Predicting High Frequency Exchange Rates using Machine Learning

ALEKSANDAR PALIKUCA
TIMO SEIDL
Predicting High Frequency Exchange Rates using Machine Learning

ALEKSANDAR PALIKUCA
TIMO SEIDL

Master’s Thesis in Mathematical Statistics (30 ECTS credits)
Master Programme in Applied and Computational Mathematics (120 credits)
Royal Institute of Technology year 2016
Supervisors at at SEB: Dr. Pär Hellström
Supervisor at KTH: Jonas Hallgren
Examiner: Timo Koski

TRITA-MAT-E 2016:19
ISRN-KTH/MAT/E--16/19-SE

Royal Institute of Technology
SCI School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden
URL: www.kth.se/sci
Abstract

This thesis applies a committee of Artificial Neural Networks and Support Vector Machines on high-dimensional, high-frequency EUR/USD exchange rate data in an effort to predict directional market movements on up to a 60 second prediction horizon. The study shows that combining multiple classifiers into a committee produces improved precision relative to the best individual committee members and outperforms previously reported results. A trading simulation implementing the committee classifier yields promising results and highlights the possibility of developing a profitable trading strategy based on the limit order book and historical transactions alone.
Att Förutsäga Högfrekventa Växelkurser med Maskininlärning

Sammanfattning
Denna uppsats tillämpar en kommitté av artificiella neuronnät och stödvektormaskiner på hög-dimensionell, högfrekvent EUR/USD växelkursdata i ett försök att förutsäga marknadsriktning på en upp till 60 sekunders tidshorisont. Studien visar att en kommitté bestående av flera klassificerare ger bättre precision än de bästa enskilda kommittémedlemmarna och överträffar tidigare rapporterade resultat. En handelssimulering där kommittén tillämpas ger lovande resultat och framhåller möjligheten att utveckla en lönsam handelsstrategi baserad på enbart limit order book och historiska transaktioner.
Acknowledgements

We would like to thank Jonas Hallgren, our supervisor at KTH Royal Institute of Technology, for his valuable ideas and guidance throughout the process of writing this thesis. We would also like to express our gratitude to Pär Hellström, Senior Quant Trader at SEB, for his feedback, support and for introducing us to the field of high frequency trading.

Stockholm, May 2016
Aleksandar Palikuca and Timo Seidl
Contents

1 Introduction 1
  1.1 Background ...................................................... 1
  1.2 Previous Studies .............................................. 1
  1.3 Problem Formulation .......................................... 3
  1.4 Outline .......................................................... 3

2 Support Vector Machines 5
  2.1 Separable Classes ............................................. 6
  2.2 Nonseparable Classes ......................................... 10
  2.3 Extension to Nonlinear Decision Boundaries ............... 13
  2.4 The SMO Algorithm ............................................ 16

3 Artificial Neural Networks 19
  3.1 Perceptron ....................................................... 19
  3.2 Neural Network Structure ................................... 21
  3.3 Neural Network Training ..................................... 22
  3.4 Improvements .................................................... 26

4 Ensemble Learning 29
  4.1 Nontrainable Committees .................................... 29
  4.2 Trainable Committees ......................................... 31

5 Implementation 33
  5.1 Raw Data ........................................................ 33
  5.2 Data Preprocessing ............................................ 34
  5.3 Experiment Setup ............................................. 36
  5.4 Support Vector Machines .................................... 37
  5.5 Artificial Neural Networks ................................ 39
  5.6 Committees ...................................................... 40
  5.7 Performance Measurement of Classifiers .................. 42
  5.8 Trading Simulation ............................................ 43

6 Results 47
CONTENTS

6.1 Classification Performance .................................. 47
6.2 Trading Performance ........................................... 49

7 Discussion .......................................................... 53
  7.1 Classification Performance ................................ 53
  7.2 Trading Performance ......................................... 55
  7.3 Future Work and Improvements ............................ 57

8 Conclusion .......................................................... 59

Appendix A Classification Performance .......................... 61
Appendix B Trading Performance ................................ 69
Chapter 1

Introduction

1.1 Background

The foreign exchange market is the largest financial market in the world. With a daily turnover in excess of $5 trillion, it is the backbone of international trade [4]. The market is largely unregulated and the foreign exchange trading is today almost exclusively done over the counter (OTC), with the majority taking place on Electronic Communication Networks (ECNs). ECNs are essentially stock exchanges where market participants can view each others orders and interact to sell or buy currency, using numerous order types ranging from simple limit orders to complex coupled strategies.

In the past decade there have been significant advances in the field of machine learning and artificial intelligence. Much is due to faster computers, but no less because of the abundance of electronic data which has created a testbed for machine learning experiments. As algorithms are starting to outperform their human counterparts in complex tasks [26], there is a growing interest in using machine learning to model and predict market movements; either to hedge risks or to seek trading opportunities.

In this thesis, therefore, machine learning algorithms are used on high-dimensional high frequency data in an effort to identify patterns and predict market movements on up to one minute prediction horizons.

1.2 Previous Studies

A prominent theory is the Efficient Market Hypothesis, which states that historic prices cannot be used to predict future prices. It assumes that the prices fully reflect all available information [8]. There are however studies
that contradict this theory. One study uses genetic programming techniques to find technical trading rules and find strong evidence of significant out-of-sample excess returns for six different exchange rates, over a period of 15 years. Reporting that "...results on the USD/DEM indicate that the trading rules are detecting patterns in the data that are not captured by standard statistical models." [21]. Another study uses Artificial Neural Networks (ANNs) on daily data of the S&P 500 Index. Looking only on previous returns and classifying the predictions in three classes, \{Buy, Hold, Sell\}, it outperforms a simple buy-and-hold strategy [17].

In [14] Support Vector Machines (SVMs) are explored in the context of forecasting weekly movement directions of the NIKKEI 225 Stock Index. The study shows that SVMs are superior to other methods, such as linear discriminant analysis, quadratic discriminant analysis and Elman backpropagation neural networks, and further shows that a combination of these methods outperform each individual method.

In [9] SVMs are used on the EUR/USD exchange rate in a three class classification task to predict if the future bid price will be above the current ask (Buy), the future ask price will be below the current bid (Sell), or none of the above (Hold).

Based on the literature survey conducted in conjunction with this thesis, the impression arises that ANNs and SVMs are the most commonly used off-the-shelf machine learning techniques for financial time series prediction.

A recent study looking at conditional probabilities of market directional movement using tick-by-tick data for the S&P 500 Index finds that trend following price developments have higher probability of occurring than trend reversals [23]. That is, two consecutive movements in the same direction occur with higher probability than movements in the opposite direction. The study indicates that the patterns hold for timespans between 5 seconds and 60 seconds, after which the signal weakens. A similar, older, study confirms the existence of significant conditional probabilities [28] and another study confirms that a fair prediction window is in the order of magnitude of minutes [9].

In summary, there are several studies that indicate that the market is, at least in part, predictable. Arguably, these studies are enough to call the Efficient Market Hypothesis into question. Studies also indicate that a good prediction horizon is in the order of seconds to minutes, and that classification as a way of predicting market movement is feasible.
1.3 Problem Formulation

The available data for the experiment is tick-by-tick data for one week of the EUR/USD exchange rate with a full depth limit order book. To complement the limit order book, data of transaction occurrences is available for the same period containing information when a transaction occurs, at what exchange rate, and whether the transaction is given or paid.

The objective is to predict patterns in the data. The patterns looked for are upward, downward and lateral movements for 5, 20 and 60 second prediction horizons. The methods used are SVMs and ANNs. As a means to improve upon the individual classifier performances, the methods are additionally combined into a committee.

Mathematically, the objective can be formulated as

\[ \hat{y}_T = h(f(x_t)) \]

where \( \hat{y}_T \) is the prediction of the directional movement for the currency pair for the next \( \Delta t \) seconds, such that \( T = t + \Delta t \), based on the information \( x \) available at time \( t \). The committee is a function \( h(x) = w^T x \) that combines multiple classifiers \( f_i(x) \) for \( i = 1, \ldots, K \), where \( K \) is the total number of classifiers used, by a weight vector \( w \). The information \( x \) is a set of features based on prices, volumes and transactions. The evaluation of the classifier is done with accuracy measures as well as through a trading simulation.

1.4 Outline

The thesis is structured as follows. Chapters 2, 3 and 4 cover the theory of Support Vector Machines, Artificial Neural Networks and Ensemble Learning. In Chapter 5 the experiment and the performance measurements are specified in detail. Chapter 6 is dedicated to the results of the classifiers and their trading performances. This is followed by Chapter 7 where the results are discussed in the context of previous studies and real world application. Chapter 8 concludes the thesis.
Chapter 2

Support Vector Machines

A very popular machine learning technique developed by Cortes and Vapnik \cite{7} is the Support Vector Machine (SVM). The SVM belongs to the class of kernel methods as it allows the user to find optimal solutions to an optimization problem in higher dimensions through the kernel without actually having to represent anything in those. In this way solutions that are nonlinear in low dimensional spaces can be found as they can be linear in higher dimensions. This will be expanded upon in Section \ref{section:kernel_trick}. Originally the SVM was developed and used in the context of classification. The general idea of a SVM is to perform a binary classification based on some input matrix $X$ of dimension $N \times D$ by separating the respective $Y$ values of dimension $N \times 1$ consisting of $+1$’s and $-1$’s, representing the two classes, using an hyperplane. Here $N$ is the number of samples used to find such an hyperplane and $D$ is the number of features. The hyperplane is determined by solving a convex optimization problem, which assures that every local optimum is also a global optimum \cite{5}. The determined hyperplane can then be used to classify new inputs $x^*$ of dimension $1 \times D$ depending on which side of the hyperplane the new sample $x^*$ lies.

The theory on SVMs in this thesis is constructed such that cases for which a linear separation of classes is possible are considered first, to then move to cases where the classes are nonseparable by linear decision boundaries. Here a method to circumvent this problem is introduced to eventually turn to nonlinear decision boundaries which require to enter higher dimensional spaces by applying the kernel trick. Lastly a method that deals with faster training of the SVM is touched upon, namely the Sequential Minimization Optimization (SMO) algorithm.

This chapter is mostly based on work by Bishop \cite{5}, Hastie, Tibshirani, and Friedman \cite{10}, Shawe-Taylor and Christianini \cite{25}, and Platt \cite{22}.
2.1 Separable Classes

First the focus is put on cases for which there exists a hyperplane that perfectly separates two classes. A hyperplane in a $D$ dimensional space is a $D - 1$ dimensional subspace that separates the $D$ dimensional space into two halves. For example, in a two-dimensional space, a hyperplane is a one-dimensional subspace, i.e. a line that separates the two-dimensional space in two halves. While hyperplanes in up to three-dimensional spaces are easy to imagine it becomes hard to visualise in any dimension $D > 3$. A $D$ dimensional hyperplane $f$ is defined as

$$f(x) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_Dx_D = 0,$$  \hfill (2.1)

which means that all $x$ that satisfy the condition in (2.1) lie on the hyperplane. The intercept $w_0$ will be called the bias $b$ from here on. $f(x)$ yielding a value $> 0$ in (2.1) for some vector $x$ indicates that the point lies on one side of the hyperplane, while a value $< 0$ indicates that the point lies on the other side of the hyperplane. It already becomes clear why hyperplanes are used as decision boundaries in machine learning. It allows the user to classify inputs into two classes, such that the corresponding class of an input $x$ will be the targets $t_i \in \{+1, -1\}$.

To use an hyperplane as a tool for classification the pair $\{x_i, t_i\}$ has to be created, where $i = 1, \ldots, N$ and $x_i$ is a $D$-dimensional vector for all $i$. Those pairs are considered the training data and will be used in order to determine the hyperplane that separates the $x$ vectors that correspond to $t = +1$ from those that have target values $t = -1$. The $x$ samples that produce a positive value in (2.1) are labeled with the target $t = +1$, while the $x$ samples that produce a negative value in (2.1) are labeled with the target $t = -1$. Thereby the following property holds for a separating hyperplane

$$t_i(b + w_1x_{i1} + \cdots + w_Dx_{iD}) > 0$$

for all $i = 1, \ldots, N$. Unseen samples $x^*$ of dimension $1 \times D$ can then be classified using the computed hyperplane as a decision boundary by evaluating $\text{sign}(f(x^*))$ which indicates the class the unseen sample $x^*$ belongs to.

It can also be shown that the magnitude of the value $|f(x^*)|$ indicates how far away the sample is from the separating hyperplane. To compute the distance of $x^*$ to a hyperplane, the unit normal vector of the hyperplane needs to be determined. Taking the gradient of $f(x)$ from (2.1) yields the normal vector $w$,

$$w = \nabla f(x) = \begin{bmatrix} w_1 \\ \vdots \\ w_D \end{bmatrix}.$$
2.1. SEPARABLE CLASSES

From this the unit normal vector is obtained with $w^* = \frac{w}{||w||}$. A vector from the plane to the point $x^*$ is given by

$$v = \begin{bmatrix} x_1^* - x_1 \\ \vdots \\ x_D^* - x_D \end{bmatrix}$$

and the distance $d$ can then be computed by projecting $v$ onto $w^*$:

$$d = w^{*T}v = \frac{1}{||w||}(w_1(x_1^* - x_1) + \cdots + w_1(x_D^* - x_D))$$

$$= \frac{1}{||w||}(w_1x_1^* + \cdots + w_1x_D^* + b)$$

$$= \frac{1}{||w||}f(x^*).$$

Hence, the signed distance $d$ from the unseen sample $x^*$ to the hyperplane is proportional to $f(x^*)$.

If the two classes of data samples are in fact linearly separable an infinite amount of separating hyperplanes can be found. To determine one unique solution the optimal hyperplane is chosen which is defined as the hyperplane farthest away from the training data. The optimal hyperplane represents the decision boundary. In order to find this particular hyperplane the goal is to maximize the margin $M$, which is defined as the minimal distance of the hyperplane to the closest data point of each class. Since the decision boundary $f(x) = xw + b = 0$ can be scaled with a constant $c > 0$ without changing its functionality, the normal vector $w$ can be scaled such that it holds for the closest point $x_s$ of dimension $1 \times D$ from the decision boundary on either side $|x_sw + b| = 1$. Hence, two additional hyperplanes $h_1$ and $h_2$ that lie parallel to the optimal hyperplane on each side are created and serve to define a region in which no points are allowed to lie

$$h_1(x_1) := x_1w + b = 1$$
$$h_2(x_2) := x_2w + b = -1.$$ 

From this it is inferred that the following must hold for all training samples in order to lie outside of the region created by $h_1$ and $h_2$

$$t_i(x_iw + b) \geq 1, \quad i = 1, \ldots, N.$$ 

Since $h_1$ and $h_2$ have the distance $M$ from the decision boundary the following holds

$$2M = \text{dist}(h_1(x_1), h_2(x_2)) = (x_1 - x_2)w^*$$

$$= \frac{1}{||w||}(x_1w - x_2w) = \frac{1}{||w||}(1 - b - (-1 - b)) = \frac{2}{||w||}.$$
Hence, the goal to maximize the margin $M$ is equivalent to maximizing $\frac{1}{||w||}$ which is in turn equivalent to the optimization problem

$$\min_{w,b} \frac{1}{2}||w||^2$$

subject to $t_i(x_iw + b) \geq 1, \ i = 1, \ldots, N,$

for $N$ training samples $x_i$ of dimension $1 \times D$ and $w = (w_1, \ldots, w_D)^T$.

