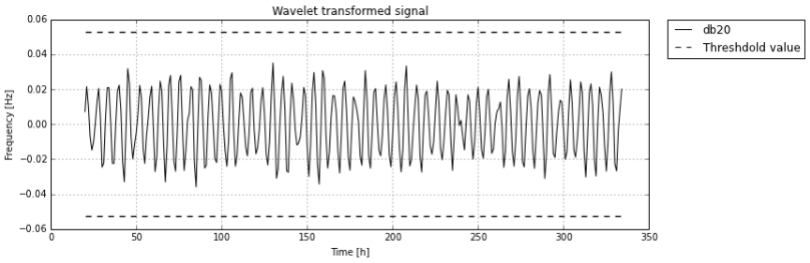

Figure 4.34: Wavelet transformed signal output for anomaly input: Sudden change in periodicity down

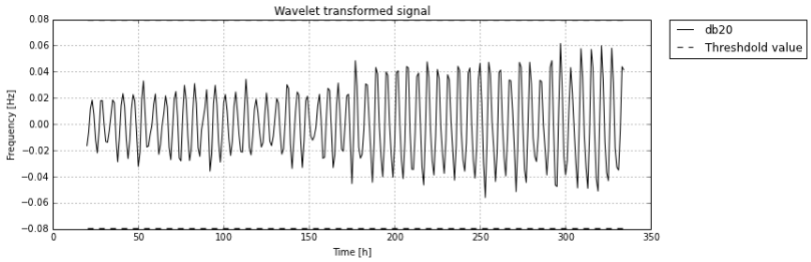

Figure 4.35: Wavelet transformed signal output for anomaly input: Changed frequency up

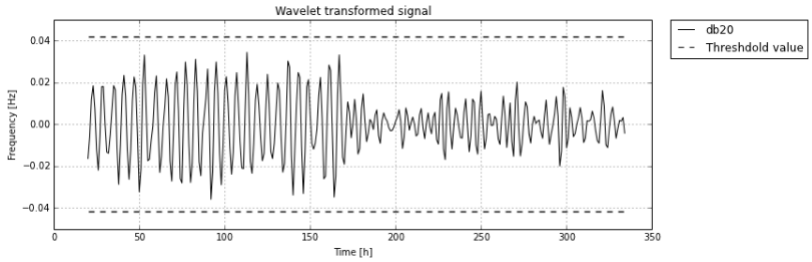

Figure 4.36: Wavelet transformed signal output for anomaly input: Changed frequency down

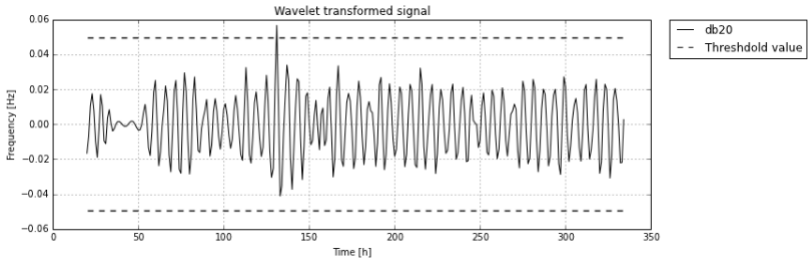

Figure 4.37: Wavelet transformed signal output for anomaly input: Anomaly located in the reference data

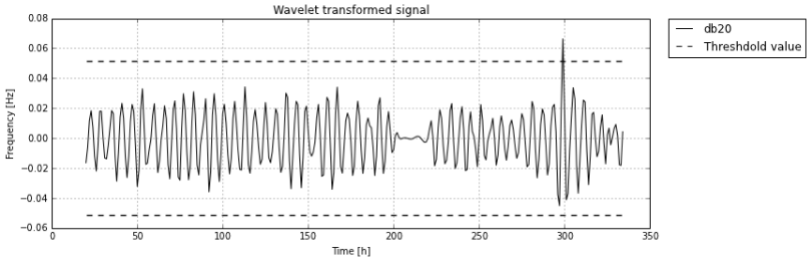

Figure 4.38: Wavelet transformed signal output for anomaly input: Multiple anomalies

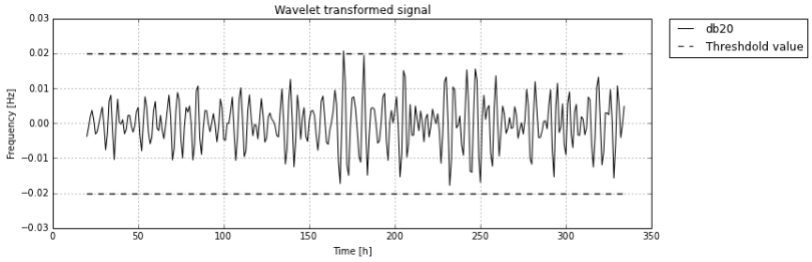

Figure 4.39: Wavelet transformed signal output for anomaly input: Signature unknown

Chapter 5

Analysis and Discussion

The beginning of this chapter answers the research questions. Furthermore, deeper analysis, thoughts about anomaly detection and the three methods are presented. Finally, the implementation is presented together with future work.

