TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JUHA MATALA-AHO

MATHEMATICAL MODEL FOR SAWNWOOD DEMAND FORECASTING

Master of Science thesis

Examiner: Prof. Juho Kanniainen
Examiner and topic approved by the Faculty Council of the Faculty of Faculty of Business and Built Environment
on 25th September 2017

ABSTRACT

JUHA MATALA-AHO: Mathematical model for sawnwood demand forecasting
Tampere University of Technology
Master of Science thesis, 51 pages, 6 Appendix pages
September 2017
Master's Degree Programme in Industrial Engineering and Management
Major: Industrial Management
Examiner: Prof. Juho Kanniainen
Keywords: Forecasting, Sawnwood, Machine Learning, Time series

Ability to predict the sawnwood demand provides competitive advantage for sawnwood producers. It helps sawnwood producers to better manage the supply against the demand in the markets they operate in. This thesis studied sawnwood demand forecasting based on machine learning approaches. The goal of the study was to examine how well different machine learning models are able to predict sawnwood demand and how does the performance of the models differ in different markets?

The final model is an ensemble of machine learning models which takes the weighted sum of the predictions produced by five different machine learning algorithms: the K nearest neighbours, the Random forest, the Support vector with radial basis function kernel, the Support vector machine with polynomial kernel and the Neural network. Six different variables were given as input features for the model. The performance of model was evaluated based on a case study in which four different data sets were used for testing the prediction accuracy of the model. The performance of the models was measured with three error metrics the MAPE, the MAE and the RMSE. In addition, the developed ensemble model was compared with the individual learning algorithms and a naive forecast.

The results show that the Ensemble estimator outperforms the five individual learning algorithms and the Naive forecast measured in all three error metrics when the errors are calculated as the average of the four data sets. However, when the results are compared at the individual data set level, the Ensemble estimator performs the best only on four out of the twelve cases. The results indicate that a single method cannot provide the best answer in all of the cases. In addition, the performance of the models vary when the results are compared by taking the moving average of the predicted values. The error rates decrease more for more advanced learning algorithms like the Support vector machines, the Neural network and the Ensemble estimator. This indicates that these models are able to capture the trend component better from the data sets. Finally, the study shows that there are differences how well the models can predict the sawnwood demand in different markets. The effect of the data sets' characteristics on the prediction accuracy of the models decreases for more advanced models, when the data sets are aggregated.

TIIVISTELMÄ

JUHA MATALA-AHO: Matemaattinen malli sahatavaran kysynnän ennustamiseen
Tampereen teknillinen yliopisto
Diplomityö, 51 sivua, 6 liitesivua
Syyskuu 2017
Tuotantotalouden koulutusohjelma
Pääaine: Teollisuustalous
Tarkastajat: Prof. Juho Kanniainen
Avainsanat: Ennustemalli, Sahatavara, Koneoppiminen, Aikasarja

Sahatavaran tuottajalle kyky ennustaa sahatavaran kysyntää tarjoaa kilpailuedun, koska tällöin tuottaja pystyy vastaamaan paremmin markkinoiden kysyntään kohdentamalla toimituksiaan oikeille markkinoille. Tässä diplomityössä tutkittiin koneoppimismenetelmien soveltamista sahatavaran kysynnän ennustamiseen. Työn tavoitteena oli tarkastella, kuinka hyvin erilaiset koneoppimismenetelmät pystyvät ennustamaan sahatavaran kysyntää ja kuinka mallien suorituskyky eroaa eri markkinoilla.

Työssä kehitettiin erilaisia koneoppimismenetelmiä yhdistävä kokoomamalli, jonka tuottama ennuste on painotettu keskiarvo viidestä eri menetelmästä: lähinaapurimenetelmästä, satunnaismetsästä, tukivektorikoneesta radiaalisella ydinfunktiolla, tukivektorikoneesta polynomisella ydinfunktiolla ja neuroverkosta. Mallit käyttävät lähtöarvoina kuutta eri markkinaindikaattoria. Mallien ennustetarkkuutta arvioitiin tapaustutkimuksessa, jossa malleilla tuotettiin ennuste neljälle eri aikasarjalle. Ennustetarkkuuden mittaus suoritettiin kolmella eri tunnusluvulla: keskimääräisellä prosentuaalisessa virheellä, keskimääräisellä absoluuttisella virheellä ja keskineliövirheen neliöjuurella. Tämän lisäksi mallien ennusteita verrattiin yksinkertaiseen ennusteeseen.