As seen in Figure 2.1, the constraints in (2.3) assure an empty band around the decision boundary of width $M = 1/||w||$ which is bordered by the maximum margin hyperplanes $h_1$ and $h_2$. This band is maximized by solving (2.2) w.r.t $w$ and $b$. The optimization problem set up in (2.2) is a quadratic programming problem that can be solved using the method of Lagrange multipliers such that the Lagrangian function

$$L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} a_i(t_i(x_iw + b) - 1)$$

(2.4)

is obtained, where the vector $a = (a_1, \ldots, a_N)^T > 0$, contains $N$ Lagrange multipliers. Setting the partial derivatives equal to zero results in the con-
2.1. SEPARABLE CLASSES

ditions for all $i$

$$w = \sum_{i=1}^{N} a_i t_i x_i^T, \quad (2.5)$$

$$0 = \sum_{i=1}^{N} a_i t_i. \quad (2.6)$$

Plugging those conditions in into (2.4) allows to eliminate $w$ from the dual representation and yields the following optimization problem:

$$\max_{\mathbf{a}} \tilde{L}(\mathbf{a}) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_i a_m t_i t_m x_i^T x_m \quad (2.7)$$

subject to $a_i \geq 0, \quad i = 1, \ldots, N$ \hspace{1cm} (2.8)

$$\sum_{i=1}^{N} a_i t_i = 0. \quad (2.9)$$

A constrained optimization problem of this form must satisfy the Karush-Kuhn-Tucker (KKT) conditions [10] which are composed by the conditions (2.5), (2.8), (2.9) and

$$a_i [t_i (x_i w + b) - 1] = 0 \quad \forall i. \quad (2.10)$$

From the last KKT condition (2.10) the following relations hold

1. $a_i > 0 \Rightarrow t_i (x_i w + b) = 1$
2. $t_i (x_i w + b) > 1 \Rightarrow a_i = 0$.

Here the first relation reveals that if the Lagrangian multiplier is larger than 0, the respective training sample will lie on the edge of the band, i.e. will lie on one of the maximum margin hyperplanes, see Figure 2.1. Moreover the second relation says that if a training sample $x_i$ lies farther away from the separating hyperplane than the margin $M$, the respective Lagrangian multiplier $a_i$ will be equal to 0. From the condition (2.5) it becomes clear that only the vectors that have a corresponding Lagrangian multiplier $a_i > 0$ will contribute to the solution. Hence, those vectors are called support vectors. Having obtained the set of support vectors and from (2.5) the weights of the separating hyperplane it is time to solve for the bias $b$ in $f(x_i) = x_i w + b$. This can be done by solving (2.10) for one support vector, i.e. for a training sample $x_i$ for which the corresponding Lagrangian multiplier $a_i > 0$. Alternatively $b$ is found by taking the average of all the solutions obtained for each support vector when solving (2.10), i.e.

$$b = \frac{1}{N_S} \sum_{i \in S} (t_i - x_i w), \quad (2.10)$$
with \( S \) denoting the set of indices of support vectors and \( N_S \) denoting the total number of support vectors. Now it is possible to classify a new unseen data sample \( x^* \) by evaluating
\[
\text{sign}(f(x^*)) = \text{sign}(x^*w + b).
\]

### 2.2 Nonseparable Classes

While in Section 2.1 the focus was on cases for which separating hyperplanes exist, cases for which those hyperplanes do not exist, i.e. the two classes are not separable by a linear boundary, are now investigated. This means that classes will overlap and a satisfying decision boundary is to be found that minimizes the number of those misclassified vectors.

While in the separable case in Section 2.1 a decision boundary was found such that \( f(x_i) = x_iw + b > 0 \ \forall i \), this is not possible with overlapping classes. Therefore a method is chosen that allows some vectors to be on the wrong side of the margin, i.e. on the correct side of the decision boundary but within the band, or falsely classified on the wrong side of the decision boundary. Slack variables \( \xi = (\xi_1, \xi_2, \ldots, \xi_N) > 0 \) are introduced and the constraint in (2.3) is modified to
\[
t_i(x_iw + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N. \tag{2.11}
\]
Furthermore another constant \( C \) is introduced that regularizes the total slack allowed
\[
\sum_{i=1}^{N} \xi_i \leq C. \tag{2.12}
\]
From (2.11) it is seen that misclassifications occur if \( \xi_i > 1 \), while a vector \( x_i \) is correctly classified but on the wrong side of the maximum margin hyperplane, i.e. in the band, if \( 1 > \xi_i > 0 \). Points for which \( \xi_i = 0 \) are correctly classified and on the right side of the margin or on the margin. Therefore the optimization problem that needs to be solved now will be to maximize the margin while keeping the constraints from before but adding the constraints (2.11) and (2.12). Analogous to Section 2.1 the optimization problem will look as follows
\[
\begin{align*}
\min_{w, b} & \quad \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i \\
\text{subject to} & \quad t_i(x_iw + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N \tag{2.14} \\
& \quad \xi_i \geq 0 \quad i = 1, \ldots, N. \tag{2.15}
\end{align*}
\]
2.2. NONSEPARABLE CLASSES

As before this represents a quadratic optimization problem with linear inequality constraints which makes it a convex problem \[10\]. This implies again that any local optimum is also a global optimum. From (2.13) it is seen that the parameter \( C > 0 \) regularizes the extent of misclassifications allowed. If \( C \to \infty \) the slack variables \( \xi_i \) are heavily penalized and the minimization problem is equivalent to the separable case in Section 2.1 where no misclassifications are allowed. In Figure 2.2 two decision boundaries that are obtained by solving the optimization problem in (2.13) for different values of \( C \) are presented for the Iris flower data set. The solution in the left graph of Figure 2.2 allows fewer points to be misclassified than the solution in the right of Figure 2.2 which results in the left solution having a smaller margin than the right solution. It is noted that this can lead to overfitting and does not necessarily perform better when classifying unseen data.

![Figure 2.2: Two linearly nonseparable classes separated by a decision boundary where the penalization parameter is set to \( C = 1000 \) in the left graph and \( C = 1 \) in the right graph.](image)

Proceeding in the same manner as before in order to solve the optimization problem this results in the Lagrange function

\[
L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} a_i(t_i(x_iw + b) - (1 - \xi_i)) + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \mu_i \xi_i
\]

(2.16)

which in turn yields the partial derivatives that we set to zero. Doing so
results in the following three equalities:

\[ w = \sum_{i=1}^{N} a_i t_i x_i^T, \quad (2.17) \]
\[ 0 = \sum_{i=1}^{N} a_i t_i, \quad (2.18) \]
\[ a_i = C - \mu_i \quad i = 1, \ldots, N. \quad (2.19) \]

Those constraints must hold for all \( i \) along the constraints for the slack variables \( \xi_i > 0 \) and the constraints for the Lagrange multipliers \( a_i > 0 \) and \( \mu_i > 0 \), for all \( i = 1, \ldots, N \). Then the dual Lagrangian representation is obtained by plugging these constraints into (2.16):

\[ \tilde{L}(a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_i a_m t_i t_m x_i^T x_m. \quad (2.20) \]

This dual representation is in fact equivalent to the dual representation derived for the separable case in (2.7), however the constraints in the optimization differ. Hence, the quadratic programming problem looks as follows

\[ \max_a \quad \tilde{L}(a) \quad (2.21) \]
\[ \text{subject to} \quad 0 \leq a_i \leq C, \quad i = 1, \ldots, N \quad (2.22) \]
\[ 0 = \sum_{i=1}^{N} a_i t_i. \quad (2.23) \]

Here the constraint (2.22) is reformulated from the two constrained Lagrange multipliers \( a_i > 0 \) and \( \mu_i > 0 \), for all \( i \) and the Equation (2.19). Similarly as in Section 2.1 the KKT conditions comprise (2.17) - (2.19), and

\[ a_i [t_i f(x_i) - (1 - \xi_i)] = 0 \quad (2.24) \]
\[ \mu_i \xi_i = 0 \quad (2.25) \]
\[ t_i f(x_i) - (1 - \xi_i) \geq 0, \quad (2.26) \]

for \( i = 1, \ldots, N \). As before in Section 2.1, the equality in (2.17) shows that only the training samples corresponding to a Lagrange multiplier \( a_i > 0 \) will be considered when computing the predictive model. As mentioned before those are called support vectors and from (2.24) it is implied that

\[ t_i f(x_i) = (1 - \xi_i) \quad (2.27) \]

must be satisfied. Furthermore if \( a_i < C \), (2.19) shows that \( \mu_i > 0 \) must hold, which in turn implies that \( \xi_i = 0 \) by (2.25). Therefore by (2.27),
2.3 Extension to Nonlinear Decision Boundaries

Thus far, linear decision boundaries were investigated for separable and non-separable classes in Section 2.1 and 2.2, respectively. In many practical applications the classification problem is not linear and therefore an optimal decision boundary using the approaches discussed cannot be found. To circumvent this problem the data can be projected into a higher dimensional space by a fixed feature-space transformation \( \phi \), such that

\[
\phi : \mathcal{X} \rightarrow \mathcal{F}, \, x \rightarrow \phi(x),
\]

where \( \mathcal{X} \) is the input space and \( \mathcal{F} \) is the feature space and is of higher dimension than \( \mathcal{X} \). In this enlarged space the chance of finding linear boundaries is higher. Those solutions in turn translate to nonlinear boundaries in the original space. SVMs are very well suited to incorporate the transformation \( \phi \) elegantly. In order to do so the dual representation in (2.20) is rewritten using the inner product of the data points \( x_i \) for all \( i \), i.e.

\[
L(a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_i a_m t_i t_m \langle x_i, x_m \rangle.
\]

Furthermore the function used to evaluate unseen samples \( x^* \) can be rewritten in the following manner

\[
f(x^*) = x^* w + b = \left( \sum_{n=1}^{N} a_n t_n x_n \right) x^T + b
\]

\[
= \sum_{n=1}^{N} a_n t_n \langle x_n, x^* \rangle + b.
\]
where \(2.17\) was used. This shows that when training as well as predicting with SVMs the problems to solve can be written in terms of inner products between input feature vectors. This paves the way for incorporating high dimensional transformations \(\phi\) as it shows that the higher dimension never has to be entered but that it is rather accessed by computing inner products. The feature mapping \(\phi\) is used by simply replacing the former feature vectors \(x\) by the transformed feature vectors \(\phi(x)\) everywhere. This results in the new Lagrange dual function

\[
\tilde{L}(a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_i a_m t_i t_m \langle \phi(x_i), \phi(x_m) \rangle \tag{2.29}
\]

and the new predictive function

\[
f(x^*) = \sum_{n=1}^{N} a_n t_n \langle \phi(x_n), \phi(x^*) \rangle + b. \tag{2.30}
\]

(2.30) and (2.29) clarify that training as well as predicting with SVMs can be done by finding optimal solutions in high-dimensional spaces without having to specify the mapping \(\phi\) but only the kernel function which is defined as a function

\[k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R},\]

and is assumed to satisfy [25]

\[k(x, z) = \langle \phi(x), \phi(z) \rangle,\]

for all \(x, z \in \mathcal{X}\). Here, \(\phi\) is a mapping from \(\mathcal{X}\) to a (inner product) feature space \(\mathcal{F}\) as defined in (2.28). Note that the input space \(\mathcal{X}\) corresponds to the space \(\mathbb{R}^D\) used before. The kernel function \(k\) does the job of computing the inner product in the transformed space. To demonstrate the functionality of the kernel function, the 2nd degree polynomial kernel

\[k(z, u) = (1 + \langle z, u \rangle)^2 = (1 + z_1 u_1 + z_2 u_2)^2 = 1 + 2 z_1 u_1 + 2 z_2 u_2 + 2 z_1 u_1 z_2 u_2 + z_1^2 u_1^2 + z_2^2 u_2^2 = \langle (1, \sqrt{2} z_1, \sqrt{2} z_2, z_1^2, z_2^2), (1, \sqrt{2} u_1, \sqrt{2} u_2, \sqrt{2} u_1 u_2, u_1^2, u_2^2) \rangle = \langle \phi(z), \phi(u) \rangle.\]

With the defined kernel function \(k\) the Gram matrix \(K\) can be introduced which is a \(N \times N\) matrix consisting of the kernel function evaluated at every element, i.e. on all pairs of data points [25]

\[K_{nm} = k(x_n, x_m), \tag{2.31}\]
where \( x_n \) and \( x_m \) are of dimension \( 1 \times D \), with \( n, m = 1, \ldots, N \). The Gram matrix \( K \) from (2.31) is referred to as the kernel matrix and is symmetric and positive semi-definite [25]. Using the kernel matrix to rewrite (2.29) yields the Lagrange dual function

\[
\tilde{L}(a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_i a_m t_i t_m K_{im}.
\]

The positive semi-definiteness and symmetry of the kernel matrix \( K \) assure the Lagrangian function \( \tilde{L}(a) \) to be bounded below and hence present a well defined optimization problem [5].

Typical choices of kernel functions that have proven to be valuable in many practical applications in the past include:

- Linear: \( k(x_i, x_j) = \langle x_i, x_j \rangle \)
- Radial Basis Function: \( k(x_i, x_j) = \exp \left(-\gamma \|x_i - x_j\|^2\right) \)
- \( d \)th degree Polynomial: \( k(x_i, x_j) = \left(1 + \langle x_i, x_j \rangle\right)^d \),

where \( \gamma \) is a free parameter in the radial basis function. This parameter is at times also written as \( \gamma = 1/(2\sigma^2) \), such as in [25] where it is stated that a small \( \sigma \) risks overfitting while a large \( \sigma \) makes it impossible to learn non-trivial classifier. The linear kernel would simply result in the problem formulation discussed in Sections 2.1 and 2.2, while the others allow the user to search for nonlinear decision boundaries.