5.1 Research Question Analysis

Answers to the research questions are presented below.

5.1.1 R1: What method is most suitable for detecting anomalies?

Generally, by comparing the results, point anomalies are easily detected by the wavelet model and trends are easily detected with LPC with high order and high window sum. This is reasonable too due to the fact that the wavelet model observes the current frequency, which will change drastically in a point anomaly. For trends, the frequency can even be normal during the entire anomaly, but a bit offset, therefore, it will only be detected by observing a larger summation of the error. Hence, the answer to **[R1.]** (research question from Section 1.7) is that both are good, one being better than the other depending on the anomaly.

5.1.2 R2: What kind of deviations should be considered as anomalies?

Mainly, there are two different kinds of anomalies, namely, short and long (point anomalies and trend anomalies respectively). The answer to **[R2.]** is that these are the kind of deviation that are considered as anomalies. How they are detected is described further in Section 5.2 and 5.3.

5.1.3 R3: Which parameters are best to maintain and which to disregard?

Considering the parameters, they sure do have an impact on the results but not as much impact as they could considering their variation. E.g. the order can vary from 1 hour up to 168 hours, but only the values for full days are useful, therefore, 161 of them can be discarded. With only $[24, 48, ..., 168]$ left, still some of these provide almost equal results. This means that even more parameters can be discarded, but now they can be discarded by what is most beneficial for the user. In this paper, only 72 and 120 (order 3 and 5) are retained. The same argument holds for the window length parameter, where the final lengths are 48 and 168 (order 2 and 7). That is the answer to **[R3.]**.

5.1.4 R4: How will the outcome be scored?

The anomaly score resembles the integral of the curve that are located outside of the threshold line. Analyzing this mathematically is done by using Definition 3.8 and is the answer to **[R4.]**. If this value is above 1 it is an anomalous point (or sequence). The larger the value is, the more critical the anomaly is. The scoring is done empirically. This works well for most of the anomalies in this project, but when extremely anomalous points enters the model it can get out of hand and result in an extremely large score. This can be misleading, but since all extreme anomalies must nevertheless be analyzed, it is a good thing that they are highlighted.

5.1.5 R5: What level of certainity is it on the models?

In statistics, 100 % certainty level is very difficult to achieve. For these three models, by analysing the rate of Error Type 1 & 2 on the artificial anomalies, the success rate for detecting anomalies are shown in Table 5.1, which is created by simply looking at the ratio regarding number of Type II Errors and total number of anomalies. Generalization of this yields a certainty of 50 to 60 % detected anomalies, which is the answer to **[R5.]**.

These numbers can be greatly improved if the models are used simultaneously. For instance, if the wavelet model is used for detecting short-term anomalies only, it presents a clear difference between normal data and anomalous data. The long-term LPC model is used for slower anomalies, such as trends, where it showed a very high anomaly score for the trend anomalies. Worth noting is that only one Type I Error occurred throughout all tests but many more Type II Error. For the latter error type a brief analysis has to be done by a person, but for the first error type nothing will be done and a possibly devastating problem will go unnoticed. Therefore, Type II Error is considered more hazardous than Type I Error. And since Error II is more common for these models, parameter tuning is added to future work to decrease the number of errors of the second type. This is done in the last row in Table 5.1, where the wavelet model handles the local anomalies and the long-term LPC model handles the global anomalies.

Model overview results

Model	Anomalies detected	Error I	Error II
Short-term LPC	57 %	74 %	40 %
Long-term LPC	57 %	20 %	13 %
Wavelet	50 %	1 %	49 %
Wavelet and long-term LPC combination*	$\sim 71\%$	~ 11 %	~ 0 %

*The result for the combination of models is only an estimation.

Table 5.1: An overview of the average for the LPC and the wavelet results.

However, the results for the combination of models are estimations based on the results for the other models, s.t. the results for the local anomalies are taken from the wavelet model and the global results are taken from the long-term LPC model.

