



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

NUMERIKUS ANALÍZIS

TANSZÉK

Wavelet transform based model driven neural networks

TDK Thesis

Supervisor:

Dózsa Tamás Gábor

Doktorandusz,

Programtervező Informatikus MSc

Author:

Ámon Attila Miklós

Programtervező Informatikus MSc

Budapest, 2024

Contents

1	Introduction	2
2	Mathematical background	6
2.1	Data driven methods and supervised learning	6
2.2	Model based machine learning	10
2.2.1	Variable projection operators	11
2.2.2	VP-Net – variable projection networks	13
2.3	Continuous wavelet transform	14
3	CWT-VP and CWFCNN	19
3.1	Architecture of the neural networks	19
3.2	CWT-VP layer	21
3.3	CWFCNN layer	22
3.4	WKN - comparison to the state of the art	23
4	Experiments	25
4.1	Open source bearing fault database	25
4.1.1	The characteristics and diagnosis of bearing fault signals . . .	25
4.1.2	Experiment setup and results on the MFPT challange dataset	27
4.2	MIT-BIH ECG database	30
4.2.1	Description of the database and test results	30
4.2.2	Interpretability of the learned parameters	33
4.3	SIEMENS bearing fault dataset	36
5	Conclusion	38
	Bibliography	39

Chapter 1

Introduction

In this thesis, we present novel machine learning algorithms that use continuous wavelet transform coefficients to produce a low dimensional representation of the input signal with human-interpretable parameters. The latter property is particularly important for our choice of applications and experiments, such as failure prediction of mechanical parts of a machinery.

In the development of modern signal processing algorithms, we aim to use as much apriori knowledge as possible when creating our method, i.e. to solve the specified problem using physical models or mathematical transformations [1]. These classical methods have the advantage that the parameters of the transformations or the differential equations describing physical phenomena can be interpreted. These classical models can be implemented efficiently, with a good example being the Fast Fourier Transform (FFT) [1]. Unfortunately, constructing such methods for problems displaying highly nonlinear is difficult. Most classical modeling approaches assume stationary signals, i.e. the information in the spectrum does not change in time, but this is rarely the case in real life examples. [1]

A common signal processing problem is to classify input signals according to the properties of the phenomenon under study. In engineering, change and fault detection problems can usually be framed as classification problems - for instance, recognizing various part failures based on sensor data [2] may be posed as a binary classification task.

Most practical problems can only be described using highly nonlinear operators and non stationary measured signals. In such cases, machine learning (ML) methods

have been successfully used to solve a variety of classification problems [3]. The success of ML methods is rooted in their data-driven nature: instead of relying on pre-defined mathematical models, these approaches aim to model the studied phenomenon purely based on obtained data. In general, artificial intelligence can solve problems where programs that mimic human thinking have an advantage over traditional solutions. [4] A popular classification ML algorithm is the neural network (NN), which, in its simplest form can be interpreted as a function composition of adaptive linear mappings together with nonlinear so-called activation functions [4]. ML algorithms traditionally require large amounts of input data to develop a trustworthy model of the problem.

In ML methods, in addition to having the right amount of data, it is essential to find the right representation of the input to increase the performance of our model. To this end, the input signals are usually subjected to so-called feature extraction transformations. For example, it may be useful to first normalize the signals [4]. It is vital that these procedures are done with forethought, because it is not always clear which transformations improve the performance of the underlying ML method.

Afterwards, our goal is usually to reduce the dimension of signals and assign some physical meaning to a property of the representation. This can be done statically, in which case the input to the ML algorithm will be the transformed data. Popular feature extraction techniques include PCA [4] and the discrete Fourier transform (DFT) (see e.g. [1]). Traditionally, the correct feature extraction transformation was selected by hand based on apriori information about the problem. Similarly to the signal processing methods proposed in this thesis, this approach to feature extraction allowed for the interpretability of parameters. Unfortunately, this way of obtaining useful features from the data largely relies on a heuristics and has been shown to provide inferior results to modern, end-to-end ML methods [3].

If the parameters of the feature extraction are learned together with the weights of the ML algorithm (automatic feature extraction), an optimized representation of the input is obtained. An example of such automatic feature extraction can be convolutional neural networks (CNN). The output of the CNN layers performing discrete convolution determines the input of the underlying classification algorithm (NN), so that the weights are learned together with the signal representation.

Application areas like automotive or medical signal processing require, the explainability of the parameters of the trained ML algorithms (explainable artificial intelligence - XAI). For example, for a traditional CNN, the learned kernel functions are difficult to explain. In critical situations, for instance deciding in real time whether to brake a self-driving car, interpretability is a necessary aspect.

By combining data-driven solutions with model-based approaches, we arrive at model-based ML algorithms. For example, a recent advancement in model-driven learning was the introduction of VP-NET [5]. This kind of ML model obtains an optimal representation of the input signals by solving a so-called separable nonlinear least squares (SNLLS) problem in the first layer of the network. If this problem is posed in an informed manner, the learned parameters (as well as the obtained signal representations) can have exact physical meaning [6], [7].

Since their introduction, this idea has been generalized to other ML models, such as the SVM classifier [3]. It is important to mention, that in addition to increased transparency of the learned parameters, variable projection based models often reduce computation complexity when compared to equivalent convolution based approaches. This allows real-time implementation even on low-power target computers, such as microcontrollers or FPGAs (field programmable gate arrays) [3]. In the field of automotive or medical signal processing, this could be an important requirement for lower production costs and a more future-proof use of energy resources.

In the 1980s, wavelet analysis opened a new chapter in signal and image processing. Similarly to the time-frequency representation of the short-time Fourier transform (STFT or Gabor-transform), a time-scale representation of the signal is obtained in the case of wavelet transforms. The various types of wavelet transforms - continuous wavelet transform (CWT) [1], discrete wavelet transform (DWT) [1], etc. - can be used to obtain useful time-localized spectral information about the input. In this way, the wavelet transform provides interpretable information about signal behavior. Thus, wavelet representations have become popular in the processing of various non-stationary signals [8].

Frequency space analysis of signals (relying on mostly the Fourier-transform) has been the basis of many industrial applications. Some vibrations are not constant, so it is important to know at which moment in time they appear in the signal, for

example, periodic frequencies in rotating parts (brake discs, rolling bearings).

Thus, when analyzing such non-stationary phenomena, CWT can provide useful features for a model-based ML algorithm. As a static feature extraction procedure, using the matrix of the coefficients as input, greatly increases the computational power required to solve the problem, since, for example, a 1-D vector classification must be considered as a 2-D image processing procedure. With the development of Deep Learning techniques and high-performance GPUs, this is now an offline solvable problem [8], but it is not applicable in real-time on a low-cost and low-power microcontroller.

In order to invert the transformation, CWT relies on a redundant representation of the signal. For most classification problems however, inverting the wavelet transformation is not necessary. Thus, a sparse signal representation may be obtained by only considering wavelet coefficients which can be used to distinguish between the different signal classes.

In the next chapters of this thesis, we present neural network models whose first layer numerically approximates the wavelet coefficients using variable projection (CWT-VP) [5] and quadrature formulas (CWFCNN). The performance of the proposed models is verified on various benchmark databases corresponding to engineering and medical signal processing problems. We confirm, that the proposed wavelet coefficient based learning methods match (and in some cases outperform) the state-of-the-art, while retaining a simple and interpretable architecture. The proposed models are also qualitatively compared with another CWT based ML method published in [8].

The rest of this thesis is structured as follows. In chapter 2 mathematical concepts necessary for the proposed methods discussed. In chapter 3 we introduce the novel CWT based neural network architectures. Chapter 4 details the conducted experiments and provides a discussion of the results. Finally, in chapter 5 a conclusion is given and future research directions are proposed.

Chapter 2

Mathematical background

In this section we review the mathematical concepts required for the proposed methods. In particular, we consider a formalization of supervised learning problems, review model based learning schemes and variable projection operators. In addition, we provide a review of the continuous wavelet transform and related quadrature formulas.

2.1 Data driven methods and supervised learning

In classical signal processing signals are represented using heuristic models (e.g. differential equations, time-frequency transformations, DFT, etc.) created for the problem. Choosing the right transformation for a given problem requires specialized expertise, while nonlinear, non-stationary processes are usually difficult to handle in this way. When the model describing the phenomenon under study is built automatically, based solely on measurement data, it is called a data-driven method. Data-driven methods, on the other hand can model nonlinear, non-stationary problems well and do not require apriori knowledge or heuristics typical of classical methods.

In this thesis, we deal with so-called supervised learning (SL) problems in particular. In SL procedures, the core concept involves approximating a mapping using a subset of data for which the values of that mapping are known. By doing so, we aim to accurately describe the mapping's behavior across the entire dataset. Consider

the operator

$$F : \mathcal{X} \rightarrow \mathcal{Y}, \quad (2.1)$$

where $\mathcal{X}$ is a compact metric space containing the input measurements (data). In the case of signal processing tasks, usually $\mathcal{X} = \mathbb{R}^{M \times N}$, where N denotes the sample points of the signal segment (float data vector), and M denotes the number of channels we use. For example, if the input consists of equal length acceleration recordings obtained simultaneously from four different sensors, we would have $\mathcal{X} = \mathbb{R}^{4 \times N}$.

In this thesis we assume that the target set is $\mathcal{Y} \subset \mathbb{R}^N$ ($N \in \mathbb{N}$) or $\mathcal{Y} := \{0, 1, \dots, N\}$ ($N \in \mathbb{N}$). Approximating F in the first case is known as a "regression", while in the second case it is referred to as a "classification" problem [4]. Regression tasks aim to approximate targets that have continuous values. In the case of classification problems, targets are contained in a finite, discrete set. The number of classes (the cardinality of the set $\mathcal{Y}$) may vary. In this thesis, the proposed signal processing methods are used to solve certain fault detection problems in engineering (see chapter 4). These kinds of problems can often be posed as binary classification tasks [2] with $\mathcal{Y} := \{0, 1\}$.

In the case of a classification or regression task, the process of finding the parameters of the model that best fits the data is called training or learning [4]. During the training process, we want to find a parameterized model whose distance to F , measured by some metric, is minimized. Model parameters are usually identified by gradient based optimization schemes.

Formally, the parameterized model of F is an operator

$$G_\theta : \mathcal{X} \rightarrow \mathcal{Y} \quad (2.2)$$

where $\theta \in \mathbb{R}^p$, ($p \in \mathbb{N}$) are the free parameters.

To evaluate the distance between G_θ and F , we use a so-called loss function, that can be formally described as

$$E : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, +\infty]. \quad (2.3)$$

Specifically, the model's error may be described by the quantities $E_{F,\theta}(x) := E(F(x), G_\theta(x))$, where $(x \in \mathcal{X})$.