Incorporating the kernel function in the Lagrange function in (2.29) results in the following QP problem that needs to be solved in order to train an SVM:

\[
\max_a \quad \tilde{L}(a) = \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{N} a_i a_m t_i t_m K_{im} \tag{2.32}
\]

subject to \( 0 \leq a_i \leq C, \quad i = 1, \ldots, N \) \hspace{1cm} \tag{2.33}
\[
0 = \sum_{i=1}^{N} a_i t_i \tag{2.34}
\]

With the obtained Lagrange multipliers \( a \) an unseen sample \( x^* \) can then be classified by evaluating

\[
f(x^*) = \sum_{n=1}^{N} a_n t_n K_{ns} + b. \tag{2.35}
\]

This underlines one strength of SVMs, namely the fast prediction of unseen data as the prediction only depends on the support vectors for which \( a_i > 0 \)
CHAPTER 2. SUPPORT VECTOR MACHINES

and hence the unseen sample only needs to be evaluated at a subset of the training data.

In Figure 2.3 the application of the theory discussed in Section 2.3 is demonstrated. Contrary to the previous solutions in Figures 2.1 and 2.2 a nonlinear decision boundary is obtained from solving the QP problem in (2.32) using the radial basis kernel function to create the kernel matrix $K$.

![Figure 2.3: Two linearly separable classes separated by a decision boundary that is created from solving the optimization problem in (2.32) for the publicly available Iris flower data set. The radial basis kernel function is used with the parameters $C = 1$ and $\gamma = 1$](image)

### 2.4 The SMO Algorithm

The QP problem to be solved in (2.32) is of space complexity $O(N^2)$, where $N$ is the number of training samples, if solved with standard QP techniques. It is obvious that this is not feasible for large training sets. In order to circumvent this problem the Sequential Minimal Optimization (SMO) algorithm, developed by Platt [22], is used which requires a memory linear rather than quadratic to the training set size. Instead of solving the QP problem in (2.32) with respect to all $a_i$, $i = 1, \ldots, N$ at the same time, the SMO decomposes the problem and solves a subset of Lagrangian multipliers at a time. Since a single Lagrangian multiplier is predetermined by the constraint (2.34) the minimum number of Lagrangian multipliers that one can optimize for at a time is 2. This is what the SMO algorithm does, it optimizes the Lagrangian function (2.32) with respect to two multipliers and then updates
the SVM to obtain a new optimal vector of Lagrangian multipliers $a$. A solution of the optimization problem with respect to two Lagrangian multipliers can be derived analytically. The algorithm continues this procedure with another pair of multipliers until convergence is reached. When applying the SMO algorithm the computer only has to store an additional $2 \times 2$ matrix instead of the full matrix for the classical approach. In order to use the algorithm it remains to show how the SMO algorithm solves the optimization problem with respect to two Lagrangian multipliers, how after doing so the SMO algorithm selects the next pair of Lagrangian multipliers to optimize and when the SMO algorithm stops iterating. As this is not the focus of this thesis it is referred to the original paper by Platt [22].
Chapter 3

Artificial Neural Networks

Neural networks have its origin in attempts to model the biological brain. The perceptron, the simplest version of a neural network, was developed by Frank Rosenblatt in the late 1950’s based on work done by Warren McCulloch and Walter Pitts in the 1940’s. In recent years the popularity of neural networks have increased significantly.

Neural networks are a class of universal approximators, meaning that they are capable of approximating any measurable function to any desired degree of accuracy \cite{12}. What follows is the theory of Artificial Neural Networks (ANNs). For further reading, see \cite{5,6}.

3.1 Perceptron

The building block of the neural network is the neuron. A neuron consists of inputs $x_i$, bias $b$, weights $w_i$, a summation function, an activation function $\phi$ and an output $y$. The inputs are multiplied by weights and passed through an activation function to generate an output, or activation, as shown in Figure \[3.1\].

The original neuron was the perceptron, which had a simple step function as its activation function. The perceptron is able to perform binary classification on linearly separable data

$$y = \begin{cases} 1 & \text{if } w_1x_1 + w_2x_2...w_nx_n + b > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

Later on, other activation functions were introduced, in part to allow for more complex models that can handle nonlinearity, but also to allow for training of the neural network. Now, different activation functions denote
different type of neurons, where the most common ones are neurons with a logistic activation function

$$\phi(x) = \frac{1}{1 + e^{-x}},$$ (3.2)

or a hyperbolic tangent activation function

$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \tanh(x).$$ (3.3)

These specific activation functions are popular largely because their derivatives have some nice properties that are helpful during training of the neural network. This is covered in more detail in Section 3.3.
3.2 Neural Network Structure

As shown in Figure 3.3, the network is constructed by layers of neurons, where each neuron in each layer is connected to each neuron in the next layer. In the feed forward architecture of a neural network, the information always flows in one direction, from the input layer to the hidden layer and from the hidden layer to the output layer.

\[ \begin{array}{c}
\text{Input} \\
\text{layer} \\
\end{array} \quad \begin{array}{c}
\text{Hidden} \\
\text{layer} \\
\end{array} \quad \begin{array}{c}
\text{Output} \\
\text{layer} \\
\end{array} \]

\[ \begin{array}{c}
\text{Input} \\
\text{layer} \\
\end{array} \quad \begin{array}{c}
\text{Hidden} \\
\text{layer} \\
\end{array} \quad \begin{array}{c}
\text{Output} \\
\text{layer} \\
\end{array} \]

Figure 3.3: A diagram of a neural network.

Let \((x^i, y^i)\) denote the \(i\):th data sample, with \(x \in \mathbb{R}^n\) being the inputs in the input layer and \(y\) the target value for the output layer. The flow through the neural network can then be structured as follows:

For notational consistency let

\[
\begin{align*}
\mathbf{a}^{(1)}_1 &= x^i_1, \\
\mathbf{a}^{(1)}_2 &= x^i_2, \\
& \vdots \\
\mathbf{a}^{(1)}_n &= x^i_n,
\end{align*}
\]

(3.4)

where \(\mathbf{a}^{(1)}\) is the vector of activations of the first layer. The activations of
the second, hidden, layer is calculated by

\[
\begin{align*}
  a_1^{(2)} &= \phi(w_{10}^{(1)} a_0^{(1)} + w_{11}^{(1)} a_1^{(1)} + \ldots + w_{1m}^{(1)} a_m^{(1)}), \\
  a_2^{(2)} &= \phi(w_{20}^{(1)} a_0^{(1)} + w_{21}^{(1)} a_1^{(1)} + \ldots + w_{2m}^{(1)} a_m^{(1)}), \\
  &\vdots \\
  a_m^{(2)} &= \phi(w_{m0}^{(1)} a_0^{(1)} + w_{m1}^{(1)} a_1^{(1)} + \ldots + w_{mm}^{(1)} a_m^{(1)}),
\end{align*}
\]  

(3.5)

where \(a_0^{(1)}\) is an added bias term and \(m\) denotes the number of neurons in the hidden layer. Finally the output layer is calculated by

\[
\begin{align*}
  a_1^{(3)} &= \phi(w_{10}^{(2)} a_0^{(2)} + w_{11}^{(2)} a_1^{(2)} + \ldots + w_{1m}^{(2)} a_m^{(2)}), \\
  a_2^{(3)} &= \phi(w_{20}^{(2)} a_0^{(2)} + w_{21}^{(2)} a_1^{(2)} + \ldots + w_{2m}^{(2)} a_m^{(2)}), \\
  &\vdots \\
  a_k^{(3)} &= \phi(w_{k0}^{(2)} a_0^{(2)} + w_{k1}^{(2)} a_1^{(2)} + \ldots + w_{km}^{(2)} a_m^{(2)}),
\end{align*}
\]  

(3.6)

where again \(a_0^{(2)}\) is an added bias term and \(k\) denotes the number of neurons in the output layer. With

\[
\begin{align*}
  x^i &= \begin{bmatrix} x_0^i \\ x_1^i \\ \vdots \\ x_n^i \end{bmatrix}, \quad W^{(1)} = \begin{bmatrix}
    w_{10}^{(1)} & w_{11}^{(1)} & \cdots & w_{1m}^{(1)} \\
    w_{20}^{(1)} & w_{21}^{(1)} & \cdots & w_{2m}^{(1)} \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{m0}^{(1)} & w_{m1}^{(1)} & \cdots & w_{mm}^{(1)}
  \end{bmatrix}, \quad W^{(2)} = \begin{bmatrix}
    w_{10}^{(2)} & w_{11}^{(2)} & \cdots & w_{1m}^{(2)} \\
    w_{20}^{(2)} & w_{21}^{(2)} & \cdots & w_{2m}^{(2)} \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{k0}^{(2)} & w_{k1}^{(2)} & \cdots & w_{km}^{(2)}
  \end{bmatrix}.
\end{align*}
\]

Equation 3.4, 3.5 and 3.6 can be written as \(a^{(1)} = x^i\), \(a^{(2)} = \phi(z^{(2)})\) and \(a^{(3)} = \phi(z^{(3)})\) where \(z^{(2)} = W^{(1)} a^{(1)}\) and \(z^{(3)} = W^{(2)} a^{(2)}\), and the activation function \(\phi\) is applied element wise.

Summarizing, given a sample input \(x^i\), the neural network produces the hypothesis

\[
h_W(x^i) = a^{(3)} = \phi(W^{(2)} a^{(2)}) = \phi(W^{(2)} \phi(W^{(1)} a^{(1)})) = \phi(W^{(2)} \phi(W^{(1)} x^i)).
\]  

(3.7)

This process of computing the hypothesis is called forward propagation. In order for the hypothesis to be accurate, that is, for \(h_W(x^i) \approx y^i\), the network needs to be trained. This is done in part by a process called backpropagation and is covered in the next section.

### 3.3 Neural Network Training

In order to train a neural network a cost function needs to be specified. The cost function expresses the difference between actual value and predicted value and the idea of minimizing such a function is the objective of most supervised learning algorithms.
3.3. NEURAL NETWORK TRAINING

3.3.1 Cost functions

The most commonly used cost function are the quadratic cost function

\[ J_i(W, x^i, y^i) = \frac{1}{2} \sum_{k=1}^{K} (h_W(x^i)_k - y^i_k)^2 \]

and the cross entropy cost function

\[ J_i(W, x^i, y^i) = -\sum_{k=1}^{K} y^i_k \log(h_W(x^i)_k) + (1 - y^i_k) \log(1 - h_W(x^i)_k), \quad (3.8) \]

where the total cost over total samples \( S \) is

\[ J(W) = \frac{1}{S} \sum_{i=1}^{S} J_i(W, x^i, y^i). \]

Here \( K \) denotes the dimension of the output layer. In the case of regression, where \( y \in \mathbb{R} \), or in the case of binary classification, where \( y \in \{0, 1\}, K = 1 \), the quadratic cost function is used. For multi-class classification, with \( K \geq 3 \), the classes are expressed as \( K \)-dimensional unit vectors, i.e.

\[
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}, \quad \begin{bmatrix}
0 \\
1 \\
\vdots \\
0
\end{bmatrix}, \quad \cdots, \quad \begin{bmatrix}
0 \\
0 \\
\vdots \\
1
\end{bmatrix},
\]

where each output of the neural network corresponds to one class. In that case, the cross entropy cost function is used.

3.3.2 Gradient Descent

In order to minimize the cost function the weights in the neural network need to be adjusted. There are multiple optimization algorithms for calculating which weights to adjust and how much to adjust them by. One of the most common, and simple, algorithms is the gradient descent algorithm \[5\]. The gradient descent algorithm is an optimization method used to find local minima of a function by taking steps in the negative direction of the functions gradient. In the context of the cost function of the neural network, the gradient descent algorithm iteratively moves towards a set of weights in the weight parameter space that minimizes the cost function. It is defined as

\[ w_{kj}^{(l)}(n + 1) = w_{kj}^{(l)}(n) - \eta \frac{\partial J(n)}{\partial w_{kj}^{(l)}(n)}, \]
where \( \eta \) is the learning rate, controlling the magnitude of the networks weight change.

In regular gradient descent, often called batch gradient descent, the whole training set has to be used before an update of the weights is made. For large training sets this method can become computationally expensive, therefore a slightly different approach is suggested, a variation of the gradient descent algorithm called stochastic gradient descent (SGD) \[5\]. SGD updates the weights after each observation, and as the name suggests, for each iteration it selects a sample at random.

\[
w^{(l)}_{kj}(n + 1) = w^{(l)}_{kj}(n) - \eta \frac{\partial J_i(n)}{\partial w^{(l)}_{kj}(n)}. \quad (3.9)
\]

Advantages of the SGD over the batch gradient descent is that it is computationally faster and less prone to get stuck in local minima.

### 3.3.3 Backpropagation Algorithm

To finalize the training of the neural network, the gradient needs to be calculated. That is, the partial derivatives of the cost function with respect to the weights need to be calculated. This is where the backpropagation algorithm comes in to the picture. The backpropagation algorithm gained popularity after a paper by Rumelhart et. al. in 1986 \[24\] was published and has been an essential part of training neural networks since. It should be noted that the term backpropagation means different things in different texts, so in order to clarify the terminology, in this thesis backpropagation refers to the method of evaluating the derivatives of the cost function.

The partial derivative of the error function with respect to the different weights is an exercise in the chain rule. Let \( J(W) \) define the cross entropy cost function. First, for the weights in the output layer the partial derivatives are calculated by

\[
\frac{\partial J}{\partial w^{(l)}_{kj}} = \frac{\partial J}{\partial a^{(3)}_k} \frac{\partial a^{(3)}_k}{\partial z^{(3)}_k} \frac{\partial z^{(3)}_k}{\partial w^{(l)}_{kj}},
\]

where

\[
\frac{\partial J}{\partial a^{(3)}_k} = \frac{y_k}{a^{(3)}_k} + \frac{1 - y_k}{1 - a^{(3)}_k} = \frac{a^{(3)}_k - y_k}{a^{(3)}_k (1 - a^{(3)}_k)}, \quad (3.10)
\]

\[
\frac{\partial a^{(3)}_k}{\partial z^{(3)}_k} = a^{(3)}_k (1 - a^{(3)}_k), \quad (3.11)
\]
3.3. NEURAL NETWORK TRAINING

\[
\frac{\partial z_k^{(3)}}{\partial w_{kj}^{(2)}} = a_j^{(2)} ,
\]
resulting in
\[
\frac{\partial J}{\partial w_{kj}^{(2)}} = (a_k^{(3)} - y_k)a_j^{(2)} .
\] (3.12)

Note that Equation 3.11 is specifically for the logistic activation function. In calculating the partial derivatives of the cost function with respect to the hidden layer weights, first observe that
\[
\frac{\partial J}{\partial z_j^{(2)}} = \sum_{k=1}^{K} \frac{\partial J}{\partial z_k^{(3)}} \frac{\partial z_k^{(3)}}{\partial a_j^{(2)}} \frac{\partial a_j^{(2)}}{\partial z_j^{(2)}} ,
\]
where
\[
\frac{\partial J}{\partial z_k^{(3)}} = (a_k^{(3)} - y_k)
\]
is calculated from Equation 3.10 and 3.11 and
\[
\frac{\partial z_k^{(3)}}{\partial a_j^{(2)}} = w_{kj}^{(2)} ,
\]
\[
\frac{\partial a_j^{(2)}}{\partial z_j^{(2)}} = (1 - (a_j^{(2)})^2) .
\] (3.13)

Here Equation 3.13 is specifically for the hyperbolic tangent activation function.