Tulokset osoittavat, että kokoomaennusteella saavutetaan pienempi ennustevirhe kuin yksittäisillä menetelmillä tai yksinkertaisella ennusteella, kun ennustevirheet lasketaan neljälle aikasarjalle tuotettujen ennusteiden keskiarvona. Verrattaessa yksittäisten aikasarjojen ennusteita keskenään, kokoomaennuste tuottaa parhaan ennusteen kuitenkin vain neljässä tutkituista kahdestatoista tapauksesta. Tuloksesta voidaan päätellä, ettei yksi yksittäinen malli pystyne tuottamaan pienintä ennustevirhettä kaikille eri aikasarjoille. Lisäk-si tulosten tarkkuus vaihtelee, jos ennustevirheet lasketaan ennusteiden liukuvista keskiarvoista. Kehittyneempien ennusteiden, kuten tukivektorikoneiden, neuroverkkojen ja kokoomaennusteiden, ennustevirheet pienenevät enemmän kuin yksinkertaisempien mallien, kun liukuvaan keskiarvoa lasketaan pidemmän aikavälin keskiarvoista. Tulos viittaa siihen, että kehittyneet mallit pystyvät tunnistamaan paremmin aikasarjojen trendikomponentin. Tulokset osoittavat myös, että mallien kyky ennustaa kysyntää vaihtelee markkinoittain. Kehittyneemmillä malleilla vaihtelu kuitenkin pienenee, kun ennustevirheet lasketaan pidemmän aikavälin keskiarvoista.

PREFACE

This thesis project was great opportunity to develop myself professionally. I could not have imagined a better topic for my masters thesis project. Therefore, I would like to thank, first of all, my instructors Kimmo Mikkonen, Jarkko Taskinen, Timmo Blauberg and Anssi Käki for supporting the work and providing the chance to work with such an interesting topic. Second, I would like to thank my supervisor Juho Kanniainen for the guidance in structuring the thesis and sharpening my arguments. Last, I would like to thank Paula for all the support in finishing this thesis.

Helsinki, August 19, 2017

Juha Matala-aho

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

AR	Autoregression
ARCH	Autoregressive conditional heteroscedasticity
DIRMO	Direct multi-input multi-output
EE	Ensemble estimator
GARCH	Generalized autoregressive conditional heteroscedasticity
HS	Harmonized system
KNN	K nearest neighbours
MA	Moving average
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MIMO	Multi-input multi-output
MSE	Mean squared error
NN	Neural network
OECD	Organization for Economic Cooperation and Development
RBF	Radial basis function
RF	Random forest
RMSE	Root mean squared error
SITC	Standard international trade classification
SVM	Support vector machine
VAR	Vector auto-regression

1. INTRODUCTION

Sawnwood production is an important employer in many European economies. The industry supplies raw material to construction industry, furniture manufacturing and prefabricated house production and it affects the prices and the supply of other forest sector products like pulp, paper, bioenergy, wood-based panels and veneer. Sawnwood production and consumption are diverged globally and the demand for sawnwood varies significantly depending on the economic activity of the industries which consume sawnwood.

In order to balance the supply and the demand on different markets, the sawnwood producers have to make choices, to which markets they allocate their production. However, this is a complex task because the demand and supply of sawnwood may vary significantly in time and place. For example, in year 2000 the sawnwood consumption was around 30 million cubic meters both in Denmark and in Poland (Figure 1.1). By year 2015 the sawnwood consumption had decreased to 20 million cubic meters in Denmark while in Poland the consumption had increased to 40 million cubic meters. The 10 million cubic meters' change in yearly consumption corresponds to the average yearly production of sawnwood in Finland between years 2010 and 2015 [30]. It is important for the sawnwood producers to detect these kind of changes, and if possible also to predict them so that they can better plan their sales efforts.

Predicting the changes in the sawnwood demand requires information about the market factors, like construction activity, income and price, that have an effect on the sawnwood demand [36]. This kind of market data is publicly available today. Multiple different organizations like Eurostat, OECD and the United Nations provide data for example on building permits, housing loans and production of furniture. The data can be accessed readily through application programming interfaces and processed efficiently with various programming tools. Combining the data with modern mathematical models, a system for predicting sawnwood demand can be developed. This thesis will seize this opportunity by studying the sawnwood demand prediction on two different markets based on machine learning methods.