Henceforth we consider classification problems, where $\mathcal{X} \subset \mathbb{R}^N$ is a countable set of input vectors. The objective of SL methods is to find θ parameters, which minimize the expression

$$r(\theta) := \sum_{x \in \mathcal{X}} E_{F,\theta}(x). \quad (2.4)$$

During the training procedure, not all elements of $\mathcal{X}$ are available. Therefore, the values of $F(x)$ are known only on a smaller $\mathcal{T} \subset \mathcal{X}$ set. For example consider the problem of recognizing road abnormalities (such as potholes) from a tire implanted force sensor [6]. In a single measurement campaign, only a finite number of tire revolutions can be recorded. Despite this, a well constructed SL method should recognize road abnormalities on any surface, (even if they were not present in the measurements used to train the model).

In order to measure the generalization properties of the trained model, the set of available measurements $\mathcal{T}$ is usually split into two disjoint subsets:

$$\mathcal{T} = \mathcal{T}_{tr} \cup \mathcal{T}_{te} \quad \mathcal{T}_{tr} \cap \mathcal{T}_{te} = \emptyset. \quad (2.5)$$

Model parameters are tuned using data from the so-called training set $\mathcal{T}_{tr}$. The trained model is evaluated on $\mathcal{T}_{te}$ called the test set. Even though the training set usually contains most of the available data samples, the ratio between the two sets is heavily application and data availability dependent [4]. If data is abundant, $\mathcal{T}_{tr}$ may contain a higher ratio of the available samples [4]. In a data scarce environment, $\mathcal{T}_{tr}$ will usually contain a lower ratio of the total number of input signals. In some cases, special techniques such as one-shot-learning (see e.g. [9]) may be required to ensure high quality model generalization.

It is crucial that the elements in the test set are not included in the set used for training, because we want to test how the model behaves on unseen data. The elements of the test and training sets are usually chosen randomly. Although for some specific problems other strategies such as time-based splitting also exist [10]. It is possible that the elements of a test set give too favourable or poor results, which can deceive us about the true performance of the model. Therefore, for a more robust evaluation, we divide the input samples into disjoint test sets. We get K test and training sets. We perform the optimization process on each of these test and training sets and take the average value to get a good estimate of the expressive power of

the model. This methodology is referred to as K-fold cross-validation, where $K \in \mathbb{N}$ [4]. If the dataset is unbalanced (biased towards certain classes), we can also make sure that for each fold the rate of classes is equal to the rate in the whole data set. (stratified K-fold cross-validation).

Overfitting occurs, when a model performs well on the training set but achieves high loss scores on the test set. To avoid this, we can choose between several strategies. One common cause of overfitting is that our model has been learning for too long. It has learned the specifics of the training set, which do not characterize the mapping F in a general sense. In order to avoid overfitting it is recommended, that training is terminated in a reasonable time [4]. Another reason could be that we are working with too many parameters, in which case we should reduce model complexity [4].

If our trained model cannot achieve sufficient accuracy, another solution is to penalize the learned parameters, using regularization. In this case, we usually increase the cost function by the norm of the parameters, because they typically increase when overfit. Then, the modified loss is expressed by

$$r(\theta) := \sum_{x \in \mathcal{T}_{te}} E_{F,\theta}(x) + R(G_\theta(x)). \quad (2.6)$$

It has been shown that data representation plays a key role in the performance of SL methods [6]. Namely, transforming the input data to emphasize certain properties can greatly enhance model performance. These so-called feature transformations can reduce the dimension of the input and increase the interpretability of ML algorithms [5]. Popular static feature extraction techniques include PCA, DFT, and the calculation of the standard deviation of the input vectors [2]. In contrast, so-called automatic feature extraction schemes (such as the first layers of CNNs) have learnable parameters and thus provide adaptive data representations. Unfortunately, as mentioned in the introduction, traditional automatic feature extraction methods have the disadvantage over model-based approaches that the features found are not explainable. Formally, feature extraction methods transform the set of inputs

$$\Psi := \mathcal{X} \rightarrow \mathcal{X}' \quad (2.7)$$

and the SL problem is solved over

$$\mathcal{X}' := \{\Psi(x) : x \in \mathcal{X}\}. \quad (2.8)$$

In the next section, we will discuss in more detail the advantages of automatic feature extraction and model-based machine learning algorithms.

2.2 Model based machine learning

Applying the appropriate mathematical transformation to solve signal processing problems is difficult task due to a variety of reasons presented in the previous section (i.e. approximation of nonlinear operators). Data-driven methods have provided a solution to this. One important disadvantage of these is that the learned parameters are difficult for humans to interpret. Neural network models for example often contain millions of parameters [4]. This makes the task of explaining model behavior very difficult. Many application areas, such as automotive or medicine, require models to be interpretable, which motivates research into explainable artificial intelligence (XAI) methods [3].

Although several approaches, such as post-hoc methods [11] have been considered in recent years, one promising research direction of XAI is model-driven learning [5, 12]. Model-based machine learning algorithms aim to combine the advantages of both classical signal processing and data driven methods. The advantage of ML algorithms is that the parameters of feature extraction transformations are learned together with parameters of the underlying ML model. Thus, during training, the search for the appropriate parameters for the $\Psi(x)$ transformation is performed automatically.

The real-time application of signal processing algorithms on low-cost devices such as microcontrollers or field programmable gate arrays (FPGAs) is often expected. Because they work with a small number of parameters, model-based machine learning algorithms meet these requirements. An example could be the detection of road abnormalities while driving using GPS coordinates [6]. Another example could be the real time detection of hardware malfunction of various sensors [3]. In previous works [3], [7] we implemented SL methods based on the model driven paradigm on microcontrollers to solve the above mentioned signal processing problems.

In the next section we review so-called variable projection operators, which form the basis of several model-driven machine learning algorithms [5, 3]. In one of our proposed methods, these operators are also used to approximate continuous wavelet coefficients. The first example of a model driven algorithm relying on variable projection operators was the VP-NET model introduced by Kovács et al. in [5].

2.2.1 Variable projection operators

Signal processing methods are usually based on analytical models. The function f describing the input is assumed to have appropriate properties for practical applications. This may be, for example the existence of its integral or derivative. For example, in machine learning methods, most optimization techniques assume (sub) differentiability of the error functional (2.3). In this thesis, the input signals will be assumed to belong to the Hilbert space $L_2(\mathbb{R})$ [1]. In particular a signal f belongs to $L_2(\mathbb{R})$ if it satisfies

$$\left(\int_{\mathbb{R}} |f(x)|^2 d\lambda \right)^{1/2} < \infty, \quad (2.9)$$

where λ denotes the Lebesgue-measure [1]. It is important to note, that in $L_2(\mathbb{R})$ two functions are considered identical if they only differ over a zero measure set. As mentioned above, this class of so-called square integrable [1] functions forms a Hilbert space with the inner product

$$\langle f, g \rangle := \int_{\mathbb{R}} f(t) \bar{g}(t) d\lambda, \quad (f, g \in L_2(\mathbb{R})), \quad (2.10)$$

where $\bar{g}$ denotes the complex conjugate of g . Elements in $L_2(\mathbb{R})$ are also referred to as signals of finite energy [1].

Let $G \subset X$ be a closed subspace and assume that $(X, \langle, \rangle)$ is a complete inner product space (Hilbert space). The theorem on best approximation in Hilbert spaces states that there exists a best approximation $\tilde{f} \in G$ to f and that the vector $f - \tilde{f}$ is orthogonal to the elements of G [13].

It follows from this theorem that f can be written as $f = f_1 + f_2$, where $f_1 \in G$ and f_2 is orthogonal to G . We can now introduce the concept of orthogonal projection:

Definition 1 (Orthogonal projection). *Let G be a closed subspace of the Hilbert space $(X, \langle \cdot, \cdot \rangle)$ and $f = f_1 + f_2$, $f_1 \in G$, $f_2 \perp G$. The operation $P_G : X \rightarrow G$ defined by $P_G f := f_1$ is called an orthogonal projection.*

If the dimension of G is finite, then from its basis vectors we can write every $\tilde{f} \in G$ as a linear combination. That is the best approximation of f can be written as:

$$\tilde{f} = \sum_{k=0}^m c_k g_k, \quad (2.11)$$

where $\tilde{f}$ is the best approximation of f from G .

Henceforth, let us consider the Hilbert space $L_2(\mathbb{R})$ as defined in Eq. (2.9). Furthermore, consider an $n \in \mathbb{N}_+$ dimensional subspace G spanned by the linearly independent system $\phi_0, \dots, \phi_{n-1}$. Suppose furthermore that each $\phi_k := \phi_k^\eta$ depends on the real parameter vector $\eta \in \mathbb{R}^m$ ($m \in \mathbb{N}$). Consequently, the subspace G also depends on this parameter and shall be denoted by $G(\eta)$. Then, obtaining an (L_2 optimal) representation of f is equivalent to solving the so-called separable nonlinear least squares problem:

$$r_2(\eta) := \|f - P_{G(\eta)} f\|_2^2 \quad (f \in L_2(\mathbb{R})), \quad (2.12)$$

where $P_{G(\eta)}$ denotes the orthogonal projection of f onto $G(\eta)$. Since this operator also depends on η it is referred to as a variable projection operator.

We can only measure phenomena in the world with finite precision. It is the succession of these measurements that produces our input signal. We process these signals on computers, so our input can only be discrete and finite. For this reason, the continuous signals used in mathematical analysis are approximated numerically.

Formally, let $\mathbf{f} \in \mathbb{R}^N$, the N point sampling of a signal $f \in L_2(\mathbb{R})$. That is, $\mathbf{f}_k = f(t_k)$ ($t_k \in \mathbb{R}$, $k = 1, \dots, N$). In feature extraction procedures, we want to find a representation of $\mathbf{f}$ using $n \ll N$ numbers. Such a signal representation may be achieved by projecting $\mathbf{f}$ onto an n dimensional subspace in $\mathbb{R}^N$.

If we take a discrete sampling of the basis functions ϕ_k^η equal to the sampling of f , we obtain a transformation matrix $\Phi(\eta) \in \mathbb{R}^{N \times n}$, where N denotes the number of sampling points and n is the dimension of the subspace $G(\eta)$. In this case, the orthogonal projection can be expressed by the Moore-Penrose pseudo inverse of Φ ,

i.e.

$$P_{\Phi(\boldsymbol{\eta})}f := \Phi(\boldsymbol{\eta})\Phi(\boldsymbol{\eta})^+f \quad (2.13)$$

Thus, in the discrete case the best approximation to $\boldsymbol{f}$ (in the least squares sense) may be obtained by minimizing

$$r_2(\boldsymbol{\eta}) := \|f - P_{\Phi(\boldsymbol{\eta})}f\|_2^2 = \|f - \Phi(\boldsymbol{\eta})\Phi(\boldsymbol{\eta})^+f\|_2^2 \quad (2.14)$$

This nonlinear optimization task may be solved by gradient method due to the result of Golub and Preyera [14], who gave an analytic formulation of the derivative of $\Phi(\boldsymbol{\eta})^+$ with respect to $\boldsymbol{\eta}$:

$$\partial\Phi^+ = -\Phi^+\partial\Phi\Phi^+ + \Phi^+[\Phi^+]^T\partial\Phi^T(I - \Phi\Phi^+) + (I - \Phi^+\Phi)\partial\Phi^+[\Phi^+]^T\Phi^+ \quad (2.15)$$

Thus we obtain a representation of f whose coefficients can be used as lower dimensional inputs that have physical meaning.