Now, the partial derivative of the cost function with respect to the hidden layer weights is calculated by
\[
\frac{\partial J}{\partial w_{ji}^{(1)}} = \frac{\partial J}{\partial z_j^{(2)}} \frac{\partial z_j^{(2)}}{\partial w_{ji}^{(1)}} = \sum_{k=1}^{K} (a_k^{(3)} - y_k)w_{kj}^{(2)} (1 - (a_j^{(2)})^2) a_i^{(1)} .
\] (3.14)

3.3.4 Summary of the Training Algorithm

The network is trained through forward propagation, where Equation 3.7 is applied on each sample \(x^i\) and a cost is calculated using the cross entropy cost function defined in (3.8). The backward propagation consists of first calculating the partial derivatives of the cost with respect to the weights using (3.12) and (3.14), for the output and hidden layer, respectively. These partial derivatives are then used together with the stochastic gradient descent algorithm defined in (3.9) to adjust the weights in an effort to minimize the error with respect to the target class \(x^i\).
3.4 Improvements

Several improvements have been suggested over the years on how to either speed up the learning process of the neural network or create a better classifier \[15\]. In this subsection some of the most common procedures are introduced.

3.4.1 Regularization

A method of improving the generalization of the network, and avoid overfitting is by including a regularization term to the cost function that penalizes large weights. The idea is that smaller weights translates to a simpler hypothesis and therefore prevents overfitting. A common variant of the regularization \[5\] is

$$J_i(W, x^i, y^i)_{\text{new}} = J_i(W, x^i, y^i)_{\text{old}} + \frac{\lambda}{2} \sum_{l=1}^{L-1} \sum_{j=1}^{s_l} \sum_{k=1}^{s_{l+1}} (w^{(l)}_{kj})^2$$

where $\lambda$ is the regularization parameter, $L$ denotes the number of layers and $s_i$ denotes the number of neurons in the $i$th layer. By convention the bias weights are not included for regularization, but in practice it makes little difference.

3.4.2 Validation

For some cases, the data is divided into a training set and a validation set, where the validation set is used to test how well the neural network generalizes on unseen data and as an early stopping mechanism, to limit overfitting, when the training continues to increase the accuracy on the training set but starts to decrease on the validation set. \[6\]

3.4.3 Momentum

The smaller the learning rate parameter $\eta$ the smaller the changes in the weights for each iteration and the longer it takes for the network to learn. If, however, the learning rate is set to high, the weight updates might oscillate and the network become unstable. A way to smooth out the updates is by including a momentum term. Given that the weight update rule without momentum is

$$w^{(l)}_{kj}(n + 1) = w^{(l)}_{kj}(n) + \Delta w^{(l)}_{kj}(n)$$

where $\lambda$ is the regularization parameter, $L$ denotes the number of layers and $s_i$ denotes the number of neurons in the $i$th layer. By convention the bias weights are not included for regularization, but in practice it makes little difference.
where previously

$$\Delta w_{kj}^{(l)}(n) = -\eta \frac{\partial J_k(n)}{\partial w_{kj}^{(l)}(n)}$$

now with momentum the update rule becomes [6]

$$\Delta w_{kj}^{(l)}(n) = -\eta \frac{\partial J_k(n)}{\partial w_{kj}^{(l)}(n)} + \alpha \Delta w_{kj}^{(l)}(n-1)$$

The momentum adds inertia to the motion through the weight parameter space, adding a damping effect to potential oscillations, which can lead to faster convergence. The constant $\alpha \in [0,1]$ regulates the impact of the momentum term. [6]

### 3.4.4 Randomness of Neural Networks

The training of a neural network involves some elements of randomness, beginning with the selection of the learning rate and the number of hidden neurons, but also in the initialization of the weights. While there are some rules of thumb introduced in the machine learning literature, there is no standard procedure in selecting the hyperparameters and the search for the optimal hyperparameters is a study in and of itself, yielding different results for different experiment.
Chapter 4

Ensemble Learning

It has been shown, both in theory and in practice, that improved performance can be obtained by combining multiple models together into a committee, rather than using just a single model \[15, 16\]. It provides robustness, especially to classifiers such as neural networks where randomness is inherent in the algorithm.

A committee’s performance rely on having multiple good models with sufficiently high diversity \[16\]. Diversity denotes the correlation in the errors of the models and can be measured in several different ways, one of which is the simple correlation metric, defined by (4.1), where a low correlation indicates a high diversity and vice versa.

\[
\rho_{x,y} = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y}
\]  

(4.1)

In ensemble learning theory, the individual classifiers are called base learners, or level-0 models, and the algorithm that combines the base learners into a committee is called meta learner, or level-1 model. The base learners can be either multiple sets of the same learning algorithm or a combination of different learning algorithms. In this study the base learners consist of Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs), both of which have been covered in the previous chapters. The following is an overview of the meta learners.

4.1 Nontrainable Committees

The taxonomy of committees can be divided into two classes, trainable committees and nontrainable committees \[16\]. For the nontrainable committees, the meta learner needs no training after the individual base learners have
been trained. For those types of committees, the meta learner is simply a majority vote or a plurality vote. The structure of a nontrainable committee is illustrated in Figure 4.1.

4.1.1 Plurality Vote

The simplest way to construct a committee is by plurality vote. Let $x$ denote the feature set, $y_i(x), i = 1, ..., K$ the base learners and $c \in \{C_1, C_2, ..., C_n\}$ the classes. Then the plurality vote is defined as

$$\hat{y} = \arg \max_c \sum_{i=1}^{K} 1(y_i(x) = c)$$

(4.2)

where $1(\cdot)$ is the indicator function and $\hat{y}$ is the prediction of the committee.

4.1.2 Simple Majority Vote

In the case of binary classification with odd number of base learners, the plurality vote and simple majority vote coincide. For classification of $\geq 3$ classes, the majority vote needs to have more than 50% of the total votes.

$$\hat{y} = \arg \max_c \sum_{i=1}^{K} 1(y_i(x) = c), \text{ such that } \sum_{i=1}^{K} 1(y_i(x) = c^*) > \frac{1}{2} K$$

(4.3)

Equation (4.3) can of course be extended to any threshold.

---

![Figure 4.1: A diagram of a naive committee.](image)
4.2 Trainable Committees

For the trainable committees, the meta learner could be a weighted majority vote, where the weights are learned, but it could also be a more advanced method such as logistic regression or even another neural network. The structure of a trainable committee is illustrated in Figure 4.2.

![Diagram of an enhanced committee](image)

Figure 4.2: A diagram of an enhanced committee.

4.2.1 Weighted Majority Vote

If the base learners in the committee are of different accuracy, it is reasonable to give the more accurate base learners more voting power by assigning them higher weights. Equation (4.2) is therefore expanded upon to allow for weights

$$\hat{y} = \arg \max_c \sum_{i=1}^{K} \omega_i I(y_i(x) = c)$$

(4.4)

4.2.2 Conditionally Weighted Majority Vote

Furthermore, in the case that the accuracy rates of base learners vary for different classes, it is reasonable to condition the weights on the output of the base learners. Equation (4.2) is therefore expanded upon even further

$$\hat{y} = \arg \max_c \sum_{i=1}^{K} \omega_i(y_i(x)) I(y_i(x) = c)$$

(4.5)

where $\omega_i(y_i(x))$ denotes the conditional weights that depend on the signal $y_i(x)$. Again, similar to how Equation 4.2 was extended to Equation 4.3 to
include a threshold, the same procedure can be implemented for the conditionally weighted majority vote.
Chapter 5

Implementation

In this chapter a connection is drawn between the theory covered in the three previous chapters and the data available for this thesis. Performance measures for the classification algorithms and the setup for the trading simulation is introduced.

5.1 Raw Data

The data available for this thesis is the full limit order book of the EUR/USD exchange rate at tick-by-tick frequency for one week. The data consists of:

- Time stamp
- Bid prices: Each price for which there exists a market participant willing to buy EUR
- Bid volumes: The total volume at each bid price market participants are willing to buy EUR
- Ask prices: Each price for which there exists a market participant willing to sell EUR
- Ask volumes: The total volume at each bid price market participants are willing to sell EUR

Unlike daily or monthly data, tick-by-tick data is updated whenever prices or volumes change, which often occur at a millisecond rate, resulting in a time series with non-equidistant time points.

Complementing this order book is transaction data for the same period. The transaction data consists of
• Time stamp: When the transaction occurred
• Price: At what price the transaction occurred
•Paid or Given

*Given* indicating that the transaction occurred at the bid price and *Paid* indicating that the transaction occurred at the ask price.

### 5.2 Data Preprocessing

In liquid currency pairs such as EUR/USD the order depth, i.e. the number of levels of different bid and ask prices that are available in the market, is larger than in illiquid markets. In the data available, the order depth was at the highest 60, meaning that at that time there exist 60 different bid prices and 60 different ask prices for the EUR/USD. As the prices farthest away from the mid price are of less significance the decision is made to focus on the 15 best bid and ask prices for any given time. In Figure 5.1 the limit order book is shown, with the blue bars corresponding to the available volume for the 15 best ask prices, and the red bars corresponding to the available volume of at the 15 best bid prices. The gap between the highest bid price and the lowest ask price is called the spread.

![Figure 5.1: The order book illustrated.](image)

Since the data is tick-by-tick data it is bound to contain a certain degree of noise due to market microstructures such as bid-ask bounces and discrete price quotes [2]. To control for the market microstructure noise the data is instead subsampled at second intervals. While there are better ways to tackle this issue [2], for the purposes of predicting movements at \( \Delta t = 5, 20 \) and 60 second prediction horizons, subsampling is found to be enough. The subsampling is conducted by taking the last observed tick each second and results in an time-equidistant time series. Similarly, the transaction data is transformed to count the number of occurrences of each kind, given
and paid, each second. In Figure 5.2 the best bid and the best ask price is presented for a subset of the data.

Figure 5.2: EUR/USD exchange rate with the best bid and ask price over 1000 seconds, with $\Delta t = 1$ second.

5.2.1 Feature Selection

The data at hand allows for creation of various feature sets to be used as inputs for the machine learning methods. The feature sets are divided into three categories: volume based, price based and transaction based.

**Volume Based Features**

The volume based features describe the behavior over time of the volume at each level of bid and ask price. The following volume based feature sets are used to identify patterns that might potentially foreshadow a movement in price.

- $F_1$: The 5 second moving average volume distribution
- $F_2$: The 20 second moving average volume distribution
- $F_3$: The 100 second moving average volume distribution
- $F_4$: The 100 second moving average of the volume

**Price Based Features**

The price based features are designed to capture potential trends in the underlying currency pair.

- $F_5$: Moving average of 5, 20, 100 and 300 seconds bid and ask returns
• \( F_6 \): Volatility of the mid price returns over 5, 20, 100 and 300 seconds

Transaction Based Features

The transaction based features aim to identify activity in the market and can distinguish between activity on the bid side from activity on the ask side.

• \( F_7 \): Average number of given transactions and paid transactions per second over the last 5, 20, 100 and 300 seconds

5.2.2 Class Creation

The three classes \( \{ C_{+1}, C_0, C_{-1}\} \) are created by looking at the bid ask spread crossing and are defined by

1. \( C_{+1} \): \( P_{t+\Delta t}^{Bid} > P_t^{Ask} \)
2. \( C_0 \): \( P_{t+\Delta t}^{Bid} \leq P_t^{Ask} \) and \( P_{t+\Delta t}^{Ask} \geq P_t^{Bid} \)
3. \( C_{-1} \): \( P_{t+\Delta t}^{Ask} < P_t^{Bid} \)

where \( P_t^{Ask} \) and \( P_t^{Bid} \) denote the best ask and bid price at time \( t \). Similarly, \( P_{t+\Delta t}^{Ask} \) and \( P_{t+\Delta t}^{Bid} \) denote the best ask and best bid price, at the end of the prediction horizon at time \( t + \Delta t \). By choosing the classes in this way it is assured that acting on any prediction would in fact secure a definite profit, if correctly classified of course. The prediction horizon is set to \( \Delta t = 5, 20 \) and 60 seconds.

5.3 Experiment Setup

The total data available is divided such that there exist 6 iterations of training and testing, starting at \( t = 0 \), see Figure 5.3. For each iteration \( i \) the data is divided into training sets \( S_{train_1}^i, S_{train_2}^i \) and a test set \( S_{test}^i \), for \( i = 1, \ldots, 6 \). \( S_{train_1}^i \) is used to train the individual base classifiers, while \( S_{train_2}^i \) is used to train the committee. When a classifier is learning from a dataset that is unbalanced, it tends to have a bias towards the majority class [19, 3]. In this case, the \( C_0 \) class is, depending on the prediction horizon, sometimes several times larger than \( C_{+1} \) or \( C_{-1} \). This can significantly hamper the performance of identifying the minority classes, which implicitly, is this paper’s objective. In order to combat the imbalance in the train sets \( S_{train_1}^i \) and \( S_{train_2}^i \), it is subsampled without replacement to produce an even data distribution.
5.4. SUPPORT VECTOR MACHINES

Figure 5.3: Illustration of the data divided into a training set and testing set, moving 12 hours ahead for each iteration. 2 iterations are presented.

Hence, both $S_{train1}^i$ and $S_{train2}^i$ are created such that they contain an even distribution of classes from the last 48 hours of trading and such that $S_{train1}^i \cap S_{train2}^i = \emptyset$, $\forall i$. To restrict training times $S_{train1}^i$ contains no more than 60,000 samples and $S_{train2}^i$ contains no more than 20,000 samples. In cases for which there exist less than $(60000 + 20000)/3$ samples for any of the three classes and therefore the number of samples does not suffice for both training sets, the train sets $S_{train1}^i$ and $S_{train2}^i$ are created such that $S_{train1}^i$ contains 75% and $S_{train2}^i$ contains 25% of the available samples, respectively, again such that they contain an even distribution of classes. The performance of the base classifiers and the committees are then tested on the $S_{test}^i$ set. The training window is then shifted 12 hours, as illustrated in Figure 5.3, and the classifiers and the committees are retrained using the same procedure.