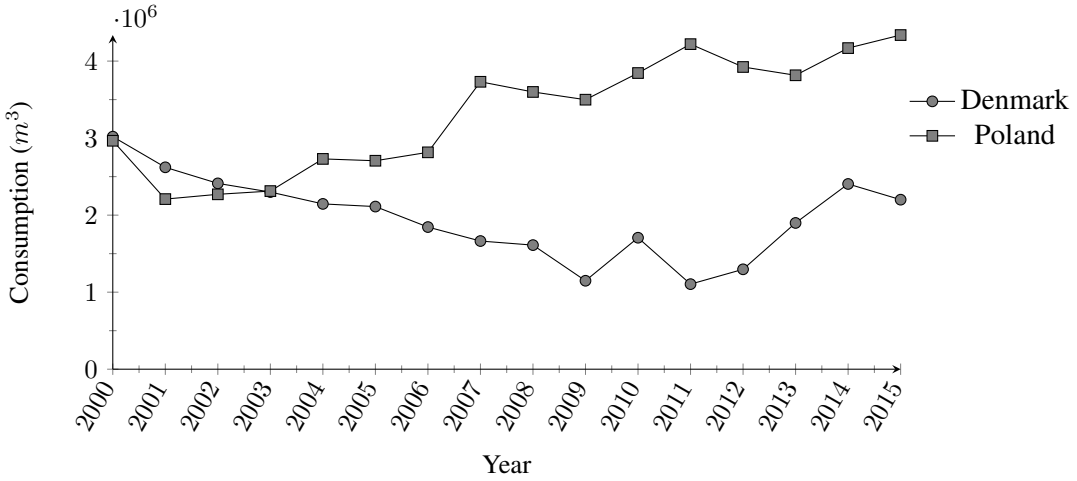


Figure 1.1 *Sawnwood consumption in Denmark and Poland between 2000 and 2015. The consumption is calculated as an apparent consumption (Apparent consumption = Production + Imports - Exports). The data is collected from the United Nations' Comtrade database (Available at: https://comtrade.un.org).*

1.1 Problem statement and thesis motivation

The objective of this thesis is to design a mathematical model for forecasting sawnwood demand in selected markets. The aim is to predict the demand at a monthly interval for the next twelve months. The forecast is made based on selected leading indicators and historical demand values. The initial goal for this thesis is to establish a systematic way to predict sawnwood demand at a monthly interval which can be used as a base scenario for future sawnwood demand forecasting efforts. In a more detailed level, this thesis tries to find an answer for the following research questions. How well different machine learning models are able to predict sawnwood demand and how does the performance of the models differ for different markets?

There is some empirical research on sawnwood demand and price forecasting [1]. The scholars have analyzed and predicted the demand or prices mainly on yearly interval and using classical statistical methods like multivariate regression [36], autoregressive [43, 56, 49] and vector autoregressive [66, 34] models, or econometric models based on dynamic partial equilibrium framework [15]. The studies have found that the consumption of sawnwood is driven by multiple different economic indicators and there are large regional differences which indicator has the biggest effect [36]. In addition to this, studies show that the prices of lumber products might undergo different kinds of structural

[1] An extensive listing of the studies for the forest sector demand and price forecasting can be found from the studies conducted by K. Niquidet & L. Sun *Do Forest Products Prices Display Long Memory?*, Canadian Journal of Agricultural Economics, 2015 or J. Buongiorno, *Forest sector modeling: a synthesis of econometrics, mathematical programming, and system dynamics methods*, International Journal of Forecasting, vol. 12, no.3, pp. 329–343, Sep. 1996.

changes and they might have both mean-reverting and non-stationary properties [15, 51]. In general, sawnwood is a commodity product and the consumption of sawnwood is well-established in many markets. In addition to the common market factors, the demand of sawnwood depends on the type of end-use. The type of end-use may vary also from market to market which adds up complexity for creating a sawnwood demand forecasting model. Due to the differences in the drivers of the sawnwood demand and the structural changes in the demand, a single model is not likely to work well for predicting the sawnwood demand in all markets and through the time. Thus, a model which can adapt to these changing conditions is needed. Machine learning methods could offer a possible solution to this problem.

However, there are no well-known studies that have applied more modern machine learning methods for forecasting the demand of sawnwood or other forest products. Hence, this thesis is inspired by the work of scientists in the field of machine learning and especially a thesis written in MIT by Runmin Xu which presents a model for taxi demand forecasting in New York area at a hourly interval [69]. The same principles are applied in this thesis for sawnwood demand forecasting at a monthly interval.

1.2 Thesis outline

After the introduction, this thesis is organized as follows. Chapter 2 gives a brief overview of the time series forecasting and different techniques applied to it. Section 2.1 introduces the theory of time series forecasting, after which Section 2.2 and Section 2.3 present how the forecasting problem has been tried to solve by statistical and machine learning methods. Finally, Section 2.4 describes the different strategies that can be applied for the multi-step time series forecasting.

Chapter 3 describes the different kinds of machine learning algorithms which are used in demand forecasting in this thesis. The chapter is divided into two sections. First, Section 3.1 gives a more detailed outlook on the individual learning algorithms. After this, Section 3.2 explains how these individual models can be combined into an ensemble of the machine learning models.