2.2.2 VP-Net – variable projection networks

Based on the above idea, we can construct a neural network with a custom variable projection layer (VP-layer) as the first layer [5]. The VP-layer will perform the automatic feature extraction and pass the transformed input - the computed coefficients, the signal approximation or the residual vector - to the classification algorithm. Formally

$$c = \Phi^+(\boldsymbol{\eta})x, \quad \tilde{x} = \Phi(\boldsymbol{\eta})\Phi^+(\boldsymbol{\eta})x, \quad r = x - \tilde{x}. \quad (2.16)$$

It is important to note once again, that in VP-NET, the parameters $\boldsymbol{\eta}$ which characterize the data representation are optimized together with the weights of the underlying neural network to achieve automatic feature extraction. On the other hand, the basis functions determining the columns of $\Phi(\boldsymbol{\eta})$ are hand-picked for the problem at hand to achieve interpretability. In many applications [6] [5] VP-NET architectures have been shown to outperform CNN models while significantly reducing model complexity.

An important consideration is the choice of the function system that is used to approximate the input signals. In several previous applications of VP-NET, so-called

adaptive Hermite function based VP-layers were used to approximate biomedical signals [5]. The VP-Net has also been shown to be suitable for industrial applications, such as road abnormality detection [6]. The quasi-compact, quasi-periodic signals produced by a sensor mounted in a tire of a car can also be well approximated by Hermite functions because they are similar to ECG signals. Figure 2.1 illustrates the basic VP-NET architecture.

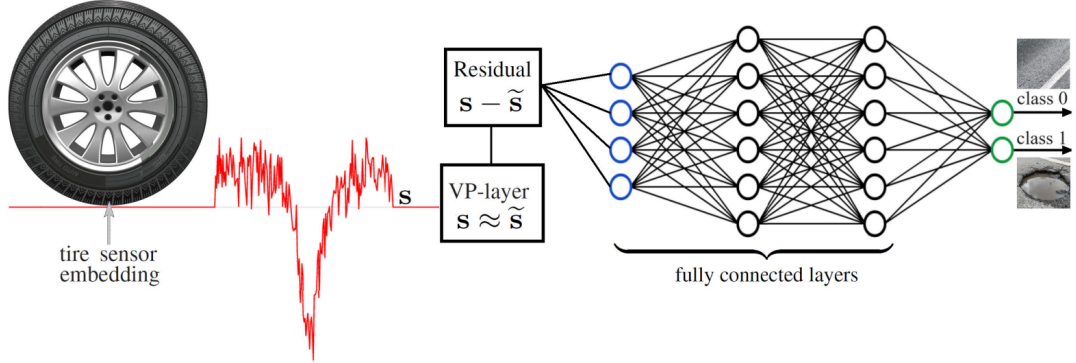


Figure 2.1: VP-Net [5]

2.3 Continuous wavelet transform

Suppose that $f, \psi \in L_2(\mathbb{R})$. The continuous wavelet transform (CWT) with respect to ψ is defined by

$$W_\psi f(x, s) := \langle f, \psi_{x,s} \rangle = |s|^{-1/2} \int_{\mathbb{R}} f(t) \overline{\psi(s^{-1} \cdot (t - x))} dt, \quad (2.17)$$

where $x, s \in \mathbb{R}, s \neq 0$ and $\langle \cdot, \cdot \rangle$ denotes the $L_2(\mathbb{R})$ inner product given Eq. (2.10). The s dilated and x translated ψ functions ($\psi_{x,s}(t)$) called wavelets. The prototype function $\psi(t)$ is called the mother wavelet or analyzing wavelet. For fixed s and x $W_\psi(x, s)$ is referred to as a "wavelet coefficient". Note that wavelet coefficients may be obtained by a convolution operation

$$W_\psi f(x, s) = (f * \psi_{\cdot, s})(x). \quad (2.18)$$

A wavelet with a small scale s exhibits fine time resolution but coarse frequency resolution, whereas a wavelet with a large scale s demonstrates fine frequency resolution but coarse time resolution. Therefore the magnitude of the $W_\psi(x, s)$ continuous

wavelet transform coefficients for large s tells us the extent to which low frequency components were present in the signal at a given time t , while for small s dilatation we can examine the presence of high frequencies.

In applications, an N sampling of f is known. Thus, we can compute the wavelet transform only for discrete parameters s_j and x_i . If for a given s_j and x_i parameters ψ_{s_j, x_i} is an orthogonal function system in $L_2(\mathbb{R})$, then ψ is called an orthogonal wavelet. Such orthogonal wavelets can be constructed using multi-resolution analysis (MRA), such as r -times differentiable Daubechies wavelets. The discrete wavelet transform (DWT) uses orthogonal wavelets. Fig.2.2 illustrates the iterative filtering based technique used to construct DWT coefficients.

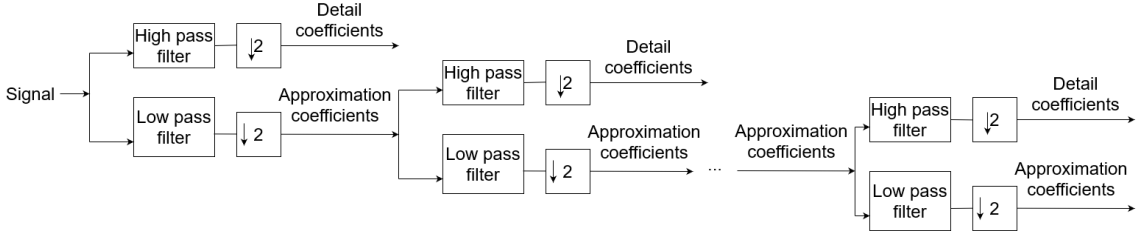


Figure 2.2: Figure of the DWT method

The DWT method is not time invariant, which causes problems when analyzing signals. Furthermore, the mother wavelets used for the transform are usually asymmetric [15]. Constructing orthogonal wavelets is challenging, particularly when the resulting mother wavelet needs to be dependent on learnable parameters.

In contrast, analytic wavelets are symmetric and have better smoothness properties than orthogonal wavelets. Furthermore, the continuous wavelet coefficients are time invariant. In practical applications, (since only a discrete sampling of f is available) we can only approximate the continuous wavelet coefficients W_ψ using numerical quadratures. Approximate coefficients obtained in this way will be denoted by CWT coefficients henceforth.

For discrete s_j and x_i , we can obtain the coefficient matrix of the continuous wavelet transform, whose elements are (the approximations of) $W_\psi(x_i, s_j)$. A common representation of $W_\psi(x_i, s_j)$ is the so-called scalogram. In this case, the absolute value of the coefficients is plotted as a heatmap.

Commonly used analytic wavelets include the Morlet and Ricker wavelets [8]. These are illustrated in Figure 2.3.

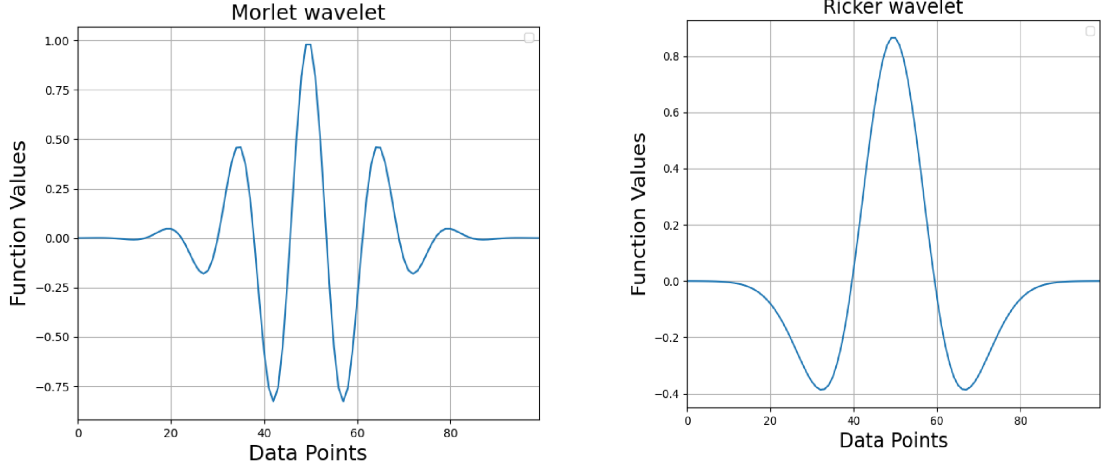


Figure 2.3: The Morlet and Ricker mother wavelets.

The Morlet and Ricker wavelets are defined by

$$\psi_{x,s}(t) := \frac{1}{\sqrt{s}} \cdot e^{-\left(\frac{t-x}{s}\right)^2/2} \cdot \cos\left(\omega_0 \cdot \frac{t-x}{s}\right), \quad (2.19)$$

where $s > 0$ and $x, \omega_0 \in \mathbb{R}$ and

$$\psi_{x,s}(t) := \frac{1}{\sqrt{s}} \cdot c \cdot e^{-\frac{\left(\frac{t-x}{s}\right)^2}{2}} \left(1 - \frac{t-x^2}{s}\right), \quad (2.20)$$

where $c \in \mathbb{R}$, respectively. The objective of the methods proposed in study is to identify appropriate s_i scale and x_j translation parameters which can be used to solve signal classification problems. To this end, we will rely on the partial derivatives of the mother wavelets with respect to these parameters:

Morlet partial derivatives:

$$\frac{\partial \psi_{x,s}(t)}{\partial s} = -\left(\frac{1}{2s} \psi_{x,s}(t) + a(t) \frac{\partial a}{\partial s}(t)\right) \psi_{x,s}(t) - \omega s^{-1/2} \frac{\partial a}{\partial s}(t) e^{-a^2(t)/2} \sin(\omega a(t)), \quad (2.21)$$

$$\frac{\partial \psi_{x,s}(t)}{\partial x} = -a(t) \frac{\partial a}{\partial x}(t) \psi_{x,s}(t) - s^{-1/2} \omega \frac{\partial a}{\partial x}(t) e^{-a^2(t)/2} \sin(\omega a(t)), \quad (2.22)$$

Ricker partial derivatives:

$$\frac{\partial \psi_{x,s}(t)}{\partial s} = -\left(\frac{1}{2s} \psi_{x,s}(t) + a(t) \frac{\partial a}{\partial s}(t)\right) \psi_{x,s}(t) - c \cdot s^{-1/2} \frac{\partial a}{\partial s}(t) e^{-a^2(t)/2} \cdot 2a(t), \quad (2.23)$$

$$\frac{\partial \psi_{x,s}(t)}{\partial x} = -a(t) \frac{\partial a}{\partial x}(t) \psi_{x,s}(t) - s^{-1/2} c \frac{\partial a}{\partial x}(t) e^{-a^2(t)/2} \cdot 2a(t), \quad (2.24)$$

where

$$a(t) = a_{x,s}(t) := \frac{t-x}{s}, \quad \frac{\partial a}{\partial s}(t) = -\frac{t-x}{s^2}, \quad \frac{\partial a}{\partial x}(t) = -\frac{1}{s}.$$

The result of the wavelet transform depends largely on the shape of the chosen mother wavelet. The Ricker wavelet for instance fits well to ECG signals. The following figure 2.4 clearly shows the difference between the scalogram of a continuous wavelet transform performed with Morlet and Ricker wavelet. With Morlet wavelet, the bands in the matrix containing important frequencies are less prominent in the image, making it more difficult for a classification algorithm to distinguish between an ill and a healthy signal.