5.4 Support Vector Machines

In order to use SVMs in the created framework a multi-class SVM approach needs to be introduced as SVMs were originally developed for binary classification tasks. Furthermore the selection of input parameters and feature sets is described in this section.

5.4.1 Multi-class SVM

Support Vector Machines were originally developed for binary classification, i.e. for problems with two classes, +1 and −1. In the case at hand however multiple classes exist and thus multi-class SVMs have to be investigated. In [5] different approaches for multi-class SVMs are discussed.

The easiest and most straight-forward of those approaches is the one-versus-the-rest approach. This requires $K$ SVMs to be trained when $K$ classes exist and labelling the respective $k$th class as +1 while labelling all other $K − 1$
classes as $-1$ for every $k$. By the final classification for each data sample $x_i$ and for all $i = 1, \ldots, N$, is then made in the manner shown in Section 2.3

$$\text{class of } x_i \equiv \arg \max_{k=1,\ldots,K}((w^k)^T \phi(x) + b^k).$$

In words this means that the class is chosen for which the return value of the function (2.30) is the farthest away from the separating hyperplane. The downside of this approach is that the previously developed balanced class distribution is disrupted and an unbalanced class distribution is created which results in the learned classifier having a bias towards the majority class.

Another multi-class SVM approach is the so called one-versus-one approach where $K(K-1)/2$ SVMs must be trained where the $k$th class is labeled $+1$ and the other class $m$ is labeled $-1$ where $m \in \{1, \ldots, k-1, k+1, \ldots, K\}$ for all $k$ and $m$. Then the final labels for each class $k$ are decided by a majority vote of all tested SVMs, i.e. if the $m$th SVM where class $i$ versus $j$ was trained suggests that the $n$th data point is in class $i$ the count for class $i$ for sample $n$ will be increased by one. In cases where each class gets the same number of votes for a sample $x$ the class that indicates no changes is chosen, such as the class for which $P_{t+\Delta t}^{Bid} \leq P_t^{Ask}$ and $P_t^{Ask} \geq P_{t+\Delta t}^{Bid}$ holds. This requires more computational power than the one-versus-the-rest approach for $K > 3$.

There also exist approaches on multi-class SVMs that solve for all classes in one optimization step. Those are computationally heavy however and are less appropriate for very large data sets.

Due to its advantages of reduced training set size and due to the good results in the one-versus-one multi-class approach is used for further experiments.

### 5.4.2 SVM Calibration

In this thesis the focus is on Support Vector Machines using the linear and radial basis function to construct the kernel, as defined in Section 2.3. Those kernel functions have proven superior in training time and performance to the polynomial kernel and also have performed well in other studies that researched financial time series forecasting with SVMs. The performance of both kernels is tested with the penalization parameter $C \in \{1, 10, 400, 800, 3000, 6000\}$ for each of the feature sets individually as well as combinations of feature sets. In case of the radial basis function, experiments were conducted for the parameter values $\gamma \in \{1, 10, 1000\}$. Recall from Section 2.3 that a small $\sigma$ risks overfitting while a large $\sigma$ prevents from
training complex classifiers where $\gamma = 1/(2\sigma^2)$. Hence, the values tested for $\gamma$ are selected to be equivalent to a medium to small sized $\sigma$.

To evaluate the performance of each experiment the training set $S_{train2}$ of the first iteration, i.e. $S_{traini}^2$, introduced in Section 5.3 is used. These initial experiments suggest the use of a small C for combinations of feature sets and the use of a larger C for individual feature sets. Recall that a large C penalizes misclassifications on the training set more but is simultaneously subject to overfitting. Furthermore the parameter value $\gamma = 1$ has proven superior in training time and performance to the larger values for most of the experiments and is selected for all feature sets. The decision making process of which feature sets to use for the committee members is a mix of experiment results and practical understanding. It is essential when working with SVMs to combine features that harmonize and discard irrelevant features such as noise, outliers and redundant features [20]. Feature sets decided upon after the initial experiments include $F_1, F_2, F_3, F_1 \cup F_2 \cup F_3, F_5 \cup F_6 \cup F_7$ and other combinations. Particularly the feature sets $F_5, F_6, F_7$ have proven useful in combination with other feature sets.

5.5 Artificial Neural Networks

5.5.1 Normalization

Before inputting the data into the models, it is normalized. In the case of the ANNs it is necessary to normalize the data in order to prevent larger numbers to override smaller ones. There are multiple ways of scaling inputs and while there is no recommended standard procedure, it is however recommended that the data is normalized between slightly offset values, to avoid saturation. For the neural networks, the inputs $x$ are therefore normalized according to the min-max normalization defined as

$$\tilde{x} = \frac{x - \max_x}{\max_x - \min_x} (\max_{\tilde{x}} - \min_{\tilde{x}}) + \min_{\tilde{x}},$$

where the new values $\max_{\tilde{x}}$ and $\min_{\tilde{x}}$ are set to 0.9 and 0.1. [3]

5.5.2 ANN Implementation

The ANN used in this thesis is a two-layer neural network trained with the standard backpropagation algorithm and stochastic gradient descent. An online method is used, rather than a batch method, due to its more efficient handling of redundancy and capability of escaping local minima [5]. The number of input neurons in the input layer corresponds to the
number of features in the feature set and the number of output neurons in
the output layer corresponds to the number of classes in the target set. The
hidden neurons implement the hyperbolic tangent activation function as they
combat saturation and the output neurons implement the logistic activation
function. During training, the number of epochs are set to 200. The cost
function used is the cross-entropy cost function. The hyperparameters that
are left to be decided on in the ANN model is the learning rate $\eta$ and the
number of hidden neurons $H$.

After a general training session, the decision is made to use two sets of
neural networks with 20 and 50 hidden neurons each, on the feature set
$F_2 \cup F_3 \cup F_6 \cup F_7$. The neural networks are then initialized four times
each, resulting in 8 neural network classifiers. The learning rate $\eta$ is set to
0.01.

5.6 Committees

The committees are divided into two classes, trainable and non trainable,
as discussed in Chapter 4. Henceforth, these committees are referred to as
naive and enhanced, respectively.

5.6.1 Naive Committee

Two versions of the naive committee are created, one with a simple majority
rule following the definitions in Section 4.1.2, where at least 50% of all votes
are necessary for a prediction to be assigned, and one with a majority rule
for which 80% of all votes are necessary for a prediction to be assigned.
Majority rule is applied, rather than plurality rule, as it yields more confident
predictions. Refer to Table 5.1 for the names of the committees. If the
majority of 50% or 80% for the two committees respectively is not reached
the data sample is labeled as not classifiable.

5.6.2 Enhanced Committee

In order for a classifier to be effective, it is of vital importance that the indi-
vidual classifiers not only perform well, but that the members have diversity
in their predictions [16]. Correlation statistics introduced in Chapter 4 are
therefore used to eliminate candidates that exhibit high linear correlation
with other classifiers. Such candidates are considered to bring little new in-
formation to the ensemble. Candidates with high correlation ($|\rho| \geq 0.9$) are
therefore discarded. Additionally, when using various different classification
of algorithms there is some variance in performance to each committee member. The phenomenon is observed that the performance of the members is not necessarily consistent and while some methods work well in a specific market environment, others potentially work better at another time. To account for poor performance of the members, those that perform worse than or equally well as a random classifier on the training set $S_{\text{train}}$ are discarded. Here a theoretical random classifier is considered that has an accuracy for a specific class equal to the proportion of that class in the entire data set considered, in this case $S_{\text{train}}$. Based on these two actions, two committees $EM_{50}$ and $EM_{80}$ are created, see Table 5.1.

To further improve upon the two enhanced committees, two more committees $CWSM$ and $CWM_{80}$ are created. While they also discard correlated members, they weight the uncorrelated members using the formula introduced in Section 4.2.2. This assures that members that perform well on the training set $S_{\text{train}}$ are assigned higher weights than members that perform poorly on the set $S_{\text{train}}$. Here the conditionally weighted majority vote rule is chosen rather than the weighted majority vote rule introduced in Section 4.2.1. This is due to prioritizing the significance of high accuracies for the directional predictions, i.e. the classes $C_{+1}$ and $C_{-1}$, rather than overall accuracies of the classifier. Overall accuracies tend to be distorted when there is a dominating class which in this case is $C_{0}$. The weights $\omega_{ki}$ are calculated in the following manner

$$\omega_{ki} = \frac{\text{Precision}_{C_{i}}^{k}}{\sum_{j=1}^{K} \text{Precision}_{C_{j}}^{k}},$$

(5.1)

for all $k = 1, \ldots, K$, where $K$ is the total number of members in the committee, $C_{i}$ with $i \in \{+1, 0, -1\}$, and $\text{Precision}_{C_{i}}$ as defined in Section 5.7.

Similarly as for the naive committees, if a sample does not reach the majority vote it is referred to as not classifiable.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{50}$</td>
<td>Naive committee with simple majority rule (50%)</td>
</tr>
<tr>
<td>$M_{80}$</td>
<td>Naive committee with 80% majority rule</td>
</tr>
<tr>
<td>$EM_{50}$</td>
<td>Enhanced committee with simple majority rule (50%)</td>
</tr>
<tr>
<td>$EM_{80}$</td>
<td>Enhanced committee with 80% majority rule</td>
</tr>
<tr>
<td>$CWM_{50}$</td>
<td>Conditionally weighted committee with 50% majority rule</td>
</tr>
<tr>
<td>$CWM_{80}$</td>
<td>Conditionally weighted committee with 80% majority rule</td>
</tr>
</tbody>
</table>

Table 5.1: Names and descriptions of committees.
5.7 Performance Measurement of Classifiers

There are various measures that can be used in order to evaluate the performance of classifiers. While many papers use the overall accuracy as a measure, it can be deceptive. Accuracy does not necessarily express predictive power, especially in cases where there is class imbalance in the test set. Therefore the accuracy measure is used with caution and more focus is given to other measures.

Starting with the confusion matrix illustrated in Table 5.2 for a two class classification system \( \{ C_{+1}, C_{-1} \} \), True Positive defines the number of samples of class \( C_{+1} \) correctly classified as such. True Negative is the number of samples belonging to class \( C_{-1} \) correctly classified as such. False Positive is the number of samples belonging to class \( C_{-1} \) that are incorrectly assigned to class \( C_{+1} \) and False Negative is the number of examples belonging to class \( C_{+1} \) that are incorrectly assigned to class \( C_{-1} \).

<table>
<thead>
<tr>
<th>Class</th>
<th>Classified as ( C_{+1} )</th>
<th>Classified as ( C_{-1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{+1} )</td>
<td>True Positive (TP)</td>
<td>False Negative (FN)</td>
</tr>
<tr>
<td>( C_{-1} )</td>
<td>False Positive (FP)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

Table 5.2: Confusion Matrix

This system is expanded for a multi class classification setting, where \( TP_{C_i} \) is defined as the True Positive for class \( C_i \) with \( i \in \{+1, 0, -1\} \), and similarly for False Positive \( (FP_{C_i}) \) and False Negative \( (FN_{C_i}) \).

From here, the following measures can be calculated

\[
\text{Precision}_{C_i} = \frac{TP_{C_i}}{TP_{C_i} + FP_{C_i}}
\]

\[
\text{Recall}_{C_i} = \frac{TP_{C_i}}{TP_{C_i} + FN_{C_i}}
\]

\[
\text{Precision}_{C_i} \text{ measures what fraction of all data classified as } C_i \text{ was labeled correctly. Recall, on the other hand measures what fraction of all data belonging to class } C_i \text{ does the classifier register. It is easy to see that one in many cases can be achieved at the expense of the other. It is therefore necessary to also measure the balance between precision and recall in order to measure the predictive power of a classifier. For this purpose the F1-Score is used, which is the harmonic mean of precision and recall.}
\]

\[
\text{F1-Score}_{C_i} = \frac{2 \text{Precision}_{C_i} \text{Recall}_{C_i}}{\text{Precision}_{C_i} + \text{Recall}_{C_i}}
\]
In order to create an overall measure for the classifier, and not just for the individual classes, the measures are combined by either micro- or macro-averaging \[27\]. In this thesis only macro-averaging is used, and defined as

\[
\text{Precision}_M = \frac{\sum_{i=1}^{\lvert C \rvert} TP_{C_i}}{\sum_{i=1}^{\lvert C \rvert} TP_{C_i} + FP_{C_i}},
\]

\[
\text{Recall}_M = \frac{\sum_{i=1}^{\lvert C \rvert} TP_{C_i}}{\sum_{i=1}^{\lvert C \rvert} TP_{C_i} + FN_{C_i}},
\]

\[
\text{F1-Score}_M = \frac{2 \times \text{Precision}_M \times \text{Recall}_M}{\text{Precision}_M + \text{Recall}_M},
\]

where \(\lvert C \rvert\) denotes the total number of classes.

5.8 Trading Simulation

To test the results obtained by the committees in a practical setting a simple trading strategy is implemented and evaluated with respect to the trading strategy accuracy and the performance of the strategy, as introduced in Sections 5.8.2 and 5.8.3 respectively. The results will also be compared to the performances of the benchmarks introduced in Section 5.8.4.

5.8.1 Trading Strategy

The trading in foreign exchange is in most aspects similar to stock trading. One key difference is short selling. To short sell stocks, the stock is borrowed and then sold in the market with hopes of being able to buy it back at a later time at a lower price. In exchange rates, the dynamic is that when you are long one currency in a currency pair, you are simultaneously short the other currency in the currency pair, and vice versa. This implies that the market participant cannot be neutral to the currency pair and trades a view at any time. The trading strategy, and how it relates to the signals from the classifiers is illustrated in Figure 5.4.

The strategy consists of going long EUR if the classifier predicts \textit{buy}, going long USD if the classifier predicts \textit{sell}, or holds the current position if the classifier predicts \textit{hold}. Here the signal \textit{buy} is equivalent to the prediction being in the class \(C_{+1}\), \textit{sell} is equivalent to \(C_{-1}\), and the signal \textit{hold}
The strategy has two states, long EUR and long USD, and starts in the state long USD. The edges represent the trading signals obtained from the committee.

is equivalent to the class $C_0$. For each transaction all capital available is used.