Chapter 4 takes a closer look at selecting and optimizing hyper-parameters. The chapter represents two strategies, the grid search and the random search, for conducting the hyper-parameter optimization process.

Chapter 5 presents the case study which was done to evaluate the performance of the different models presented in Chapter 3. Section 5.1 describes the content and the pre-processing of the data. Section 5.2 presents the objectives, procedures and the programs

that where used in conducting the experiment. Section 5.3 presents the error metrics that were applied for evaluating the performance of the different models, before Section 5.4 highlights the key findings of the study.

Finally, Chapter 6 provides a conclusion of this thesis and suggests goals for future research.

2. LITERATURE REVIEW

Chapter 2 gives an overview of the time series forecasting and different techniques applied to it. Section 2.1 introduces briefly the concept of time series forecasting, after which two different approaches to the time series forecasting are presented in Sections 2.2 and 2.3. Finally, Section 2.4 describes the different strategies that can be applied for the multi-step time series forecasting.

2.1 Time series forecasting

Time series is a set of data points, each of which presents the value of the same variable at different times, normally at uniform intervals [12]. The data can be recorded continuously through time or as discrete values, for example at daily, monthly or yearly interval [12]. Continuous time series are usually transformed to discrete values by sampling or aggregating the data on a chosen time interval. This helps to simplify further analysis on the data.

Especially many economic time series are characterized by time-dependent components. Therefore, they can be divided into three parts which are the trend, the seasonal and the residual components. The trend of a time series can be defined as a long-term direction of changes in the data. For the economic time series the trends usually last for multiple years [19]. The seasonal component is a systematic or calendar-related effect on the data which can be observed as a repetitive pattern in the data [19]. The residual component is the left-over part of the data which cannot be explained either as a trend or a seasonal variation in the data [19]. Dividing the time series into these three components acts usually as starting point for the time series analysis.

Time series analysis can be used for describing, modelling, controlling or forecasting the chosen data set [12]. This thesis focuses on the time series forecasting where the goal is to create a model which can predict future values based on the past data. Normally these models include a set of parameters which are estimated based on the data. After the model is fitted into the data, the model is applied for extrapolating the future values. Thus, most forecasting models assume that the future behaves like the past. The forecasting models can also be applied for what-if analysis by exploring the effect of changing policy

variables. Also in this case, the forecasting models are dependent on the assumptions which have been made when the model was built.

2.2 Statistical methods

For a long time scholars in the field of economy, mathematics, physics and engineering have applied various statistical models for analyzing and forecasting time series. Next a quick overlook on these models is given. More detailed description of the models can be found in various text-books which cover the topic thoroughly[1].

Simple statistical models are usually based on two linear models called autoregressive (AR) and moving average (MA) models. An AR model assumes that the output variable depends linearly on previous output values and on a stochastic term which cannot be predicted perfectly [12]. Therefore, the AR model can be presented as a stochastic difference equation which can be used for simulating the future values. A MA model uses past forecast errors to forecast future values [12]. The name moving average is technically incorrect since the MA coefficients may not sum to unity and may also be negative [53, 12]. This label is used by convention. Each future value can be thought of as a weighted moving average of the forecast errors. A widely used model combines these two models into a so called ARIMA model which is a short name for the autoregressive integrated moving average model. This model can predict non-stationary time series which means that the time series can have for example seasonal or trend properties [12]. The ARIMA model has been further developed by adding exogenous variables (ARIMAX) or seasonal components to it (SARIMA). A generalization of an AR model, called a vector-auto-regression (VAR), is a popular model especially in economics [48]. The model uses multiple values and their interrelationships as input variables and it can be used for analyzing multivariate time series [48].

The AR and MA based models can be applied for problems where the main interest is in the conditional mean of the given time series, or more exactly in the process that is generating the series. Sometimes, especially in finance, the interest might be more in the conditional variance of the given process, i.e the heteroskedasticity. The heteroskedasticity means that the magnitude of the error terms varies over time. An auto-regressive conditional heteroscedastic (ARCH) and a generalized auto-regressive conditional heteroscedastic (GARCH) models focus on modelling this kind of behaviour [24, 10]. The ARCH and GARCH models can also be applied for predicting non-linear time series or

[1] See for example: H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*, Springer Science & Business Media, 2005 or G. Box et al., *Time Series Analysis: Forecasting and Control,* $5^{th} Edition, Wiley, 2015$.