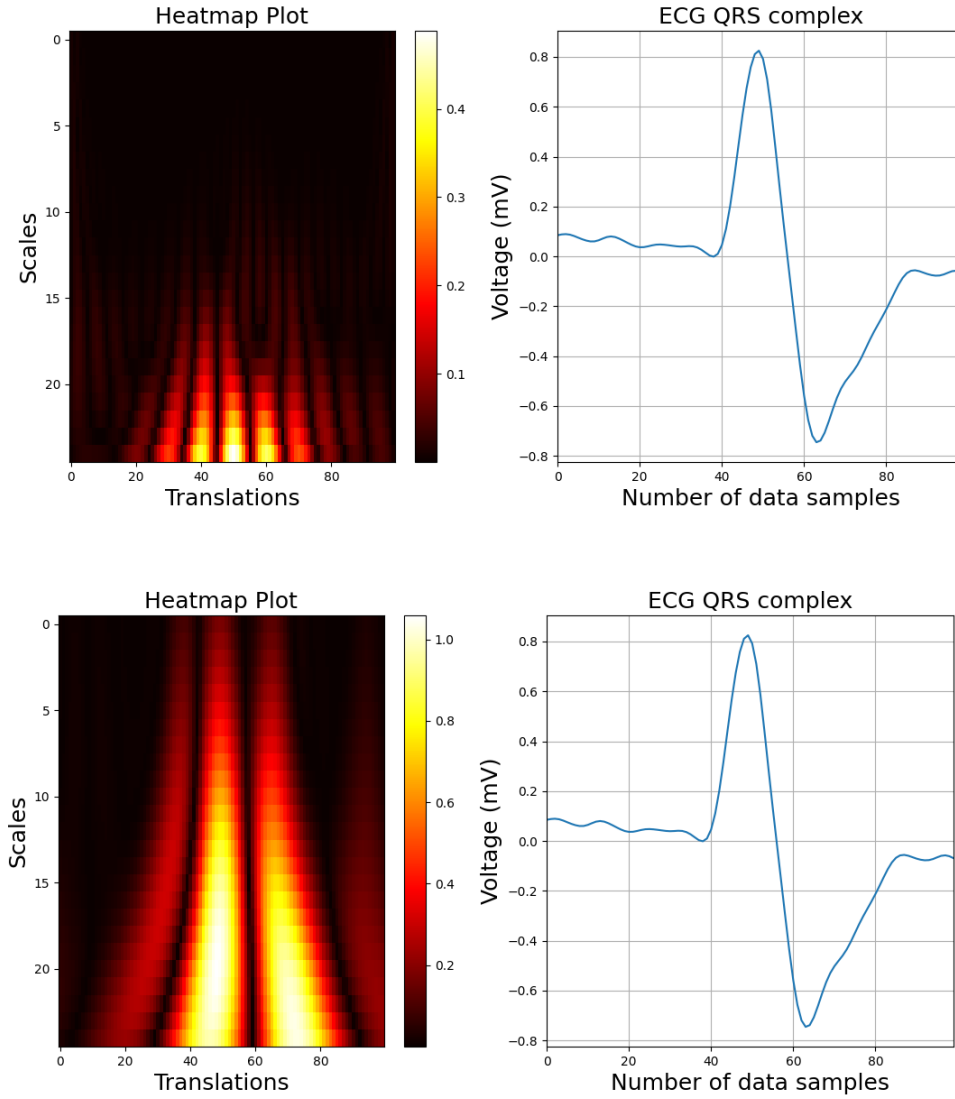


Figure 2.4: Scalogram of an ECG signal with Morlet and Ricker wavelets, respectively.

The Ricker wavelet is the second derivative of the Gaussian function. In chapter 4 we will show, that the proposed wavelet coefficient based neural networks perform particularly well for ECG processing tasks, when CWT is realized using the Ricker wavelet.

Two formulas will be used to approximate the CWT coefficients. Since the wavelet transform can be written as the inner product of the function f and $\psi_{x,s}(t)$, wavelet coefficients may be approximated using a variable projection based approach. In particular, consider a finite set of $\boldsymbol{\eta} := \{(x_i, s_i) : x_i \in \mathbb{R}, s_i > 0\}$ ($i = 0, \dots, m-1$) scale-translation parameter pairs and the corresponding ψ_{x_i, s_i} system of wavelets. These are clearly linearly independent and span an m dimensional subspace denoted by $G(\boldsymbol{\eta})$. Then, the wavelet coefficients $W_\psi f(x_i, s_i)$ can be approximated by

$$W_\psi f(x_i, s_i) \approx \Psi(\boldsymbol{\eta})^+ \mathbf{f} =: \mathbf{w} \in \mathbb{R}^m.$$

Although the CWT approximation obtained in this way does not precisely match wavelet coefficients (by definition), the components of $\mathbf{w}$ still capture the similarity between f and ψ_{x_i, s_i} resulting in an appropriate time-scale representation for classification problems. This hypothesis is verified by the experimental results detailed in chapter 4.

In particular, the variable projection based ML algorithms work well when the condition of the transformation matrix is low, so that the samples of orthogonal basis functions are almost discrete orthogonal to each other and linearly independent.

The other proposed method for implementing CWT coefficients is the approximation of the integral (2.17). In this thesis, wavelet coefficients were approximated by interpolatory quadrature rules of the form:

$$Q(f) := \sum_{i=0}^{N-1} w_i f(x_i) \approx \int_a^b f(x) dx \quad (2.25)$$

where the weights w_i correspond to the integrals of Lagrange basis polynomials defined by the sampling points x_i . In particular, in our implementation the composite midpoint rule was utilized [16].

Chapter 3

CWT-VP and CWFCNN

In this chapter, we discuss the proposed wavelet transformation-driven ML methods. First, we describe the architecture of the CWT coefficient based methods, and then we discuss the differences between the proposed variations. Finally, a qualitative comparison against a state-of-the-art wavelet convolution network proposed in [8] is provided.

3.1 Architecture of the neural networks

The core idea behind the proposed ML methods is that the first layer of a neural network is replaced by a custom layer that using the continuous wavelet transform. Using the usual scalograms as input features to our ML models would result in a redundant signal representation which is contrary to our objectives. Instead, we only want to compute n coefficients needed to perform the classification. In the proposed methods, the ML models learn which n wavelet coefficients can best represent the input signals to achieve high classification accuracy. Since continuous wavelet transform coefficients depend on the s dilatation and x translation parameters, we randomly initialize n pair of (s, x) parameters. Thus, we compute the $\psi_{x_i, s_i}(i = 0, \dots, n - 1)$ dilated and translated wavelets sampled at N points. These are used to compute the coefficients. Thus $\boldsymbol{\eta} := (s_i, x_i) \quad (i = 0, \dots, n - 1)$ denote the learnable parameters of the CWT layer, which are varied during training. At the end of the training, the learned η parameters can be interpreted by humans and contain important time-frequency information of the signal.

The linear layers of a fully connected neural network can be written as follows

$$g_k(x) = W_k x + b_k, \quad (3.1)$$

where $W \in \mathbb{R}^{L_k \times L_{k-1}}$ is the weight matrix, $b_k \in \mathbb{R}^{L_k}$ is the bias vector, L_k is the number of neurons in the k th layer, $L_0 = n$ is the input vector, i.e., the output of the CWT coefficient layer. Thanks to the automatic feature extraction, we can greatly reduce the dimension of the input, so that the dimension of the first linear layer is $\mathbb{R}^{n \times L}$ instead of $\mathbb{R}^{N \times L}$, resulting in simpler model architecture.

In order to model nonlinearities, neural networks often employ so-called activation functions on the output of linear layers:

$$g_k(x) = h(W_k x + b_k). \quad (3.2)$$

A neural network can be described as a nested composition of 3.2 functions, where the argument of the innermost mapping is the input data.

$$G_{\theta} = (g_K \circ \cdots \circ g_k(x) \circ \cdots \circ g_0)(x) \quad (3.3)$$

The activation function of the last layer is the Sigmoid function, which maps the output of the neurons between 0 and 1. If we have to classify the elements of $\mathcal{X}$ into two classes (binary classification), the last layer contains a single neuron. The neural networks considered in this study used the ReLU (rectified linear unit) [4] activation for intermediate layers. Together with the CWT layer, the parameters of the network are $\Theta = \{\eta, W_1, b_1, \cdots, W_k, b_k, \cdots, W_K, b_K\}$, where K is the number of layers.

We chose Binary Cross Entropy (BCE) [4] as the cost function for binary classification:

$$r(\theta)_{BCE} := -\frac{1}{q} \sum_{i=1}^q F(x_i) \log G_{\theta}(\mathbf{x}_i) + (1 - F(x_i)) \cdot \log(1 - G_{\theta}(\mathbf{x}_i)), \quad (3.4)$$

where q denotes the number of available input signals in the training set. We apply the chain rule when calculating the derivative of G_{θ} . We will also be able to give an analytical formula for the derivatives of CWT layers. In this study, the proposed neural network models were trained using the so-called backpropagation algorithm [4], which computes the required derivatives and then optimizes the network parameters based on a gradient method.

A commonly used stochastic gradient method is ADAM [4], which we also utilized in the experiments detailed in this thesis. In ML, due to high number of signals in the training set, we usually optimize the model parameters using an average of the derivative of a set of inputs. The hyperparameter associated this is the batch size, denoted by q in Eq. (3.4)

The networks were implemented with the python Pytorch package [17], using its built-in functions. For the custom layers, we created a separate layer class, where we also specified the unique backward pass, i.e. the partial derivatives.

The matrix multiplications of the linear layers of a trained neural network can be easily implemented on a microcontroller [7]. The CWT layers also work with few parameters so the network may be suitable for real-time use on low-power devices [3].

3.2 CWT-VP layer

For the CWT-VP model, the coefficients for the first CWT layer are computed with variable projection based on Eq. (2.16). Thus, the first layer is formally gives as

$$g^\eta : \mathbb{R}^N \rightarrow \mathbb{R}^n, \quad g^\eta(\mathbf{f}) := \Psi(\eta)^+ \mathbf{f}. \quad (3.5)$$

operator. A schematic representation of the CWT-VP network is provided in figure 3.1.