The strategy presented in Figure 5.4 trades according to the signal obtained from the underlying committee every second. The trading strategy does not only consider the signal of the respective committee at $t$ for all $t$, but also all signals in the $p = \left\lfloor \frac{\Delta t}{2} \right\rfloor$ seconds before, i.e. all signals from the interval $[t - p, t]$. Based on all signals in this interval one signal $\hat{y}$ at $t$ is created, that is the signal that occurred at least $\frac{2}{3} \cdot p$ of the times. If this threshold is not reached the signal hold is assigned. Mathematically expressed this becomes

$$\hat{y} = \left[ \arg \max_c \sum_{i=1}^{p} 1(y_i = c) \right] 1 \left( \sum_{i=1}^{p} 1(y_i = c^*) > \frac{2}{3} \cdot p \right), \quad (5.2)$$

where $c \in \{C_{+1}, C_0, C_{-1}\}$ denotes the classes, $1(\cdot)$ denotes the indicator function, and $c^*$ denotes the most occurred class that solves the maximization in the left hand side of (5.2). From (5.2) it can be inferred that the trading strategy applies a majority vote rule of the past $p$ signals with a threshold of 66%. The idea behind this is to act on fewer but more certain trades. The selection of the threshold of 66% stems from the process of trying to find a generalizable number for all prediction horizons. In particular the trading strategies for the 5 second prediction horizon use the $p = \left\lfloor \frac{5}{2} \right\rfloor = 2$ past signals to decide upon a view. To assure that not just 1 signal suffices to make a certain prediction a threshold $> 50\%$ must be considered.

There are several differences between the backtesting of the trading strategies developed in this thesis and the application of those strategies in a live trading scenario. A number of assumptions and simplifications are made in the simulation. It is assumed that there is no spread and no transaction costs. The size of the transaction is always available at the current price level and the transactions made have no impact on the market.
5.8. TRADING SIMULATION

5.8.2 Trading Strategy Accuracy

To measure the accuracy of the trading strategy the concept of Trading Strategy Accuracy (TSA) following [1] is applied. The idea is to compare the number of hits to the number of misses. For this thesis, a hit in the TSA framework is defined as a signal that was correctly classified for the classes $C_{+1}$ and $C_{-1}$, while a miss is defined as a signal that was predicted to belong to class $C_{+1}$ but was in fact element of class $C_{-1}$ or a signal that was classified as $C_{-1}$ but was in fact in class $C_{+1}$. The idea is to create a plot that represents the distribution of hits compared to misses over the course of all test periods. To create this plot the number of total hits $H$ and the number of total misses $M$ need to be computed. Furthermore $K = \max(H, M)$ is defined and the cumulative hit and miss rates can be determined

$$H_i = \begin{cases} H_{i-1} + 1/K, & \text{if } i \text{th trade is a hit} \\ H_{i-1}, & \text{otherwise} \end{cases}$$

$$M_i = \begin{cases} M_{i-1} + 1/K, & \text{if } i \text{th trade is a miss} \\ M_{i-1}, & \text{otherwise} \end{cases}$$

The area under the curve (AUC) plotted with respect to $M_i$ and $H_i$ for all $i$ yields the total accuracy of the forecast. The random predictor is simply a line from the left bottom corner to the right top corner of a square and results in an accuracy of 50%, i.e. has an AUC of 0.5. Every strategy with an AUC value greater than 0.5 is considered better than random.

5.8.3 Trading Performance Measure

To evaluate the overall performance of the trading strategies for each committee the annualized return of the strategy is considered. The annualized return $R_{ann}$ is calculated by

$$R_{ann} = \left( \frac{\sum_{s=1}^{T} R_s}{T} + 1 \right)^{\frac{252}{p}} - 1$$

where $p$ denotes the fraction of one second relative to a day, i.e. $p = 1/(60 \times 60 \times 24)$, 252 is the number of trading days per year, and $R_s$ denotes the secondly returns, calculated as $R_s = \frac{V_s}{V_{s-1}} - 1$, for $s = 2, \ldots, T$. $T$ denotes the length of the test period in seconds and $V_i$ denotes the value of the portfolio at time $i$.

5.8.4 Benchmarks

Two naive trading strategies are implemented as a benchmark. The first strategy is a simple buy and hold strategy that enters a long position at
$t = 1$, i.e. buys EUR, and holds this position until $t = T$. It is denoted by 
*Buy and Hold*. The second strategy, referred to as *Previous Signal*, uses the 
previous directional movement as its prediction for the next horizon $\Delta t$. It 
is defined as

$$\hat{y}_t^B = y_{t-\Delta t},$$

where $y_{t-\Delta t}$ denotes the realized directional movement in the time interval 
$[t - \Delta t, t]$ and where $y_{t-\Delta t} \in \{C_+, C_0, C_-\}$. 
# Chapter 6

## Results

Below, the results of the classification performance and the trading performance are presented.

### 6.1 Classification Performance

In Table 6.1, the class distribution of the dataset is presented for reference purposes. While all committees described in Table 5.1 are evaluated, the focus is on the enhanced committees \( EM50, EM80, CWM50, \) and \( CWM80 \) due to better and more consistent performance. The performance measurements of the committees are summarized in Table 6.2, 6.3, and 6.4, for the 5, 20 and 60 second prediction horizons, respectively. Precision (PPV) and Recall (TPR) are presented for \( C_{+1} \) and \( C_{-1} \) together with the Not Classifiable rate, which denotes the fraction of the data samples on which the classifier did not manage to achieve the desired majority threshold. Note that the values for best SVM and best ANN refer to the highest values achieved by individual members of the respective method, and do not necessarily belong to the same classifier. A more extensive presentation of the results can be found in Appendix A.

<table>
<thead>
<tr>
<th>Class</th>
<th>5 seconds</th>
<th>20 seconds</th>
<th>60 seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{+1} )</td>
<td>0.111</td>
<td>0.225</td>
<td>0.311</td>
</tr>
<tr>
<td>( C_{0} )</td>
<td>0.779</td>
<td>0.550</td>
<td>0.373</td>
</tr>
<tr>
<td>( C_{-1} )</td>
<td>0.110</td>
<td>0.225</td>
<td>0.316</td>
</tr>
<tr>
<td>Total</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Table 6.1: Class distribution for different prediction horizons.
### CHAPTER 6. RESULTS

<table>
<thead>
<tr>
<th>Type</th>
<th>PPV(_{+1})</th>
<th>PPV(_{-1})</th>
<th>TPR(_{+1})</th>
<th>TPR(_{-1})</th>
<th>Not Class.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best SVM</td>
<td>0.164</td>
<td>0.163</td>
<td>0.457</td>
<td>0.628</td>
<td>-</td>
</tr>
<tr>
<td>Best ANN</td>
<td>0.175</td>
<td>0.197</td>
<td>0.396</td>
<td>0.466</td>
<td>-</td>
</tr>
<tr>
<td>SVM EM50</td>
<td>0.180</td>
<td>0.180</td>
<td><strong>0.387</strong></td>
<td>0.501</td>
<td>0.289</td>
</tr>
<tr>
<td>SVM EM80</td>
<td>0.197</td>
<td>0.204</td>
<td>0.321</td>
<td><strong>0.522</strong></td>
<td>0.696</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>0.164</td>
<td>0.194</td>
<td>0.312</td>
<td>0.362</td>
<td>0.109</td>
</tr>
<tr>
<td>ANN EM80</td>
<td>0.162</td>
<td>0.204</td>
<td>0.207</td>
<td>0.368</td>
<td>0.383</td>
</tr>
<tr>
<td>Full EM50</td>
<td>0.185</td>
<td>0.199</td>
<td>0.326</td>
<td>0.434</td>
<td>0.199</td>
</tr>
<tr>
<td>Full EM80</td>
<td><strong>0.208</strong></td>
<td><strong>0.213</strong></td>
<td>0.247</td>
<td>0.375</td>
<td>0.645</td>
</tr>
<tr>
<td>Full CWM50</td>
<td>0.183</td>
<td>0.195</td>
<td>0.348</td>
<td>0.410</td>
<td>0.119</td>
</tr>
<tr>
<td>Full CWM80</td>
<td><strong>0.208</strong></td>
<td><strong>0.213</strong></td>
<td>0.251</td>
<td>0.468</td>
<td>0.647</td>
</tr>
</tbody>
</table>

Table 6.2: Performance of all committees for \(\Delta t = 5\).

<table>
<thead>
<tr>
<th>Type</th>
<th>PPV(_{+1})</th>
<th>PPV(_{-1})</th>
<th>TPR(_{+1})</th>
<th>TPR(_{-1})</th>
<th>Not Class.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best SVM</td>
<td>0.283</td>
<td>1.000</td>
<td>0.388</td>
<td>0.624</td>
<td>-</td>
</tr>
<tr>
<td>Best ANN</td>
<td>0.284</td>
<td>0.307</td>
<td>0.239</td>
<td>0.501</td>
<td>-</td>
</tr>
<tr>
<td>SVM EM50</td>
<td>0.301</td>
<td>0.305</td>
<td><strong>0.255</strong></td>
<td>0.527</td>
<td>0.184</td>
</tr>
<tr>
<td>SVM EM80</td>
<td><strong>0.337</strong></td>
<td><strong>0.346</strong></td>
<td>0.099</td>
<td>0.385</td>
<td>0.807</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>0.278</td>
<td>0.312</td>
<td>0.212</td>
<td>0.444</td>
<td>0.070</td>
</tr>
<tr>
<td>ANN EM80</td>
<td>0.288</td>
<td>0.322</td>
<td>0.161</td>
<td>0.485</td>
<td>0.290</td>
</tr>
<tr>
<td>Full EM50</td>
<td>0.304</td>
<td>0.316</td>
<td>0.172</td>
<td>0.502</td>
<td>0.145</td>
</tr>
<tr>
<td>Full EM80</td>
<td>0.300</td>
<td>0.339</td>
<td>0.048</td>
<td><strong>0.551</strong></td>
<td>0.698</td>
</tr>
<tr>
<td>Full CWM50</td>
<td>0.304</td>
<td>0.317</td>
<td>0.170</td>
<td>0.492</td>
<td>0.135</td>
</tr>
<tr>
<td>Full CWM80</td>
<td>0.300</td>
<td>0.339</td>
<td>0.047</td>
<td>0.550</td>
<td>0.698</td>
</tr>
</tbody>
</table>

Table 6.3: Performance of all committees for \(\Delta t = 20\).

<table>
<thead>
<tr>
<th>Type</th>
<th>PPV(_{+1})</th>
<th>PPV(_{-1})</th>
<th>TPR(_{+1})</th>
<th>TPR(_{-1})</th>
<th>Not Class.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best SVM</td>
<td>0.367</td>
<td>0.377</td>
<td>0.487</td>
<td>0.570</td>
<td>-</td>
</tr>
<tr>
<td>Best ANN</td>
<td>0.366</td>
<td>0.372</td>
<td>0.392</td>
<td>0.464</td>
<td>-</td>
</tr>
<tr>
<td>SVM EM50</td>
<td>0.373</td>
<td>0.380</td>
<td><strong>0.346</strong></td>
<td>0.475</td>
<td>0.151</td>
</tr>
<tr>
<td>SVM EM80</td>
<td><strong>0.400</strong></td>
<td><strong>0.398</strong></td>
<td>0.290</td>
<td>0.445</td>
<td>0.743</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>0.350</td>
<td>0.376</td>
<td>0.184</td>
<td>0.462</td>
<td>0.124</td>
</tr>
<tr>
<td>ANN EM80</td>
<td>0.343</td>
<td>0.391</td>
<td>0.125</td>
<td><strong>0.478</strong></td>
<td>0.457</td>
</tr>
<tr>
<td>Full EM50</td>
<td>0.384</td>
<td>0.382</td>
<td>0.230</td>
<td>0.474</td>
<td>0.173</td>
</tr>
<tr>
<td>Full EM80</td>
<td>0.387</td>
<td><strong>0.398</strong></td>
<td>0.071</td>
<td>0.438</td>
<td>0.795</td>
</tr>
<tr>
<td>Full CWM50</td>
<td>0.381</td>
<td>0.379</td>
<td>0.218</td>
<td>0.460</td>
<td>0.172</td>
</tr>
<tr>
<td>Full CWM80</td>
<td>0.394</td>
<td>0.389</td>
<td>0.060</td>
<td>0.460</td>
<td>0.767</td>
</tr>
</tbody>
</table>

Table 6.4: Performance of all committees for \(\Delta t = 60\).
6.2 Trading Performance

In this Section the results of a trading simulation, implemented in the manner specified in Section 5.8, are presented. Two trading strategies based on the signals created by the full committees EM50 and EM80 are implemented. Since the committees CWM50 and CWM80 performed similarly to the committees EM50 and EM80 (see Tables 6.2, 6.3 and 6.4) but require additional training, the latter are chosen to maintain simplicity. Both trading strategies are compared to a simple *Buy and Hold* strategy as well as the *Previous Signal* benchmark, defined in Section 5.8.4. All four trading strategies are evaluated with respect to the concept of TSA introduced in Section 5.8.2 and performance measures defined in Section 5.8.3. In Figures 6.1, 6.3, and 6.5 the TSAs for \( \Delta t = 5, 20, 60 \) are presented. Graphs of the trading performances for all prediction horizons are presented in Figures 6.2, 6.4, and 6.6 while the annualized returns as well as the total number of transactions and the areas under the curves (AUC) of the TSAs are presented in Tables 6.5, 6.6, and 6.7 for the prediction horizons \( \Delta t = 5, 20, 60 \) respectively. Here the number of transactions includes all cases where the position changed, see Figure 5.4

Additionally to the trading results presented in this section, the results of the trading simulation when method specific committees are used, i.e. SVM and ANN committees, are presented in Appendix B.
CHAPTER 6. RESULTS

Figure 6.1: TSA for 5 second prediction horizon.

Figure 6.2: Trading Performances for 5 second prediction horizon.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Return</th>
<th>Transactions</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM50</td>
<td>0.200</td>
<td>3183</td>
<td>0.597</td>
</tr>
<tr>
<td>EM80</td>
<td>−0.316</td>
<td>444</td>
<td>0.651</td>
</tr>
<tr>
<td>Previous Signal</td>
<td>−0.824</td>
<td>9604</td>
<td>0.786</td>
</tr>
<tr>
<td>Buy and Hold</td>
<td>−0.507</td>
<td>2</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 6.5: Annualized returns, total number of transactions, and area under the curve (AUC) for the 5 second prediction horizon.
6.2. TRADING PERFORMANCE

Figure 6.3: TSA for 20 second prediction horizon.

Figure 6.4: Trading Performances for 20 second prediction horizon.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Return</th>
<th>Transactions</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM50</td>
<td>0.216</td>
<td>676</td>
<td>0.552</td>
</tr>
<tr>
<td>EM80</td>
<td>−0.089</td>
<td>30</td>
<td>0.420</td>
</tr>
<tr>
<td>Previous Signal</td>
<td>−0.609</td>
<td>6116</td>
<td>0.851</td>
</tr>
<tr>
<td>Buy and Hold</td>
<td>−0.507</td>
<td>2</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 6.6: Annualized returns, total number of transactions, and area under the curve (AUC) for the 20 second prediction horizon.
CHAPTER 6. RESULTS

Figure 6.5: TSA for 60 second prediction horizon.