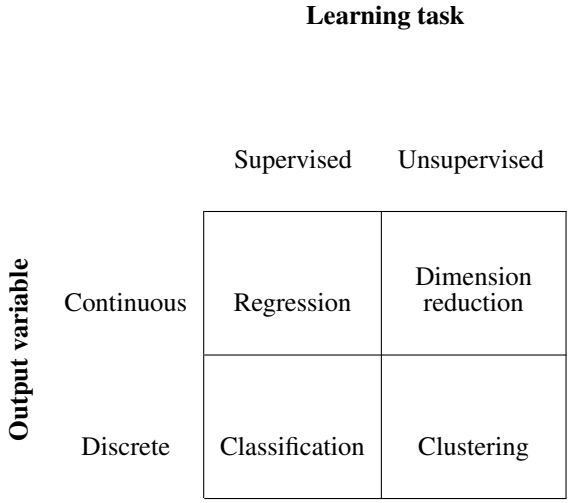


Figure 2.1 *Types of machine learning problems. Machine learning problems can be divided into four groups based on the type of the learning task or the type of the output variable (also known as the predicted variable). The type of the learning task (horizontal axis) can be either supervised or unsupervised. The type of the output variable (vertical axis) can be either continuous or discrete.*

ones which have zero auto-correlation at all lags [33]. The non-linearity allows the models to change over time [38]. Since economic and financial systems are known to go through both structural and behavioral changes, it is reasonable to assume that different time series models may be required to explain the empirical data at different times [70].

2.3 Machine learning approaches

Statistical methods assume that the data is generated by a given stochastic model [14]. However, in reality the data mechanism can also be unknown [14]. In these situations applying the statistical methods for analyzing the data becomes increasingly more complex or even impossible task. Instead of relying on hard-coded knowledge systems need the ability to acquire their own knowledge, by extracting patterns from raw data [32]. This capability is known as machine learning [32].

Machine learning algorithms can learn to execute tasks by generalizing from examples [23]. This is a cost effective way compared to manual programming [23]. Machine learning enables the use of historical data for learning stochastic dependency between the past and the future by using nonparametric nonlinear models [11]. Machine learning models are also called as ”black-box” models because there is typically little knowledge of the internal workings of these models [18].

During the last decades they have been challenging increasingly the traditional methods

for the time series forecasting [1] by having better accuracy in their predictions and by introducing open source tools for implementation these systems. In the 1980s, the neural network models were introduced which are today accompanied by the support vector machines, the random forest, the k nearest neighbours and multiple variations of these models.

Machine learning problems can be classified into four different groups based on the type of the learning task and the type of the output variable (Figure 2.1). The time series forecasting is a machine learning problem where the learning task is supervised and the output variable is continuous. This types of problems are also known as regression problems. Most of the modern machine learning models can be applied for supervised learning tasks with both continuous and discrete output variables. This ability separates them from the traditional statistical methods.

The machine learning algorithms are generally good at learning highly complicated patterns from the given data. Sometimes learning can even be too good which leads to a problem called over-fitting [23]. The over-fitting means that the results of the developed model will not generalize very well [23, 22]. The error of training data can get small but as soon as new data is introduced for the model the error gets larger. The over-fitting problem can be explained by decomposing error in terms of bias and variance [22]. Simpler models usually have a low bias and a high variance while more complex models tend to have a higher bias and a lower variance. Choosing the right model for usually means balancing between these two error sources. This choice affects a lot to the behaviour of the prediction model.

In order to avoid the problem of over-fitting, the machine learning models use different kinds of tactics to penalize the algorithm from this behaviour. Two basic ways to reduce the over-fitting are: limiting the number of dimensions of the parameter space and limiting the effective size of each dimension [55]. Techniques, like regularization or early stopping, can be used for limiting the size of each parameter dimension [39, 61, 50]. The number of parameters can be controlled using methods, like greedy constructive learning, pruning or weight sharing [29, 45, 52].

2.4 Multi-step forecasting

Time series forecasting can be complicated process even if the aim is to produce a forecast only one step forward. The process becomes considerably more complex when the goal is changed to produce a forecast for multiple time-steps. In multi-step forecasting the model designer has to choose between different strategies and deal with increased uncertainty and accumulation of errors.

There are at least five basic strategies for multi-step forecasting, named the Recursive, the Direct, the DiRec, the Multi-Input Multi-Output (MIMO) and the DIRMO [4]. The first one of these strategies, the Recursive strategy, produces the forecast by iterating the forecast process n times [4]. After each iteration, the output of the N_{-1} iteration is fed as an input variable for the next iteration N. The process is continued until the forecast is calculated for the entire horizon. Thus, the Recursive strategy can be formulated as follows

$$y_{N+h} = \begin{cases} f(y_N, ..., y_{N-d+1}) & \text{if } h = 1 \\ f(y_{N+h-1}, ..., y_{N+1}, y_n, ..., y_{N-d+h}) & \text{if } h \in \{2, ..., d\} \\ f(y_{N+h-1}, ..., y_{N-d+h}) & \text{if } h \in \{d+1, ..., H\}, \end{cases} \tag{2.1}$$

where N is the number of data points in the time series, H is the forecasting horizon and d is the number of lagged values of the time series used as an input values [4]. The Recursive strategy may suffer from the accumulation of errors because the potentially inaccurate forecast are used as an input for the subsequent time-steps [4]. Since the Recursive strategy is simple to implement, the computing time is relatively low compared to the other strategies [4].