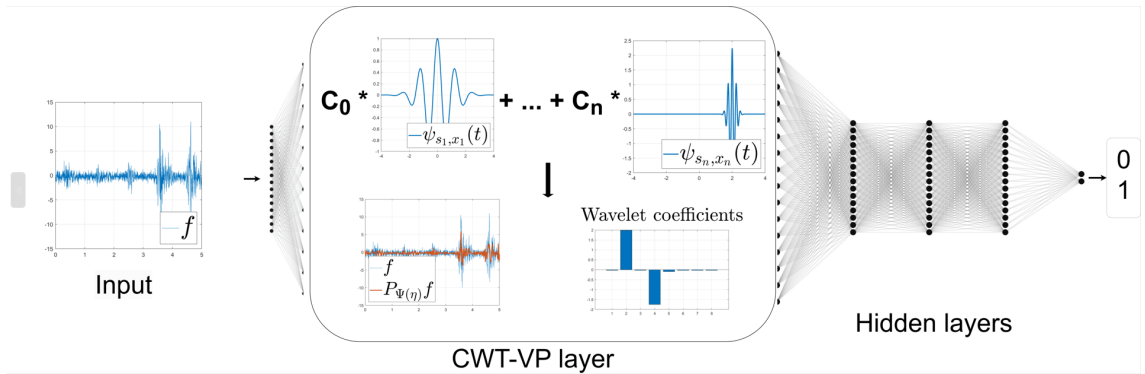


Figure 3.1: CWT-VP networks.

For backpropagation, we can compute the partial derivative of the pseudo inverse matrix using the Eq. 2.15. The partial derivatives of the Ricker and Morlet

wavelets can also be calculated analytically according to 2.21, 2.22 and 2.23, 2.24. In order to avoid overfitting, we apply the percent root-mean-square difference (PRD) regularization for the layer parameters as proposed in Kovács et al [5].

$$r_{VP}(\boldsymbol{\eta}) := \frac{\alpha}{q} \sum_{i=1}^q \frac{\|\mathbf{f}_i - g^{\boldsymbol{\eta}}(\mathbf{f}_i)\|_2^2}{\|\mathbf{f}_i\|_2^2},$$

where α is the VP penalty term hyperparameter and q denotes the batch size.

3.3 CWFCNN layer

In this case, the CWT coefficients will be given by a quadrature formula, so the formula for the layer forward pass is

$$g^{\boldsymbol{\eta}} : \mathbb{R}^N \rightarrow \mathbb{R}^n, \quad g^{\boldsymbol{\eta}}(\mathbf{f}) := \Psi(\boldsymbol{\eta})^T \mathbf{f}. \quad (3.6)$$

In this case, the Jacobian of $g^{\boldsymbol{\eta}}$ only depends on the derivative of the mother wavelets. Note that other wavelets than those mentioned above can be used if we can provide their partial derivatives with respect to the scale and translation parameters. CWFCNN is illustrated in the figure 3.2 below.

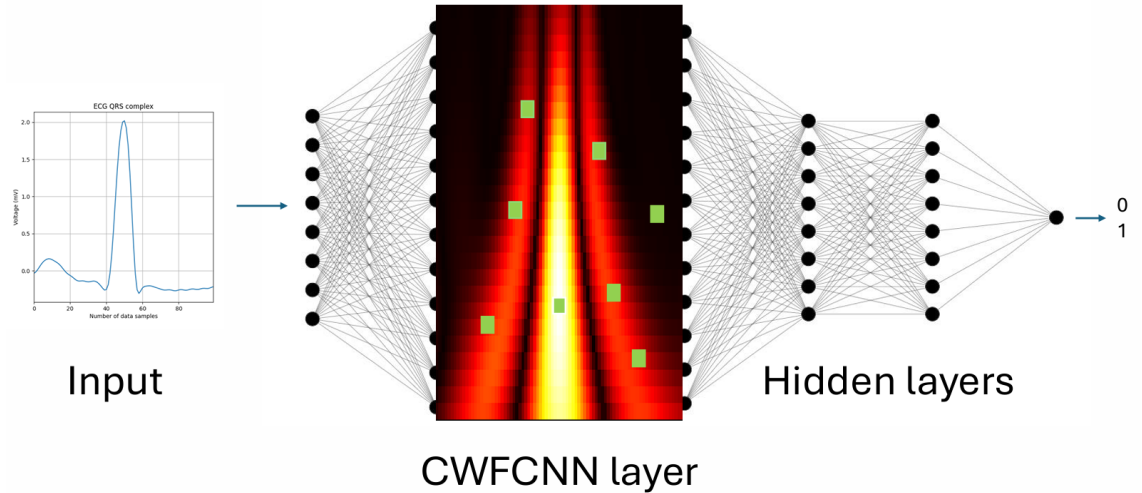


Figure 3.2: CWFCNN architecture.

The learning of $\boldsymbol{\eta}$ parameters is controlled by Tikhonov regularization. We calculate the $\|\cdot\|_2$ norm of the parameters of our model:

$$r(\theta) := r(\theta)_{BCE} + \lambda \sum_{i=1}^p \eta_i^2 \quad (3.7)$$

where λ is the penalty parameter. If $\lambda = 0$, we get the original expression and if it is too high, the model is learning almost 0 parameters.

3.4 WKN - comparison to the state of the art

We want to compare our methods to a state-of-the-art wavelet-based method. The WaveletKernelNet (WKN) exploits the convolution form of the continuous wavelet transform 2.18, and uses analytic wavelets as the kernel function of the discrete convolution in its first layer:

$$W_\psi f(\cdot, s_i) \approx \boldsymbol{w} := \boldsymbol{f} * \psi_{s_i, x_i}. \quad (3.8)$$

Thus, for each convolution channel, the network learns the parameters (x_i, s_i) $i = 1, \dots, n$. For each ψ_{x_i, s_i} , it produces the wavelet coefficients for all translations, i.e., a particular row of the CWT matrix for s_i dilation. The number of CWT coefficients produced by the layer is $n \cdot M$, where M is the number of coefficients produced by convolution of the input signal and ψ_{x_i, s_i} . The operation of the layer is illustrated in the figure 3.3.

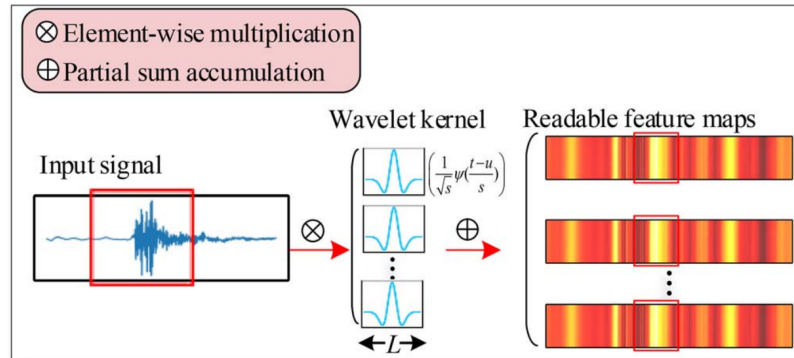


Figure 3.3: WKN layer architecture. [8]

Since $M \approx N$, a high-resolution version of the CWT coefficient matrix of the input signal is produced. The layer has been incorporated into Deep Learning convolutional networks (Alexnet, Inception, LeNet) [8] by replacing the first convolutional

layer with a wavelet-based convolutional layer. The advantage of this layer over the traditional convolutional 1D layers is that the learned kernel functions and their parameters (dilatation, translation) are interpretable by humans. The transforms of the input signal (feature maps) produced by automatic feature extraction, being a CWT coefficient matrix of n rows, are also interpretable by humans.

Since WKN is an algorithm developed for signal processing, the input is a one-dimensional signal. In these deep learning convolutional networks, 1D convolutions are performed on multiple channels in the inner layers. These techniques were originally developed for image processing, so the computational requirements and the number of parameters of the networks are large. The size of deep learning models also damages explainability of the learned parameters. The $n \cdot M$ feature map produced after feature extraction increases the dimension of the input. In contrast, the networks proposed in this thesis reduce input dimension. This sparse representation makes the learned parameters interpretable and allows for a network architecture with lower complexity. After the wavelet layer of CWT-VP or CWFCNN, there is no need to perform dropout or maxpooling operation to reduce the dimension of the features, thus increasing the interpretability of our networks.

Because of the deep learning network models and the large number of learned parameters, WKN is not suitable for implementation on low-power devices and its real-time application would require a high-performance GPU.

WKN was developed to predict the failure of machine parts. Industrial applications tested in [8] include, for example, the classification of different failure types of rolling bearings. These industrial applications may often require high-resolution feature extraction of the signal. In order to compare the expressive power of the layers, we have implemented each wavelet layer in front of the same network architecture.

Chapter 4

Experiments

Our methods have been tested on several datasets. We tried to keep the complexity of the models used as low as possible, such as the number of layers, neurons and CWT coefficients. We first compared the performance of the proposed wavelet based models with WKN on an open source bearing fault database. Then, the proposed models were used to solve a biomedical signal processing problem using a benchmark ECG database. Finally, to demonstrate the utility of the proposed method for current industrial applications, a bearing fault detection problem using real measurements obtained from our collaboration partners in Siemens Digital Industries was solved using the proposed wavelet based neural networks.

The tests were performed with Python PyTorch implementations of the methods. The hyperparameters of the networks (number of neurons, learning rate, batch size, regularization penalty, epoch, number of CWT coefficients) were fine tuned using a grid search of the parameter space. The implementation and experiments can be found at a public repository [18].

4.1 Open source bearing fault database

4.1.1 The characteristics and diagnosis of bearing fault signals

It is important in the maintenance of mechanical systems to predict failures of elements early. This protects other components and the whole system from signifi-

cant damage. A rolling bearing allows the transfer of forces between stationary and rotating components via rolling elements. The advantage of a rolling bearing is the reduction of rolling friction compared to sliding friction. Common fault types are inner and outer race fault. Figure 4.1 depicts these faults and a cross-section of a bearing.

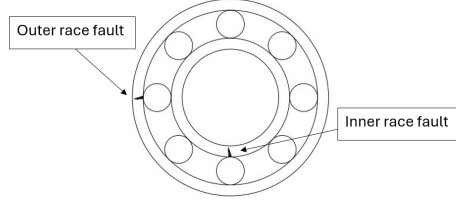


Figure 4.1: Rolling element bearing cross-section with inner and outer faults.

When a fault strikes a component in the assembly, a shock event accrues. These fault amplitudes modulate the signal and cause high frequency resonance. These frequencies can be calculated as follows:

Ballpass frequency, outer race:

$$BPFO = \frac{nf_r}{2} \left(1 - \frac{d}{D} \cos \phi \right) \quad (4.1)$$

Ballpass frequency, inner race:

$$BPFI = \frac{nf_r}{2} \left(1 + \frac{d}{D} \cos \phi \right) \quad (4.2)$$

where n is the number of rolling elements, f_r is the shaft speed, ϕ is the angle of the load from the radial plane, d is the ball bearing diameter and D is the bearing pitch diameter. However, the load changes over time and ϕ varies [19], thus the deviation of the rolling elements from their position causes random slip, which results 1-2% frequency changes. This random slip is small, but can cause big change in signal morphology. In recent years, Antoni [19] developed the mathematical formulas of bearing signals and invented tools to analyse the behavior of these signals, such as the Kurtogram [19], which uses short time Fourier-transform to detect important frequencies in the signal. By Antoni's model, the random slip in the signal can be characterised by the following equation:

$$\Delta T_i = T_{i+1} - T_i \quad (4.3)$$

This model suggests, that the position in frequencies in time is random, and the uncertainty grows over time (number of periods). Therefore, bearing fault signals must be separated from periodic and deterministic signals such as gears [19]. Cyclostationary is a special case of non-stationary random signals, which has a hidden periodic behavior. White noise modulated by a sinus wave for example falls into this category [19]. However, bearing fault signals stochastic, not periodic, and they look like a cyclostationary signal, so the mathematical model which best describe their characteristic is pseudo-cyclostationary [19].