Figure 6.6: Trading Performances for 60 second prediction horizon.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Return</th>
<th>Transactions</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM50</td>
<td>0.530</td>
<td>460</td>
<td>0.501</td>
</tr>
<tr>
<td>EM80</td>
<td>-0.401</td>
<td>10</td>
<td>0.625</td>
</tr>
<tr>
<td>Previous Signal</td>
<td>-0.758</td>
<td>3823</td>
<td>0.771</td>
</tr>
<tr>
<td>Buy and Hold</td>
<td>-0.507</td>
<td>2</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 6.7: Annualized returns, total number of transactions, and area under the curve (AUC) for the 60 second prediction horizon.
Chapter 7

Discussion

This chapter covers the classification performance, trading performance as well as possible improvements and future work.

7.1 Classification Performance

By comparing the results of the committees from Tables 6.2, 6.3 and 6.4 to the distribution of the classes for 5, 20 and 60 seconds prediction horizons (shown in Table 6.1), it is evident that all committees outperform random chance. The committee with the highest precision for class $C_{+1}$ outperforms random chance by 87% for the 5 second prediction horizon. The same numbers are 50% and 29% for the 20 and 60 seconds prediction horizons, respectively. Similarly, for class $C_{-1}$, the best performing classifier outperforms random chance by 94%, 54% and 28% for each prediction horizon, respectively. The precision in absolute numbers is however higher for longer prediction horizons, the best performing committee achieving 21%, 34% and 40% precision for $C_{+1}$, for 5, 20 and 60 seconds, respectively. The same values for $C_{-1}$ are 21%, 35% and 40%.

For SVMs, the committees $EM50$ and $EM80$ yielded better precision than even the best performing base classifier for all prediction horizons. The only exception is the base classifier yielding a 100% precision for class $C_{-1}$ at the prediction horizon of 20 seconds. This unique case predicted only three downward movements, all of which were correct. Excluding this member the highest precision reached by an individual SVM is 29%. In general the SVM classifiers have lower precision and higher recall than the ANN classifiers.

The ANN benefited from committee creation to a lesser extent. In general, the ANN committees $EM50$ and $EM80$ perform worse than the single best
performing base classifier with respect to class $C_{+1}$, while the opposite holds true for class $C_{-1}$. Considering the relatively high precision of the neural networks, the poor committee performance is thought to stem from the low diversity among the ANN base learners. A correlation analysis yielded a significantly higher positive correlation for the ANN base learners than the SVM base learners. This may have been caused by the selection of the feature set which, unlike for the SVMs, is the same for all members. This would explain why the SVM only committees outperform the ANN only committees despite having relatively poorer individual members. The high correlation would also explain the significantly lower Not Classifiable rates for the ANN committees. These findings confirm what has been shown in previous studies as well as in ensemble learning theory that committee creation improves performance if the committee members perform well individually and have sufficiently high diversity.

For the SVM committees, the precision for $C_{+1}$ and $C_{-1}$ show consistent improvement moving from EM50 to EM80 for all prediction horizons. Again, an indication of the well diversified base learners. However this comes with a trade off of more Not Classifiable samples as well as lower recall values. A higher majority threshold can yield a higher number of false negatives, which by definition has a negative impact on recall. The ANN committees with EM50 and EM80 yield overall similar results and the Not Classifiable rate is moderate compared to the significantly larger drop in classifiable data that the SVM committees experience.

In the case of full EM50 committees, where ANNs and SVMs are combined, results show an increase in precision versus their corresponding method specific committees, with a relative to the SVM EM50, low Not Classifiable rate. For the full EM80 committee, the precision of $C_{-1}$ increases substantially compared to the full EM50 committee. This only holds true at the 5 second prediction horizon for the $C_{+1}$ precision. It appears that the high correlation of the ANN members limits the potential improvement of an increased threshold.

Regarding the Conditionally Weighted Majority committees, the results suggest that additional training of the committees on the training set $S_{train2}$ does not consistently improve performance. The full committees CWM50 and CWM80 yield accuracies very similar to the full committees EM50 and EM80.

A general observation, discussed in more detail in the following section, is the increased rate of Not Classifiable data for the EM80 committees vis-a-vis the EM50 committees. In a the trading simulation non-classifiable data is translated to a Hold signal, meaning that a high number of non-classifiable data would possibly incur high opportunity costs.
7.2 TRADING PERFORMANCE

In the context of previous studies, the performance of the classifiers in this thesis are significant. The study most comparable to this thesis is [9]. Conducted on the EUR/USD exchange rate, using the same class definitions, the study was unable to predict positive or negative movements at a 5 second prediction window and had a precision of 12% and 16% for \( C_{+1} \) and \( C_{-1} \), respectively, for the 20 second prediction horizon. This, compared to 20.8% and 21.3% as well as 33.7% and 34.6% achieved for the best committee in this study for 5 seconds and 20 seconds prediction horizons, respectively. At \( \Delta t = 60 \), the best performing committee achieved 40.0% and 39.8% precision for \( C_{+1} \) and \( C_{-1} \), respectively. This could be compared to [9]’s 50 second prediction performance of 30% and 16%. Furthermore the findings in this thesis confirms the study [14] in which the author showed that the creation of a committee consisting of multiple different methods yields improved and more consistent accuracies than the individual members.

On a side note, experiments show that \( C_{+1} \) and \( C_{-1} \) are more often confused with each other than with \( C_0 \). This result is perhaps counterintuitive, as the classification imposed on the data, dividing class \( C_{+1} \) from \( C_0 \) and \( C_0 \) from \( C_{-1} \) is more or less arbitrary. However, by reclassifying \( C_{+1}, C_0 \) and \( C_{-1} \) as

1. \( C_{\text{movement}} \equiv C_{+1} \cup C_{-1} \)
2. \( C_{\text{no-movement}} \equiv C_0 \)

the precision and recall is significantly improved, in terms of absolute numbers. For instance, the full \( EM_{50} \) Committee on 60 second prediction horizon is able to predict market movement with 75.1% precision and 69.3% recall. This type of prediction has its merits, assisting market making by widening and narrowing the spread on instruction from the classifier signals. However, measuring the viability of such a strategy is beyond the scope of this study.

7.2 Trading Performance

Despite the full \( EM_{80} \) committee yielding higher precisions than the full \( EM_{50} \) committee, the trading strategy created using the full \( EM_{50} \) outperforms the full \( EM_{80} \). Both, in turn, outperform the Buy and Hold and Previous Signal benchmarks across all prediction horizons in terms of annualized returns, as seen in Tables 6.5, 6.6 and 6.7. Even though the exchange rate drops over the three trading days resulting in a negative return for the Buy and Hold benchmark, the \( EM_{50} \) trading strategy yields a positive annualized return of 20.0%, 21.6% and 53.0% for 5, 20 and 60 second prediction horizon, respectively.
The case of \textit{EM50} outperforming \textit{EM80} in the trading simulation stems from the nature of the currency markets, where at any given point it is necessary to hold a view of the future. As seen in Section 5.8 and more specifically in Figure 5.4 there are only two states in the trading strategy. Either long EUR or long USD. While raising the majority threshold to 80\% increases the precision, the recall worsens and the number of non classifiable data increases. Since \textit{Not Classifiable} is translated into a \textit{hold} signal, the trading strategy remains in the previous position at times where a majority of the committee members votes against doing so, but does not change the position simply because the 80\% are not reached. Hence, a \textit{EM50} committee is better suited to be implemented in a trading strategy. It should be noted that the lower number of \textit{Not Classifiable} data results in the \textit{EM50} performing significantly more transactions than the \textit{EM80}, which would be a disadvantage when considering potential transaction costs.

In contrast to the performance of the trading strategies relative to the \textit{Previous Signal} benchmark, the TSA and AUC indicate higher hit rates for the benchmark for all prediction horizons, see Figures 6.1, 6.3 and 6.5, and the AUC values in Tables 6.5, 6.6 and 6.7. Although this is counterintuitive it seems that while the absolute number of hits is higher for the \textit{Previous Signal} benchmark, it fails to detect the directional changes as it always uses the previous realized movement as future prediction. The performance of the benchmark is strongest for the 20 second prediction horizon which suggests a positive trend following behavior for this prediction horizon, while 5 and 60 seconds prediction horizons might have an inverse relationship. This is in line with the results in [23], where it is shown that for timespans of up to 5 seconds and over 60 seconds an inverse trend following relationship predominates and that for timespans in between two consecutive movements in the same direction occur with higher probability than that of movements in the opposite direction.

Both strategies, \textit{EM50} and \textit{EM80}, have their highest AUC for the 5 second prediction horizons, see Table 6.5. Furthermore the higher number of trades made in case of the shorter prediction horizons suggest the majority being reached more often and hence stronger predictive power, particularly for \textit{EM80} strategy. A discrepancy that becomes obvious is the lower annualized returns despite higher AUC values for the 5 and 20 second prediction horizons. This potentially stems from the committees having higher precision values in absolute terms for the 60 second prediction horizon, see Tables 6.2, 6.3, 6.4 even if the precision values are smaller relative to random chance.

Similarly to the committee accuracies discussed in Section 7.1, the trading strategies perform better when considering SVM specific committees. Here a maximum annualized return of 120\% is obtained with a corresponding AUC
of 0.619 for the SVM EM50 committee on 5 second predictions, see Figures B.1, B.2 and Table B.1. Lastly it is noted that when introducing transaction costs the performances of the trading strategies are significantly impaired.

7.3 Future Work and Improvements

In general there are three areas which are subject to possible improvements, the base learners, the meta learners, and the trading strategy.

Firstly, as the meta learners are mostly limited by the performance of the base learners it is suggested that focus is put on improving the base learners. For this purpose, there are possible changes to be made in how the data is used. A more pragmatic approach of selecting training windows as well as features can be implemented. In this thesis, this has been done by trial and error but there exist more scientific approaches such as the ones discussed in [20]. Further improvements for the SVM performances might be achieved by studying the kernel in more detail. For example, investigating the approach of combining kernels by multiple kernel learning as it is done in [9]. Similarly, there are extensions to the neural network framework that are promising, such as Deep Neural Networks in general and Convolutional Neural Networks in particular. Furthermore, different class setups can be implemented. One alternative is to divide the target values in only two classes, upwards and downwards movement or increasing the threshold for $C_{+1}$ or $C_{-1}$, in an effort to capture larger movements. Another, as discussed above, very promising alternative is to implement binary classification with $C_{movement}$ and $C_{no-movement}$.

Secondly, the training of the committees can be expanded upon. One suggestion to potentially improve the committee is to add more members to the committee. Another suggestion is to use reinforcement learning which updates the weights of the committee members on-line depending on the realized precision performance on the test set, which can be used in conjunction with a trading strategy.
Chapter 8

Conclusion

This thesis applies different committees of Artificial Neural Networks and Support Vector Machines on high-dimensional, high-frequency EUR/USD exchange rate data in an effort to predict directional market movements on 5, 20 and 60 seconds prediction horizons.

Results show that it is possible to classify future market movements significantly better than random chance. While the precision increases in absolute number for longer prediction horizons, the precision relative to random chance is higher for shorter prediction horizons. The precisions achieved in this thesis outperform results previously reported.

The combination of multiple classifiers into a committee yields better precision than even the best performing base classifier, provided that the committee members perform well individually and have sufficiently high diversity. The study suggests that due to higher diversity Support Vector Machines are better suited for the creation of committees even though Artificial Neural Networks achieve higher precisions individually. These findings confirm what has been shown in previous studies as well as in ensemble learning theory.

Finally, a trading simulation implementing the committee classifier, disregarding transaction costs, yields an annualized return of up to 53%, beating the Buy and Hold benchmark and the Previous Signal benchmark, and highlighting the possibility of developing a profitable trading strategy based on the limit order book and historical transactions alone.
Appendix A

Classification Performance

SVM Committees

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>7429</td>
<td>33837</td>
<td>11781</td>
<td>0.18</td>
<td>0.39</td>
<td>0.25</td>
</tr>
<tr>
<td>$C_{0}$</td>
<td>78043</td>
<td>5540</td>
<td>62669</td>
<td>0.93</td>
<td>0.55</td>
<td>0.70</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>9751</td>
<td>44549</td>
<td>9476</td>
<td>0.18</td>
<td>0.51</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Macro-average: 0.43  0.48  0.46

Table A.1: Performance measurements of the SVM only EM50 Committee for 5 second forecasts. 28.9% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>1880</td>
<td>7660</td>
<td>3970</td>
<td>0.20</td>
<td>0.32</td>
<td>0.24</td>
</tr>
<tr>
<td>$C_{0}$</td>
<td>48586</td>
<td>2626</td>
<td>15946</td>
<td>0.95</td>
<td>0.75</td>
<td>0.84</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>3219</td>
<td>12580</td>
<td>2950</td>
<td>0.20</td>
<td>0.52</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Macro-average: 0.45  0.53  0.49

Table A.2: Performance measurements of the SVM only EM80 Committee for 5 second forecasts. 69.6% of the data was not classifiable.
APPENDIX A. CLASSIFICATION PERFORMANCE

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>11351</td>
<td>26312</td>
<td>33232</td>
<td>0.30</td>
<td>0.25</td>
<td>0.28</td>
</tr>
<tr>
<td>$C_0$</td>
<td>68649</td>
<td>21409</td>
<td>47335</td>
<td>0.76</td>
<td>0.59</td>
<td>0.67</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>23705</td>
<td>54069</td>
<td>21250</td>
<td>0.30</td>
<td>0.53</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Macro-average | 0.46   | 0.46   | 0.46   |

Table A.3: Performance measurements of the SVM only EM50 Committee for 20 second forecasts. 18.4% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>611</td>
<td>1203</td>
<td>5582</td>
<td>0.34</td>
<td>0.10</td>
<td>0.15</td>
</tr>
<tr>
<td>$C_0$</td>
<td>32533</td>
<td>6895</td>
<td>3268</td>
<td>0.83</td>
<td>0.91</td>
<td>0.86</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>2514</td>
<td>4762</td>
<td>4010</td>
<td>0.35</td>
<td>0.39</td>
<td>0.36</td>
</tr>
</tbody>
</table>

Macro-average | 0.50   | 0.46   | 0.48   |

Table A.4: Performance measurements of the SVM only EM80 Committee for 20 second forecasts. 80.7% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>22793</td>
<td>38388</td>
<td>43073</td>
<td>0.37</td>
<td>0.35</td>
<td>0.36</td>
</tr>
<tr>
<td>$C_0$</td>
<td>41193</td>
<td>27142</td>
<td>39396</td>
<td>0.60</td>
<td>0.51</td>
<td>0.55</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>32022</td>
<td>52338</td>
<td>35399</td>
<td>0.38</td>
<td>0.47</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Macro-average | 0.45   | 0.44   | 0.45   |