The Direct strategy is another commonly used strategy for the multi-step forecasting [4]. In the Direct strategy, each time step h is predicted based on independent models f_h so that

$$y_{t+h} = f_h(y_t, ..., y_{t-d+1}) + w \tag{2.2}$$

with $t \in \{d, ..., N - H\}$ and $h \in \{1, ..., H\}$ [4]. The w denotes the noise or the error that is included in the forecast [4]. The final prediction is a collection of the predictions calculated based the individual models which can be formulated as follows:

$$y_{N+h} = f_h(y_N, ..., y_{N-d+1}). \tag{2.3}$$

Thus, the Direct strategy is not prone to the accumulation of errors [62]. However, it takes a longer time to compute the forecast with the Direct strategy than with the other strategies, since the Direct strategy requires training of multiple models [62].

The third strategy, called the DiRec strategy, is a combination of the Recursive and the Direct strategies. The DiRec strategy uses different models to compute each forecast

like the Direct strategy. However, similarly to the Recursive strategy, the forecast of the previous time-step y_{t+h-1} is fed as an input variable for the next time-step. This increases the input data set in every time step with one more input variable. Each forecast y_{t+h} is produced thus based on an own model f_h where

$$y_{t+h} = f_h(y_{t+h-1}, ..., y_{t-d+1}) + w \tag{2.4}$$

with $t \in \{d, ..., N - H\}$ and $h \in \{1, ..., H\}$ like with the Direct strategy. The final prediction is a collection of the forecasts produced by the H models so that

$$y_{N+h} = \begin{cases} f_h(y_N, ..., y_{N-d+1}) & \text{if } h = 1 \\ f(y_{N+h-1}, ..., y_{N+1}, y_N, ..., y_{N-d+h}) & \text{if } h \in \{2, ..., H\} \end{cases} \tag{2.5}$$

The DiRec strategy has proven to provide better results than the Recursive strategy and the Direct strategy [63]. However, the model is computationally inefficient, which makes it less suitable for learning problems where the models needs to be retrained continuously.

The Multi-Input Multi-Output (MIMO) strategy uses one model to predict multiple outputs as one operation [5]. The predicted value is a vector of future values $\{y_{t+1}, ..., y_{t+H}\}$ of the time series y_N [5]. The estimation of the H next values are given by

$$[y_{t+H}, ..., y_{t+1}] = F(y_t, ..., y_{t-d+1}) + \mathbf{w}, \tag{2.6}$$

where F is a vector-valued function from R^d to R^H and $\mathbf{w} \in R^H$ is a noise vector [4]. This way, the MIMO strategy can preserve the stochastic dependency characterizing the time series[5]. However, at the same time the flexibility and the variability is reduced compared to the single-output approaches [5]. In addition, the returned model might get biased [5].

The fifth, and last strategy for multi-step forecasting discussed in this thesis, is the DIRMO strategy which is a combination of the Direct and the MIMO strategies. The DIRMO strategy learns n models F_p from the time series $[y_1, ..., y_N]$ where

$$[y_{t+ps}, ..., y_{t+(p-1)s+1}] = F_p(y_t, ..., y_{t-d+1}) + \mathbf{w}, \tag{2.7}$$

with $t \in \{d, ..., N - H\}, p \in \{1, ..., n\}$ and $F_p : R^d \rightarrow R^s$ is a vector-valued function if $s > 1$ [4]. The number of consecutive outputs to be predicted at a time is determined

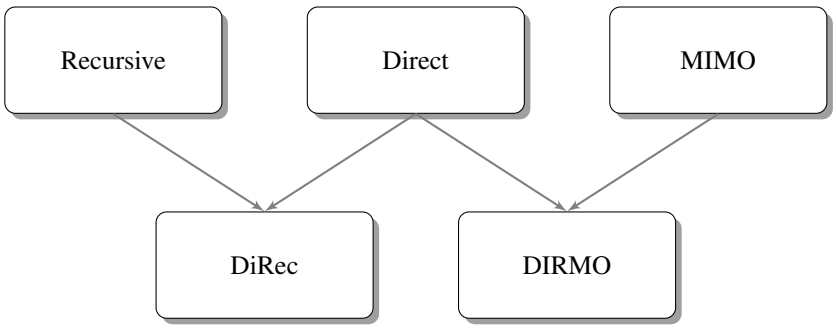


Figure 2.2 *Five different strategies for the multi-step forecasting. There are three basic strategies called the Recursive, the Direct and the MIMO and two combination strategies called the DiRec and the DIRMO.The DiRec is a combination of the Recursive and the Direct strategies while the DIRMO is a combination of the Direct and the MIMO strategies. [4]*