Pseudo-cyclostationary behavior makes bearing fault classification a hard task. Antoni recommends envelope analysis of the signals, but other approaches are available as well [19]. Some article said that wavelet analysis may superior to envelope analysis at bearing fault detection [19]. In fact, envelope analysis also often relies on the wavelet transform. Envelope analysis is an amplitude demodulation technique. The envelope of a function can be obtained by the Hilbert-transform. The squared modulus of the wavelet coefficients of the signal is effectively a squared envelope signal. The above mentioned procedure can be seen in Figure 4.2.

4.1.2 Experiment setup and results on the MFPT challenge dataset

We collect acceleration signals from the MFPT challenge open source rolling element bearing fault database [20]. The dataset includes measurements gathered from bearings working under different fault conditions. From a bearing test rig, measurements include good conditions and outer race faults under constant load. Outer and inner race faults under different loads are also included. For the experiments of this thesis, data was obtained under good (baseline) conditions with a load of 270 lbs (122.47 kg). For inner and outer race conditions data was collected under a similar load of 250 lbs (113.40 kg). Figure 4.3 shows a baseline, an outer race fault and an inner race fault measurement data plot from an accelerometer, respectively.

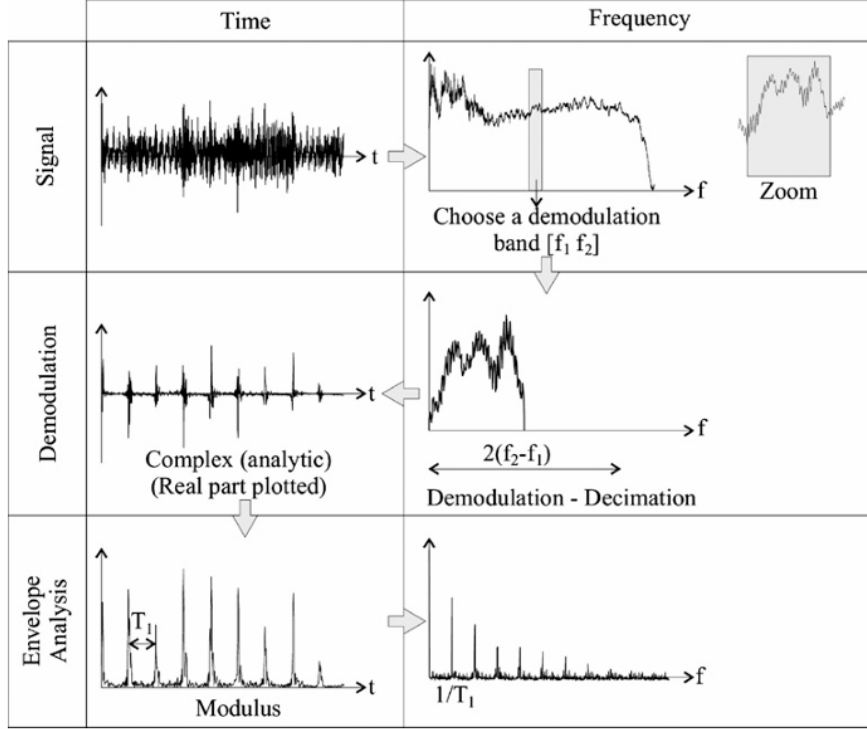


Figure 4.2: The procedure of envelope analysis [19].

We created three sets of data for training and testing. The input shaft rate was set at 25 Hz, and the sample rate was 97,656 sps (samples per second) for baseline and 48,828 sps for inner and outer race faults. We downsampled the baseline datasets two times and windowed the measurement with the number of samples per revolution per second of the inner shaft. We created the sets by slicing one measurement dataset with a window size of $48,828 / 25 = 1953$. This window size is particularly significant for inner and outer race faults, as it represents the longest modulation period in the pseudo-cyclostationary signals [19].

In the labeled sets, we included 150 normal, 75 inner fault and 75 outer fault data examples. In each set, there are two types of data: baseline (good condition) and outer race fault, baseline and inner race fault, and inner and outer race fault, respectively. We performed 5-fold cross validation on each dataset and each model. We used stratified folds to preserve the rate of each class.

Table 4.1 contains the results of the test scenarios. Each algorithm achieved high accuracy. The CWT-VP and CWFCNN consisted of a wavelet layer and three linear layer with 50 neurons. We used ReLU activation functions between each layer and

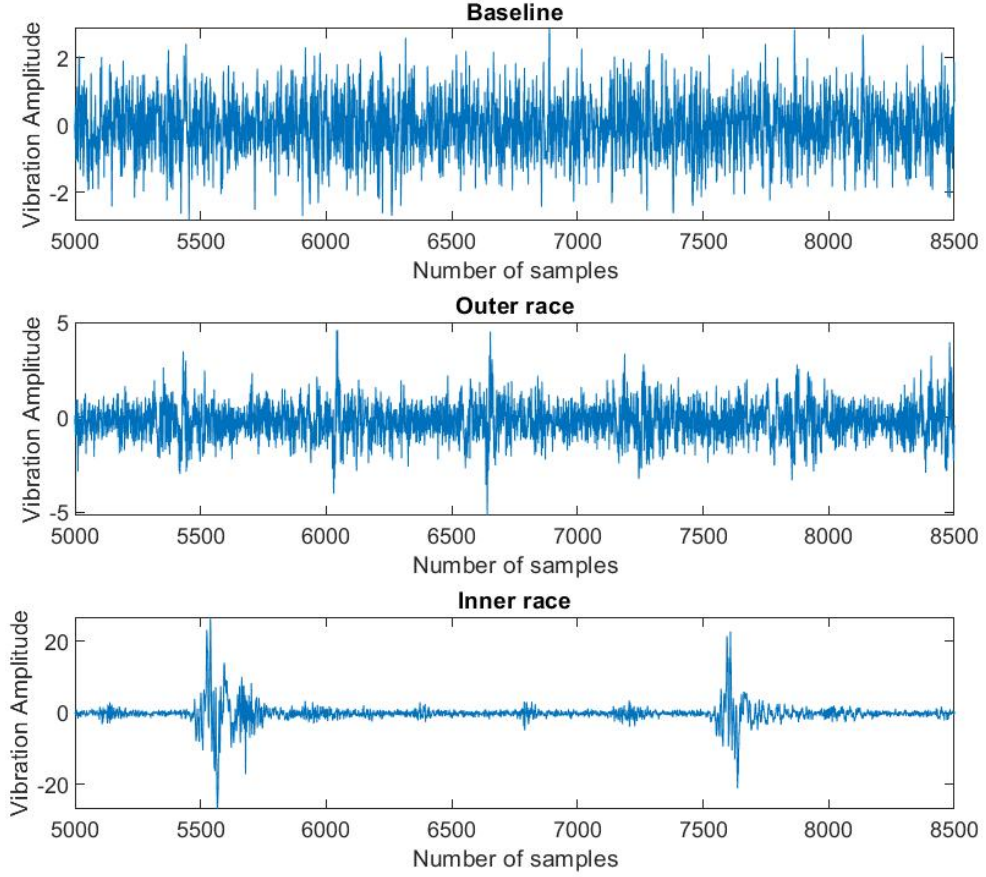


Figure 4.3: Baseline, outer race fault and inner race fault signal plot, respectively.

Sigmoid function for binary classification. We wanted to measure the difference between our layers and WKN, so the WKN model used the same network architecture, and the first wavelet layer used the [21] implementation. All wavelet layers used the Morlet analytic wavelet. WKN used 124160 wavelet coefficients and 64 dilation and translation parameter. In contrast, CWT-VP used 24-32 wavelet coefficients and 24-32 dilation and translation parameters. Despite this, CWT-VP outperformed WKN in terms of average accuracy on each dataset. This sparse representation allowed more interpretability. The CWFCNN layer also performed well, achieving the highest accuracy scores in the baseline-inner and inner-outer experiments.

Set	Layer	# of cfs.	Mean Acc.	Highest	Lowest
baseline-outer	CWT-VP	32	96%	100%	88.89%
	CWFCNN	20	94.67%	100%	82.22%
	WKN	124160	92.89%	100%	64.45%
baseline-inner	CWT-VP	24	96.89%	100%	91.11%
	CWFCNN	20	98.67%	100%	93.34%
	WKN	124160	95.11%	100%	75.56%
inner-outer	CWT-VP	32	96.67%	100%	90%
	CWFCNN	24	97.34%	100%	90%
	WKN	124160	96%	100%	80%

Table 4.1: CLASSIFICATION RESULTS OF THE BEARING DATASET

4.2 MIT-BIH ECG database

We also wanted to test our methods on a biological signal processing task. We used real ECG data from the MIT-BIH ECG database [22, 23] and compared the results to some methods reviewed in [24].

4.2.1 Description of the database and test results

MIT-BIH is an open source benchmark ECG database. It contains normal (healty) and abnormal (multiple types of arrhythmias) signals from real patients. ECG signals represent the electrical activity of the heart. Each heartbeat signal pre-processed according to [5]. The plot of an ECG heartbeat signal can be seen on 4.4. In the proposed experiments a binary classification of ECG signals was performed. In particular, the detection ventricular ectopic beats (VEBs) amongs healthy ECG signals was considered. VEBs produce abnormal QRS complexes, therefore in the experiments we used only the QRS complex of each signal [5].