The anomaly detection rate is also a bit misleading, for the implemented anomalies are all tested once, which represents a world where all types of problems occur once and none occurs frequently. Realistically, that is not the case since some problems are more common than others, e.g. hasty software updates or local server crashes. If a frequently repeating error is an easy one to detect, the anomaly detection rate would increase (and vice versa). Nevertheless, more common problems are more likely to be focused on, therefore, the detection rate should increases with more distinct anomaly type distribution.

5.2 Detecting Point Anomalies

When a point anomaly appears {1}, the filter will smooth out this and the deviating point will not even be noted, unless it is deviating very much (such that $d_i >> E[D]$ or $d_i << E[D]$). However, the filter can be regulated by decreasing itâĂŹs length so that it is more sensitive to deviations in one point. The forward-backward filter that is used has a length of eight steps (eight hours) so it is not very sensitive. If the desire is to detect such anomalies, the filter length should be lowered to, e.g. three. This will cause the signal to oscillate more. Eventually, this might make other anomalies more difficult to detect. The reason for this is that the reference data set will be less consistent and, e.g. the α-parameter for the LPC method will become more complex.

In general, all methods performed satisfactorily to detect point anomalies but the wavelet model is the one with the best results for this.

5.3 Detecting Trends

A trend jump {2} has an outcome from the LPC that looks like a delay. The LPC-signal needs some time to adjust to this sudden jump and while the LPC-signal is doing that, the difference between the input signal and the LPC-signal is very large. By only considering the absolute value of the difference, a large block appears at the time of the anomaly. Looking at the absolute value is done because one more task will be eliminated, namely, to determine whether the trend jumps up or down (both are however anomalies).

In the changing trend-case {3}, the error is very small, hence, it will not be considered as an anomaly in the difference comparison between the LPC signal and the input signal. On the other hand, the error is constantly located at one side of the zero mark. Benefits can be drawn from this, namely, if there is no absolute operator and the window length for the error summing is increased. Such small but persistent deviations are added with the same sign, hence, a sum of a large number of elements with the same sign will make the value of the sum large, even though the elements themselves are not. Therefore, a changing trend can be detected by summing over a longer time period, e.g. seven days or ten days.

Slow trends were not even detected by some models, but the long-term LPC model with long window length performs best for this.

5.4 LPC Analysis

The definition of a good model is a model that has as few Type I & II Errors as possible. In Table 4.1 it can be shown that the LPC model is not completely enough for the local anomaly detection task. If the threshold is set to a narrow value, the Type I Error increases immensely. To prevent this, the threshold is set to a more spacious value, which in turn makes Type II Error increase significantly. A compromise is necessary and a decision must be made about what is desired.

For the slower anomalies, such as trends, the long-term LPC model is a good choice. It detects the majority of the anomalies with few Type I & II Errors and high margin for the threshold indicator.

5.5 SLP Analysis

In the analysis of the shorter reference data set (one week) the SLP model cannot improve α to make it more sparse. Only at very large γ the SLP model increases the number of zero-elements in α. But in fact, the model now consider a zero-element more valuable than almost any number, therefore, it converges to generate an α containing only zeroes because having a zero in an element is more appraised than having a number that possibly could make the SLP signal to fit the input signal.

For reference data sets of length of one week or less, the SLP model creates no zero-element in α until γ reaches 10^8 where the system starts to converge to zero.

What happens in the "grey area" in between one and ten weeks is not investigated in this paper and therefore added to the Future work, see Section 5.8.

5.6 Wavelet Analysis

Wavelets are great for detecting local anomalies, almost no Type I & II Errors at all, see Table 4.5. However, this model is terrible for detecting trends. A slowly moving trend does not change the frequency much, therefore, it is simply not being detected by the model. Resulting in a hideous amount of Type II Error for trend anomalies. The only detected trend resulted in a anomaly score of 1.28, which is not very convincing.

Anomalies directed downwards are resulting in lower frequencies. This means that if the signal is normal from the start, the anomalies are considered to be even more normal (because they are located even more in between the thresholds). This is actually misleading, because anomalies directed downwards are considered equally anomalous as upwards. A solution to this could be to introduce another threshold for when the signal starts decrease. This problem occurs when a signature suffers from decreasing usage over time (down-pointing trend). At this time, any change in the signature have more difficulties to be detected by the system. The reason for this is that the wavelet outcome itself is located much lower than before, regarding the threshold. E.g. a decreasing trend followed by an up-pointing point anomaly is more difficult to be detected, see the filtered input signal in Figure 5.1 and the outcome in Figure 5.2.