by the variable s. The parameter s helps the DIRMO strategy to decrease the bias of the returned model but at the same time it increases the complexity of the whole forecasting task [65, 5]. The H^{th} forecast can be calculated based on the n learned models as follows:

$$[y_{N+ps}, ..., y_{N+(p-1)s+1}] = F_p(y_N, ..., y_{N-d+1}). \tag{2.8}$$

Figure 2.2 summarizes the different strategies that were discussed in this section and shows how they are interconnected to each other. In this thesis, the Direct strategy is applied for the sawnwood demand forecasting problem because the models developed in Python best support this strategy.

3. MODELS

Chapter 3 describes the different models which are applied for sawnwood demand forecasting in this thesis. There exists multiple different variations of the models discussed in this chapter. This thesis focuses on the basic forms of these models. The chapter is divided into two sections. First, Section 3.1 gives a more detailed outlook on the individual learning algorithms applied in this thesis. After this, Section 3.2 explains how these individual models can be combined into an ensemble of the machine learning models.

3.1 Learning algorithms

3.1.1 Random forests

Random forests are combinations of decision trees [13]. Each tree depends on the values of a random vector, sampled independently and with the same distribution for all trees in the forest [13]. The random forest acts as any other ensemble method based on randomization. They introduce random perturbations into the learning process which produces multiple different models from a single learning data set [46]. These predictions are later combined into a single prediction which should help the model to increase the generalization of the results [46]. The individual randomly sampled decision trees are ideal candidates for ensemble methods because they have high variance and low bias [46]. The random forest is a computationally efficient technique that can operate quickly over large data sets [46].

The random forest algorithm constructs a multitude of decision trees at training time. In general, these decision trees can be represented by a set of questions which divides training data in smaller and smaller data sets. Like the k nearest neighbours algorithm, the decision trees are also non-parametric which means that they can model arbitrarily complex relationships between input and output values without any a priori assumptions [46]. They are easy to interpret even for people who have very little experience of statistics.

The decision tree algorithm divides the data at each node t into two subsets based on threshold parameters [46]. The threshold parameters are chosen so that they minimize an

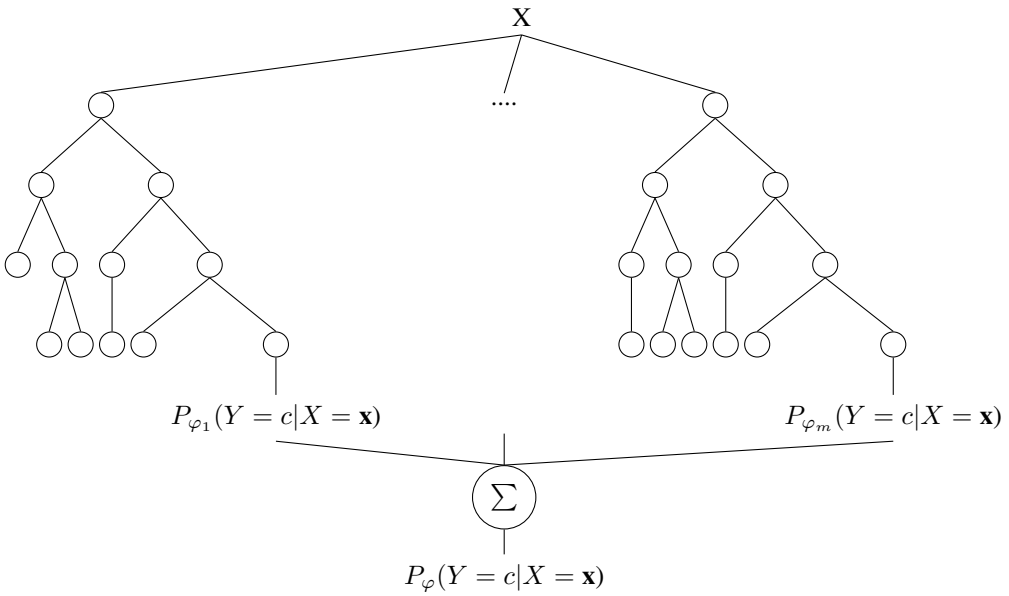


Figure 3.1 *An example of a random forest. X denotes the input variables, Y is the output variable, and P is a partition of the data from which X and Y are drawn. The random forest algorithm constructs a multitude of decision trees at training time and averages the results to improve the predictive accuracy and to control over-fitting. [46]*