Table A.5: Performance measurements of the SVM only EM50 Committee for 60 second forecasts. 15.1% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>5191</td>
<td>7780</td>
<td>12717</td>
<td>0.40</td>
<td>0.29</td>
<td>0.34</td>
</tr>
<tr>
<td>$C_0$</td>
<td>20191</td>
<td>10648</td>
<td>7963</td>
<td>0.65</td>
<td>0.72</td>
<td>0.68</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>8361</td>
<td>12666</td>
<td>10414</td>
<td>0.40</td>
<td>0.45</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Macro-average | 0.48   | 0.48   | 0.48   |

Table A.6: Performance measurements of the SVM only EM80 Committee for 60 second forecasts. 74.3% of the data was not classifiable.
## ANN Committees

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>7521</td>
<td>38431</td>
<td>16607</td>
<td>0.16</td>
<td>0.31</td>
<td>0.21</td>
</tr>
<tr>
<td>$C_0$</td>
<td>117706</td>
<td>16173</td>
<td>58724</td>
<td>0.88</td>
<td>0.67</td>
<td>0.76</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>8639</td>
<td>35947</td>
<td>15220</td>
<td>0.19</td>
<td>0.36</td>
<td>0.25</td>
</tr>
<tr>
<td>Macro-average</td>
<td></td>
<td></td>
<td></td>
<td>0.41</td>
<td>0.45</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table A.7: Performance measurements of the ANN only EM50 Committee for 5 second forecasts. 10.9% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>2869</td>
<td>14805</td>
<td>10966</td>
<td>0.16</td>
<td>0.21</td>
<td>0.18</td>
</tr>
<tr>
<td>$C_0$</td>
<td>101093</td>
<td>12123</td>
<td>26897</td>
<td>0.89</td>
<td>0.79</td>
<td>0.84</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>5010</td>
<td>19551</td>
<td>8616</td>
<td>0.20</td>
<td>0.37</td>
<td>0.26</td>
</tr>
<tr>
<td>Macro-average</td>
<td></td>
<td></td>
<td></td>
<td>0.42</td>
<td>0.45</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Table A.8: Performance measurements of the ANN only EM80 Committee for 5 second forecasts. 38.3% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>11187</td>
<td>29063</td>
<td>41662</td>
<td>0.28</td>
<td>0.21</td>
<td>0.24</td>
</tr>
<tr>
<td>$C_0$</td>
<td>82432</td>
<td>36070</td>
<td>46012</td>
<td>0.70</td>
<td>0.64</td>
<td>0.67</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>23547</td>
<td>51975</td>
<td>29434</td>
<td>0.31</td>
<td>0.44</td>
<td>0.37</td>
</tr>
<tr>
<td>Macro-average</td>
<td></td>
<td></td>
<td></td>
<td>0.43</td>
<td>0.43</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table A.9: Performance measurements of the ANN only EM50 Committee for 20 second forecasts. 7.0% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>6320</td>
<td>15606</td>
<td>33053</td>
<td>0.29</td>
<td>0.16</td>
<td>0.21</td>
</tr>
<tr>
<td>$C_0$</td>
<td>69680</td>
<td>27631</td>
<td>30302</td>
<td>0.72</td>
<td>0.70</td>
<td>0.71</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>19160</td>
<td>40423</td>
<td>20305</td>
<td>0.32</td>
<td>0.49</td>
<td>0.39</td>
</tr>
<tr>
<td>Macro-average</td>
<td></td>
<td></td>
<td></td>
<td>0.44</td>
<td>0.45</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Table A.10: Performance measurements of the ANN only EM80 Committee for 20 second forecasts. 29.0% of the data was not classifiable.
## APPENDIX A. CLASSIFICATION PERFORMANCE

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>12553</td>
<td>23325</td>
<td>55649</td>
<td>0.35</td>
<td>0.18</td>
<td>0.24</td>
</tr>
<tr>
<td>$C_{0}$</td>
<td>51123</td>
<td>47893</td>
<td>31560</td>
<td>0.52</td>
<td>0.62</td>
<td>0.56</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>32262</td>
<td>53503</td>
<td>37512</td>
<td>0.38</td>
<td>0.46</td>
<td>0.41</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.41</td>
<td>0.42</td>
<td>0.42</td>
<td>0.41</td>
<td>0.42</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Table A.11: Performance measurements of the ANN only EM50 Committee for 60 second forecasts. 12.4% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>5124</td>
<td>9836</td>
<td>35848</td>
<td>0.34</td>
<td>0.13</td>
<td>0.18</td>
</tr>
<tr>
<td>$C_{0}$</td>
<td>37446</td>
<td>32652</td>
<td>16151</td>
<td>0.53</td>
<td>0.70</td>
<td>0.61</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>20243</td>
<td>31584</td>
<td>22073</td>
<td>0.39</td>
<td>0.48</td>
<td>0.43</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.42</td>
<td>0.43</td>
<td>0.43</td>
<td>0.43</td>
<td>0.43</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table A.12: Performance measurements of the ANN only EM80 Committee for 60 second forecasts. 45.7% of the data was not classifiable.
### Full Committees

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>6937</td>
<td>30496</td>
<td>14373</td>
<td>0.19</td>
<td>0.33</td>
<td>0.24</td>
</tr>
<tr>
<td>$C_0$</td>
<td>107391</td>
<td>11293</td>
<td>52217</td>
<td>0.90</td>
<td>0.67</td>
<td>0.77</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>9118</td>
<td>36678</td>
<td>11877</td>
<td>0.20</td>
<td>0.43</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Macro-average 0.43 0.48 0.45

Table A.13: Performance measurements of the Full EM50 Committee for 5 second forecasts. 19.9% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>1554</td>
<td>5913</td>
<td>4739</td>
<td>0.21</td>
<td>0.25</td>
<td>0.23</td>
</tr>
<tr>
<td>$C_0$</td>
<td>64004</td>
<td>3908</td>
<td>12810</td>
<td>0.94</td>
<td>0.83</td>
<td>0.88</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>2986</td>
<td>11027</td>
<td>3299</td>
<td>0.21</td>
<td>0.48</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Macro-average 0.45 0.52 0.48

Table A.14: Performance measurements of the Full EM80 Committee for 5 second forecasts. 64.5% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>8439</td>
<td>37696</td>
<td>15832</td>
<td>0.18</td>
<td>0.35</td>
<td>0.24</td>
</tr>
<tr>
<td>$C_0$</td>
<td>112571</td>
<td>12899</td>
<td>61156</td>
<td>0.90</td>
<td>0.65</td>
<td>0.75</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>9844</td>
<td>40571</td>
<td>14178</td>
<td>0.20</td>
<td>0.41</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Macro-average 0.43 0.47 0.45

Table A.15: Performance measurements of the Full CWM50 Committee for 5 second forecasts. 11.9% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>1550</td>
<td>5913</td>
<td>4630</td>
<td>0.21</td>
<td>0.25</td>
<td>0.23</td>
</tr>
<tr>
<td>$C_0$</td>
<td>63978</td>
<td>3904</td>
<td>12617</td>
<td>0.94</td>
<td>0.84</td>
<td>0.89</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>2901</td>
<td>10725</td>
<td>3295</td>
<td>0.21</td>
<td>0.47</td>
<td>0.29</td>
</tr>
</tbody>
</table>

Macro-average 0.45 0.52 0.48

Table A.16: Performance measurements of the Full CWM80 Committee for 5 second forecasts. 64.7% of the data was not classifiable.


### Table A.17: Performance measurements of the Full EM50 Committee for 20 second forecasts. 14.5% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>8188</td>
<td>18778</td>
<td>39328</td>
<td>0.30</td>
<td>0.17</td>
<td>0.22</td>
</tr>
<tr>
<td>$C_0$</td>
<td>81701</td>
<td>31595</td>
<td>38829</td>
<td>0.72</td>
<td>0.68</td>
<td>0.70</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>23741</td>
<td>51321</td>
<td>23537</td>
<td>0.32</td>
<td>0.50</td>
<td>0.39</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table A.18: Performance measurements of the Full EM80 Committee for 20 second forecasts. 69.8% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>574</td>
<td>1338</td>
<td>11462</td>
<td>0.30</td>
<td>0.05</td>
<td>0.08</td>
</tr>
<tr>
<td>$C_0$</td>
<td>43591</td>
<td>10070</td>
<td>7877</td>
<td>0.81</td>
<td>0.85</td>
<td>0.83</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>6980</td>
<td>13630</td>
<td>5699</td>
<td>0.34</td>
<td>0.55</td>
<td>0.42</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table A.19: Performance measurements of the Full CWM50 Committee for 20 second forecasts. 13.5% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>8173</td>
<td>18729</td>
<td>39915</td>
<td>0.30</td>
<td>0.17</td>
<td>0.22</td>
</tr>
<tr>
<td>$C_0$</td>
<td>83241</td>
<td>33206</td>
<td>38560</td>
<td>0.71</td>
<td>0.68</td>
<td>0.70</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>23609</td>
<td>50942</td>
<td>24402</td>
<td>0.32</td>
<td>0.49</td>
<td>0.39</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table A.20: Performance measurements of the Full CWM80 Committee for 20 second forecasts. 69.8% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>571</td>
<td>1333</td>
<td>11454</td>
<td>0.30</td>
<td>0.05</td>
<td>0.08</td>
</tr>
<tr>
<td>$C_0$</td>
<td>43591</td>
<td>10070</td>
<td>7863</td>
<td>0.81</td>
<td>0.85</td>
<td>0.83</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>6969</td>
<td>13610</td>
<td>5696</td>
<td>0.34</td>
<td>0.55</td>
<td>0.42</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A.21: Performance measurements of the Full EM50 Committee for 60 second forecasts. 17.3% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>14727</td>
<td>23601</td>
<td>49320</td>
<td>0.38</td>
<td>0.23</td>
<td>0.29</td>
</tr>
<tr>
<td>$C_0$</td>
<td>49320</td>
<td>39641</td>
<td>29715</td>
<td>0.55</td>
<td>0.62</td>
<td>0.59</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>30961</td>
<td>50111</td>
<td>34318</td>
<td>0.38</td>
<td>0.47</td>
<td>0.42</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.44</td>
<td>0.44</td>
<td>0.44</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.22: Performance measurements of the Full EM80 Committee for 60 second forecasts. 79.5% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>888</td>
<td>1407</td>
<td>11704</td>
<td>0.39</td>
<td>0.07</td>
<td>0.12</td>
</tr>
<tr>
<td>$C_0$</td>
<td>21913</td>
<td>12897</td>
<td>3951</td>
<td>0.63</td>
<td>0.85</td>
<td>0.72</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>5789</td>
<td>8774</td>
<td>7423</td>
<td>0.40</td>
<td>0.44</td>
<td>0.42</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.47</td>
<td>0.45</td>
<td>0.46</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.23: Performance measurements of the Full CWM50 Committee for 60 second forecasts. 17.2% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>14037</td>
<td>22805</td>
<td>50294</td>
<td>0.38</td>
<td>0.22</td>
<td>0.28</td>
</tr>
<tr>
<td>$C_0$</td>
<td>50610</td>
<td>42321</td>
<td>28815</td>
<td>0.54</td>
<td>0.64</td>
<td>0.59</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>29870</td>
<td>49017</td>
<td>35034</td>
<td>0.38</td>
<td>0.46</td>
<td>0.42</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.43</td>
<td>0.44</td>
<td>0.44</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.24: Performance measurements of the Full CWM80 Committee for 60 second forecasts. 76.7% of the data was not classifiable.

<table>
<thead>
<tr>
<th>Class</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{+1}$</td>
<td>899</td>
<td>1381</td>
<td>14024</td>
<td>0.39</td>
<td>0.06</td>
<td>0.10</td>
</tr>
<tr>
<td>$C_0$</td>
<td>23831</td>
<td>14609</td>
<td>4752</td>
<td>0.62</td>
<td>0.83</td>
<td>0.71</td>
</tr>
<tr>
<td>$C_{-1}$</td>
<td>6996</td>
<td>11007</td>
<td>8221</td>
<td>0.39</td>
<td>0.46</td>
<td>0.42</td>
</tr>
<tr>
<td>Macro-average</td>
<td>0.47</td>
<td>0.45</td>
<td>0.46</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

Trading Performance
APPENDIX B. TRADING PERFORMANCE

Figure B.1: TSA of SVM and ANN committees for $\Delta t = 5$.

Figure B.2: Trading Performances of SVM committees for $\Delta t = 5$.

Figure B.3: Trading Performances of ANN committees for $\Delta t = 5$.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Return</th>
<th>Transactions</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM EM50</td>
<td>1.201</td>
<td>5338</td>
<td>0.619</td>
</tr>
<tr>
<td>SVM EM80</td>
<td>0.268</td>
<td>1098</td>
<td>0.704</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>0.179</td>
<td>1624</td>
<td>0.604</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>-0.436</td>
<td>412</td>
<td>0.604</td>
</tr>
</tbody>
</table>

Table B.1: Performance and Accuracy measures for SVM and ANN committees for $\Delta t = 5$
Figure B.4: TSA of SVM and ANN committees for $\Delta t = 20$.

Figure B.5: Trading Performances of SVM committees for $\Delta t = 20$.

Figure B.6: Trading Performances of ANN committees for $\Delta t = 20$.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Return</th>
<th>Transactions</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM EM50</td>
<td>0.241</td>
<td>2012</td>
<td>0.572</td>
</tr>
<tr>
<td>SVM EM80</td>
<td>−0.069</td>
<td>70</td>
<td>0.585</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>−0.273</td>
<td>370</td>
<td>0.528</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>−0.168</td>
<td>96</td>
<td>0.566</td>
</tr>
</tbody>
</table>

Table B.2: Performance and Accuracy measures for SVM and ANN committees for $\Delta t = 20$
APPENDIX B. TRADING PERFORMANCE

Figure B.7: TSA of SVM and ANN committees for $\Delta t = 60$.

Figure B.8: Trading Performances of SVM committees for $\Delta t = 60$.

Figure B.9: Trading Performances of ANN committees for $\Delta t = 60$.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Return</th>
<th>Transactions</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM EM50</td>
<td>-0.020</td>
<td>737</td>
<td>0.617</td>
</tr>
<tr>
<td>SVM EM80</td>
<td>0.256</td>
<td>70</td>
<td>0.716</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>-0.394</td>
<td>118</td>
<td>0.452</td>
</tr>
<tr>
<td>ANN EM50</td>
<td>-0.119</td>
<td>18</td>
<td>0.556</td>
</tr>
</tbody>
</table>

Table B.3: Performance and Accuracy measures for SVM and ANN committees for $\Delta t = 60$
Bibliography