We evaluated our method on two datasets created according to [5]. Each dataset contained measurements from various patients. Since automatic ECG signal classifications used for unseen patient in practice, the training and test sets were constructed from signals obtained from different patients. A balanced data set was

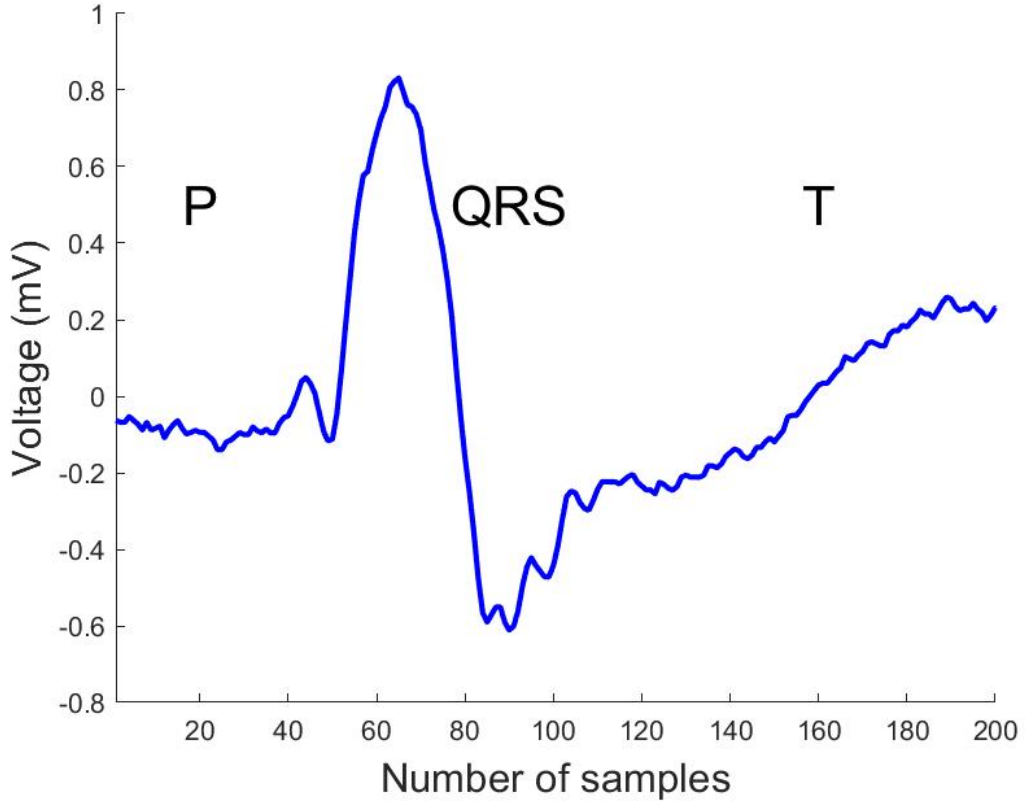


Figure 4.4: ECG heartbeat. P wave corresponds to atrial depolarization. QRS complex represents depolarization of the ventricles and T wave represents repolarization of the ventricles [3].

created with the same number of normal and VEB beats. We used 4260-4260 data samples and 3220-3220 data samples for training and for testing, respectively. This dataset provided an undistorted comparison to other methods. In real life situations, normal heartbeats appear in bigger numbers than diseased heartbeats, so an unbalanced set was created to model this phenomenon. This set contained all normal beats and VEB beats, reaching 50,000 data samples for both training and testing. The QRS complexes were represented with 100 data points.

In the previous bearing fault database experiments, CWFCNN used less wavelet coefficients and achieved better results compared to CWT-VP, so in this test scenario to achieve more interpretable result, we used only CWFCNN. We compared our results to traditional CNN and FCNN. The network architecture was created according to [5] to achieve the most interpretable result with the least number of parameters. CWFCNN used one wavelet layer with Ricker analytic wavelet and two

linear layers, with ReLU activations and a sigmoid layer applied to the output of the model. The network was tested using 5 to 20 neurons in each linear layer. We found that the model reached peak performance with 18 neurons. We note, that this classification task required less complex network architecture than bearing fault analysis. The wavelet layer only used 8 CWT coefficients. The hyperparameters of the model were found with grid search and the best result was displayed in Table 4.2.

Set	Network	Total	Normal		VEB	
		accuracy	Se	+P	Se	+P
Balanced	CWFCNN	95.92%	99.26%	93.04%	92.58%	99.20%
	FCNN	94.38%	93.79%	94.91%	94.97%	93.86%
	CNN	96.34%	97.76%	95.05%	94.91%	97.70%
Unbalanced	CWFCNN	97.54%	99.59%	97.76%	76.30%	94.68%
	FCNN	97.49%	98.50%	98.81%	83.70%	80.21%
	CNN	98.35%	99.39%	98.85%	84.07%	90.93%
Methods from Lit.		N/A	80-99%	85-99%	77-96%	63-99%

Table 4.2: CLASSIFICATION RESULTS OF THE ECG DATASET

In addition to overall accuracy, common performance metrics, sensitivity/precision (Se) and Positive predictivity/recall (+P), also calculated:

$$Se = \frac{TP}{TP + FN} \quad \text{and} \quad +P = \frac{TP}{TP + FP} \quad (4.4)$$

where TP, FN, FP are true positive, false negative and false positive, respectively.

The metrics about FCNN and CNN were obtained from the experiments detailed in [5]. CWFCNN achieved high total accuracy on both datasets. Its performance comparable to other methods from the literature. The proposed model performed relatively well even in terms of the unbalanced Normal Se and VEB +P metrics. CNN and FCNN models achieve higher accuracy in certain metrics, however this is achieved at the cost of reduced interpretability.

4.2.2 Interpretability of the learned parameters

As mentioned in the introduction, our goal was to develop an explainable AI method. CWT-VP and CWFCNN produce a sparse representation of the input signal with physical meaning (time-frequency information). The learned η parameters of CWFCNN can be mapped to coordinates of the CWT coefficient matrix. This allows us to display the learned CWT coefficients on a scalogram.

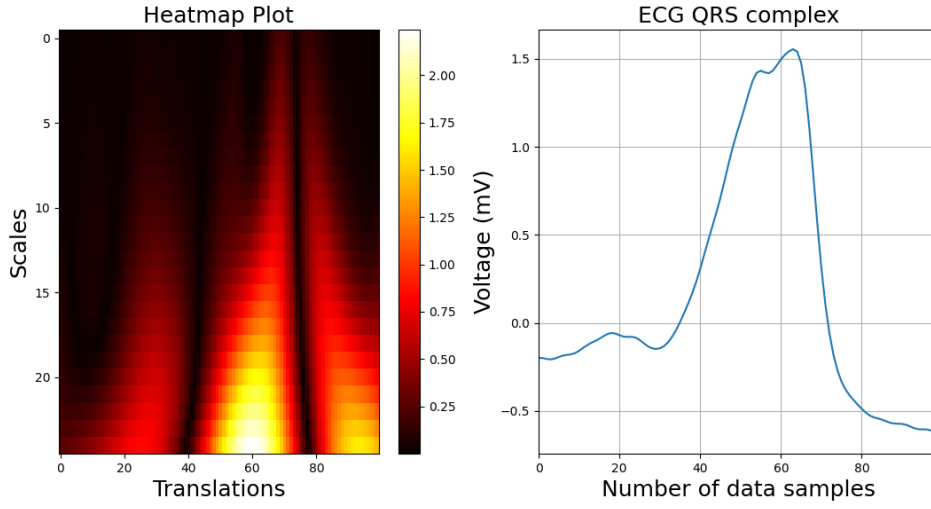


Figure 4.5: Abnormal ECG heartbeat signal.

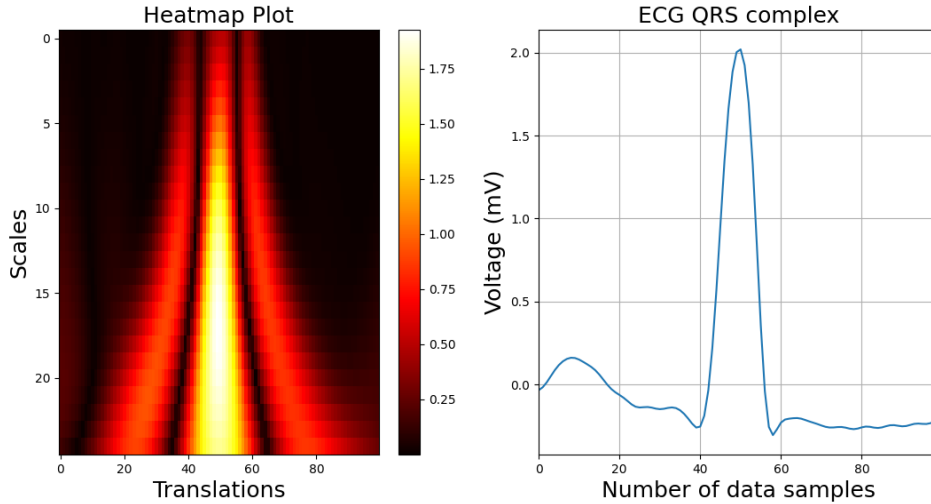


Figure 4.6: Normal ECG heartbeat signal.

In Figure 4.5 and 4.6 we can examine the differences between the scalogram of an abnormal and a healthy ECG signal. The CWT coefficients were generated by

taking their absolute values using the CWFCNN layer. As can be seen in figure 4.6, the energy of the healthy ECG signal is concentrated in the middle of the signal. The diseased signals also tend to show high energy components at locations offset from the center in time. The importance of choosing the mother wavelet has already been discussed in the 2 chapter.

The figure 4.7 illustrates the 8 wavelets initialized by random η parameters. Figure 4.8 shows the Ricker wavelets with the learned η parameters. It can be seen that the parameters have shifted from their initial values to a large extent.

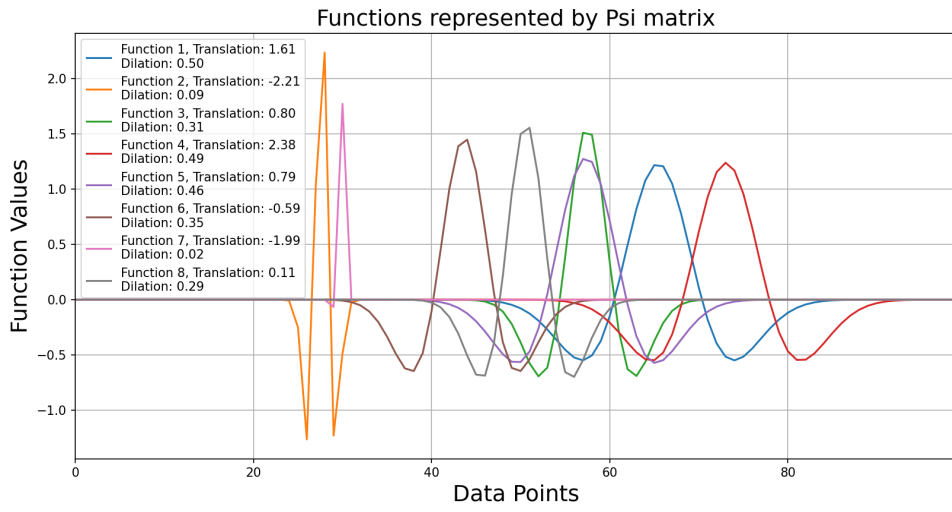


Figure 4.7: Initial Ricker wavelets of the CWFCNN ECG classification.