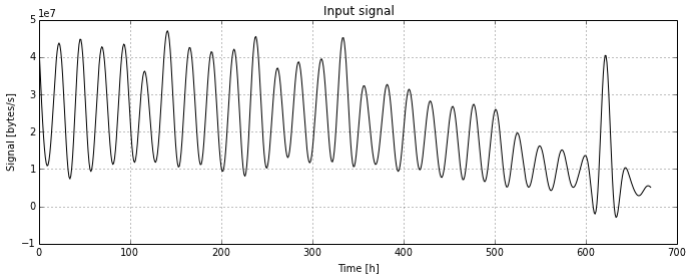

Figure 5.1: A late point anomaly in an already anomalous down-pointing trend.

5.6.1 Window Sum

A window sum for wavelet outcome is not reasonable because the outcome is located around the zero-mark. Therefore, the sum will also oscillate around zero. The problem is solved by taking the sum of the absolute values of the

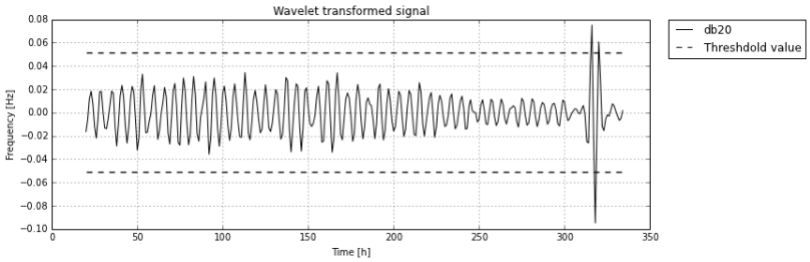

Figure 5.2: The output signal from the signal in Figure 5.1 using wavelet (Daubechies wavelet).

outcome. Hence, the sum would increase for increasing frequencies and decrease for decreasing frequencies.

5.7 Implementation of Methods

The current working process for the anomaly detection program is that a data set (D_{ref}) is sent to the machine so that it can learn from this. This data set is supposed to be as normal as possible, containing no anomalies. If any of these signatures are already anomalous, the program will not perform very well on this particular signature, see Figure 4.20, resulting in many alarms and the reference set must be replaced with a normal data set. From this point, anomalies can be detected. When the analyzed data set (D_{test}) is analyzed and an anomaly are detected for any signature, the anomaly score will indicate this (values above 1 are anomalous). The higher the score is the more anomalous the signature is.

The program is done calculating in matter of seconds, which makes Problem specification 4 a success! And no further code optimization is required.

5.8 Future Work

- *Thresholds* has optimization opportunities. A more general and simple function could be a solution.
- *Anomaly score* has optimization opportunities too. The algorithms find them but the score is sometimes a bit misleading. However, this is correlated with good models.
- *Analyze the input parameter*, upload speed and download speed that are reduced to a collective parameter. Separate them and analyze them individually instead and see if the result differs.
- *The wavelet families*, other than Daubechies, can be closer analyzed to see if any of them performs perhaps better for its specific purpose.

- *The SLP model* can be closer analyzed to see what happens with the output for a larger variety of γs.

Bibliography

[1] Wikipedia, 2014. *Enterprises*. [Internet]
Available at: *http://en.wikipedia.org/wiki/Small_and_medium-sized_enterprises*
[Accessed 26 November 2014]

[2] Hotrakool, W., 2012. *l0-Norm, l1-Norm, l2-Norm, ..., l-infinity Norm*. [Internet]
Available at: *https://rorasa.wordpress.com/2012/05/13/l0-norm-l1-norm-l2-norm-l-infinity-norm/*
[Accessed 20 December 2014]

[3] MathWorks, 2014. *Butterworth filter design*. [Internet]
Available at: *http://se.mathworks.com/help/signal/ref/butter.html*
[Accessed 3 February 2015]

[4] MathWorks, 2014. *Linear Predictive Coding coefficients*. [Internet]
Available at: *http://se.mathworks.com/help/signal/ref/lpc.html*
[Accessed 3 February 2015]

[5] Holdt Jensen, S. Lindstrøm Jensen, T., 2013. *Sparse Linear Prediction framework*. [Internet]
Available at: *http://sparsesampling.com/sparse_lp/*
[Accessed 10 Mars 2015]

[6] Polikar, R., 2014. *The Wavelet Tutorial*. [PDF]
Available at: *http://web.iitd.ac.in/ sumeet/WaveletTutorial.pdf*
[Accessed 4 February 2015]

[7] Richert, W. Coelho, L.P., 2013, *Building Machine Learning Systems with Python*, 1^{st}. Birmingham: Packt Publishing.

TRITA -MAT-E 2015:46
ISRN -KTH/MAT/E--15/46--SE