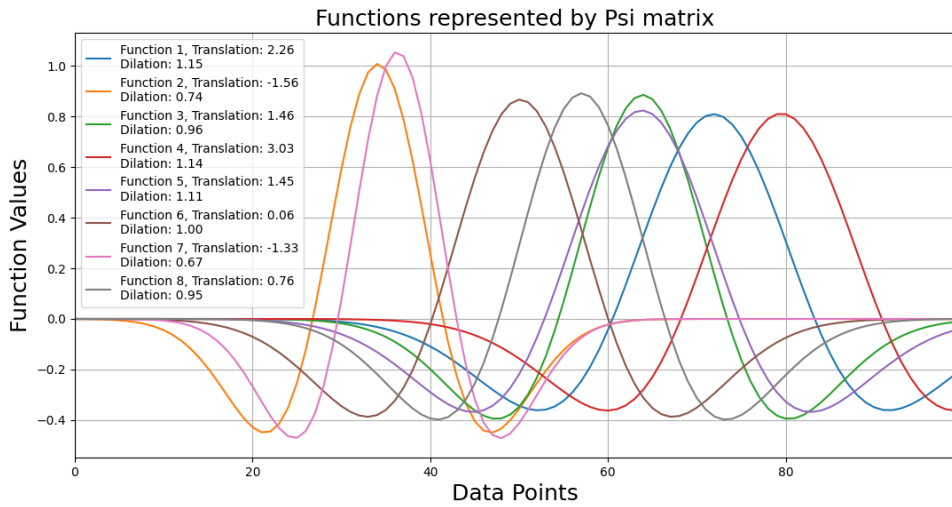


Figure 4.8: Trained Ricker wavelets of the CWFCNN ECG classification.

Figure 4.9 and figure 4.10 illustrate the parameters on scalograms of ECG signals. The initial values are shown in blue and the learned values in green. It can be seen that CWFCNN has learned the characteristic features that distinguish between a patient and a normal ECG signal. The learned parameters explain how the proposed method work and they also have a physical meaning, since they carry time-frequency information.

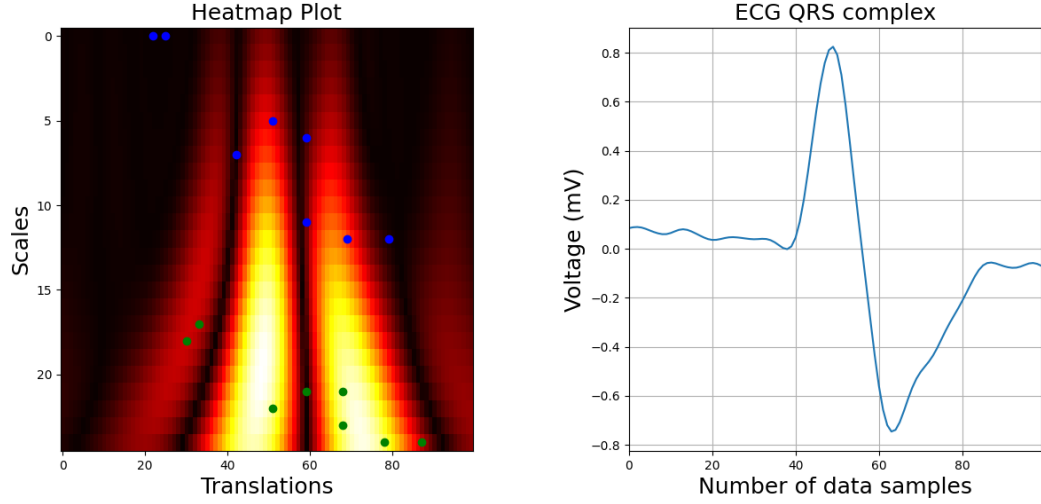


Figure 4.9: Abnormal ECG heartbeat signal with initial (blue) and trained (green) coefficients.

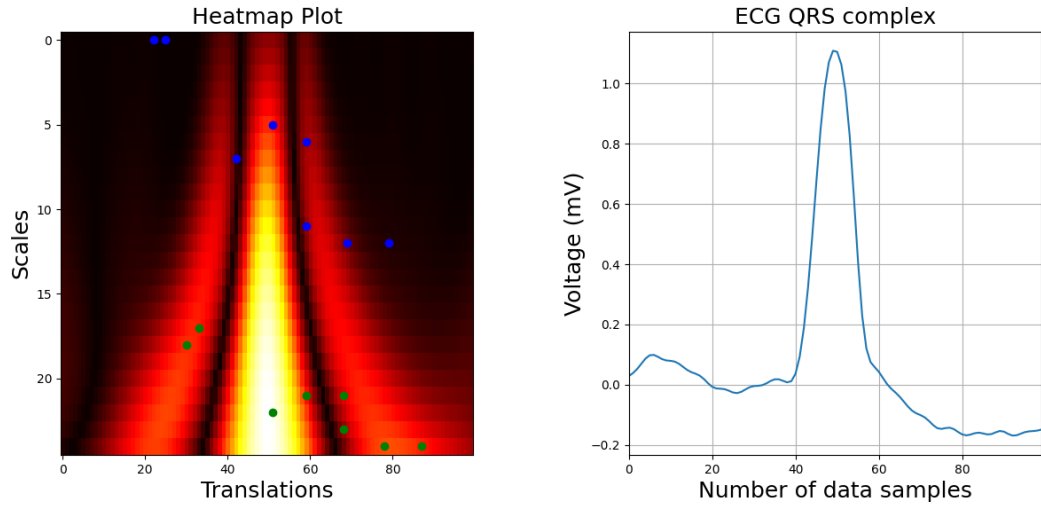


Figure 4.10: Normal ECG heartbeat signal with initial (blue) and trained (green) coefficients.

4.3 SIEMENS bearing fault dataset

We cooperated with an industrial partner (Siemens Digital Industries) and tested our method on their bearing fault dataset. The dataset contains four type of conditions: healthy, inner race fault, outer race fault and ball defect fault. The setup was equipped with two bearing units. In all tests, the left bearing was the healthy one and the right bearing was the defective one. The database includes measurements taken at constant speed and run-up measurements. We selected the 300 rpm measurements. Our labeled datasets were similar to those used for our previous bearing fault experiments (baseline-inner, baseline-outer and inner-outer sets). The bearing signals were recorded with multiple measurement instruments. The best quality recordings were made with an acoustic emission sensor. Following in terms of quality are the measurements recorded with 3D accelerometers and microphones, respectively. The preprocessing steps were made following the considerations described in the open source bearing experiments. The test sets contain 130 elements for each type.

Set	# of cfs.	Mean Acc.	Highest	Lowest
baseline-outer	48	91.92%	100%	67.31%
baseline-inner	48	99.23%	100%	96.15%
inner-outer	48	97.69%	100%	90.39%

Table 4.3: CLASSIFICATION RESULTS OF THE SIEMENS Acoustic Emission BEARING DATASET

Table 4.3 contains the results of the tests evaluated on the acoustic emission measurements. The tests were also performed with 5-fold cross validation. The used network was CWFCNN. Our method achieved high overall accuracy on all dataset.

Table 4.4 shows the results of the baseline-inner dataset compared to the tests recorded with the different sensors. As expected, CWFCNN achieved the best results on the data recorded with the acoustic emission sensor. This is followed by the accelerometer and the microphone measurements.

Set	# of cfs.	Mean Acc.	Highest	Lowest
Acoustic Emission	48	99.23%	100%	96.15%
Accelerometers	48	97.31%	100%	86.54%
Microphone	48	93.08%	100%	82.69%

Table 4.4: CLASSIFICATION RESULTS OF THE SIEMENS baseline-inner
BEARING DATASET

Chapter 5

Conclusion

In this thesis we developed wavelet based machine learning methods. The proposed neural networks approximate continuous wavelet transform coefficients of input signals to achieve sparse representation. Wavelet coefficients allow the construction of explainable AI. We compared our methods with a state-of the-art wavelet based method. The results of the experiments are comparable to other ECG signal classification methods.

In the future, we would like to test the networks with additional analytical wavelets. We want to extend the existing biological and engineering applications in collaboration with industrial partners. We also would like to test our methods on image processing tasks.

Bibliography

- [1] Claude Gasquet and Patrick Witomski. *Fourier analysis and applications: filtering, numerical computation, wavelets*. Vol. 30. Springer Science & Business Media, 2013.
- [2] Tamás Dózsa et al. “Brake Disc Deformation Detection Using Intuitive Feature Extraction and Machine Learning”. In: *Machines* 12.4 (2024), p. 214.
- [3] Tamás Dózsa et al. “Variable projection support vector machines and some applications using adaptive Hermite expansions”. In: *International Journal of Neural Systems* 34.1 (2024).
- [4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [5] Péter Kovács et al. “VPNet: variable projection networks”. In: *International Journal of Neural Systems* 32.01 (2022), p. 2150054.
- [6] Tamás Dózsa et al. “Road abnormality detection using piezoresistive force sensors and adaptive signal models”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–11.
- [7] Tamás Dózsa et al. “Towards intelligent tire development”. In: (2023).
- [8] Tianfu Li et al. “WaveletKernelNet: An interpretable deep neural network for industrial intelligent diagnosis”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.4 (2021), pp. 2302–2312.
- [9] N O’Mahony et al. “One-shot learning for custom identification tasks; a review”. In: *Procedia Manufacturing* 38 (2019), pp. 186–193.

- [10] Yingzhe Lyu et al. “An empirical study of the impact of data splitting decisions on the performance of aiops solutions”. In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30.4 (2021), pp. 1–38.
- [11] Sebastian Bach et al. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation”. In: *PloS one* 10.7 (2015), e0130140.
- [12] John Martinsson and Maria Sandsten. “DMEL: The Differentiable Log-Mel Spectrogram as a Trainable Layer in Neural Networks”. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 5005–5009.
- [13] Penco Petrov Petrushev and Vasil Atanasov Popov. *Rational approximation of real functions*. 28. Cambridge University Press, 2011.
- [14] Gene H Golub and Victor Pereyra. “The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate”. In: *SIAM Journal on numerical analysis* 10.2 (1973), pp. 413–432.
- [15] Carl Taswell. “Computational algorithms for Daubechies least-asymmetric, symmetric, and most-symmetric wavelets”. In: *Proceedings of the International Conference on Signal Processing Applications and Technology*. Citeseer. 1997, pp. 1834–1838.
- [16] Gisbert Stoyan and Agnes Baran. *Elementary numerical mathematics for programmers and engineers*. Springer, 2016.
- [17] <https://pytorch.org/docs/stable/index.html>. 2024.
- [18] Attila Miklos Amon. *WaveletNet*. <https://gitlab.com/tamasdzs/waveletnet/>. 2024.
- [19] Robert B Randall and Jerome Antoni. “Rolling element bearing diagnostics—A tutorial”. In: *Mechanical systems and signal processing* 25.2 (2011), pp. 485–520.
- [20] Eric Bechhoefer. *Condition Based Maintenance Fault Database for Testing Diagnostics and Prognostic Algorithms*. <https://www.mfpt.org/fault-data-sets/>. 2013.

- [21] Li Tianfu. *WaveletKernelNet*. [https : / / github . com / HazeDT / WaveletKernelNet/](https://github.com/HazeDT/WaveletKernelNet/). 2020.
- [22] Ary L Goldberger et al. “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals”. In: *circulation* 101.23 (2000), e215–e220.
- [23] George B Moody and Roger G Mark. “The impact of the MIT-BIH arrhythmia database”. In: *IEEE engineering in medicine and biology magazine* 20.3 (2001), pp. 45–50.
- [24] Eduardo José da S Luz et al. “ECG-based heartbeat classification for arrhythmia detection: A survey”. In: *Computer methods and programs in biomedicine* 127 (2016), pp. 144–164.