impurity function $i(t)$ which depends on the task being solved. For regression problems a common goal is to minimize the mean squared error or the residual sum of squares given that

$$i(t) = \frac{1}{N_t} \sum_{x,y \in \mathcal{L}_t} (y - \widehat{y}_t)^2 \tag{3.1}$$

where N_t is the number of node samples at node t, $\mathcal{L}_t$ is the subset of learning samples falling into the node t, $\widehat{y}_t$ is the prediction for that node and y is the actual value of the output variable Y [46].

After each tree is constructed and trained, the random forest algorithm uses each decision tree to calculate a prediction for the output variable y [46]. These predictions are then combined into a final prediction value by taking an average of them. This process is illustrated in Figure 3.1 [46].

The random forest regression algorithm also has a set of parameters which define how the model behaves. The maximum depth of the tree determines how many levels of nodes the model can have [59]. The larger the maximum depth is, the lower the bias usually gets [59]. The maximum features number defines how many features the model considers when looking for the best split [46]. The number of estimators is used for setting how

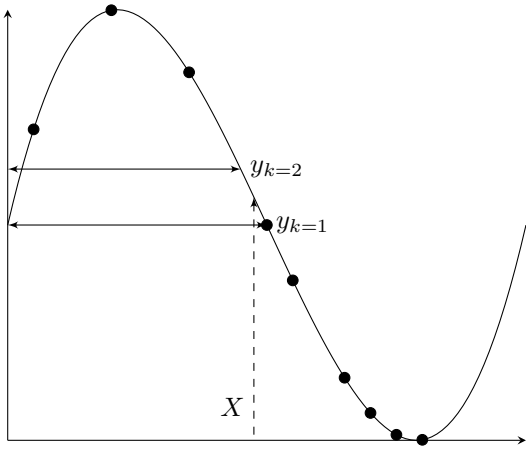


Figure 3.2 *An example of the K nearest neighbours regression. The K nearest neighbours algorithm calculates weighted average of k closest observation. Line $y_{k=1}$ illustrates situation where the $k = 1$ and line $y_{k=2}$ situation where the $k = 2$. The solution for $y_{k=n}$ is given by taking the average of the outcome of observations.*

many trees the forest includes [46]. Using more estimators helps to decrease the variance but at the same time the computation time increases.

In this thesis, a Python package called *sklearn.ensemble.RandomForestRegressor* is used for building the random forest model.

3.1.2 K nearest neighbours

K nearest neighbours (KNN) is a simple extension of the nearest neighbours method, which is a non-parametric pattern recognition method used for classification and regression [31, 2]. The nearest neighbours methods differ from other learning methods because their memory-based approach requires no model to be fit [46]. The nearest neighbours algorithm queries the data set in order to find the closest observations for the given data point in the data set [2]. The algorithm uses an Euclidean distance metric to calculate the distance between the observations [2]. Instead of finding the single closest observation for the given data point, the k nearest neighbours algorithm uses the weighted average of k closest observations so that

$$\varphi(x) = \frac{1}{k} \sum_{(x_i, y_i) \in NN(x,L,k)} y_i, \tag{3.2}$$

where $NN(\mathbf{x}, L, k)$ denotes the k nearest neighbours of x in L [31].

The choice of the parameter k is critical to the performance of the estimator [2]. Larger k leads to a lower variance but at the same time the bias grows larger [2]. In contrast,

by choosing a lower k value, the bias of the estimator is reduced but the variance grows larger [2]. This is due to the fact that the k nearest neighbours algorithm calculates the target output values as the average of the k data points, which lay closest to the given data point [2]. The larger the k is, the more data points are taken into account when calculating the final predicted result.

A Python package called *sklearn.neighbors.KNeighborsRegressor* is used for building the k nearest neighbours model in this thesis.

3.1.3 Neural networks

Neural networks are mathematical models inspired by biological neurons. They have been used for time series forecasting since 1980s and due to the development of new more sophisticated structures, like the long short-term memory (LSTM) networks, the neural networks are likely to provide answers to many time series prediction problems in the future [47]. In general, the neural networks fit well for a variety of time series forecasting problems because they provide flexible computing frameworks and serve as universal function approximators.

The neural networks contain layers of individual neurons which are connected to each other fully or partially. A training algorithm is used to find the weights for each neuron. Based on these weights, the network calculates the weighted sum of the input values and passes the result to the next layer through a non-linear activation function. More formally, the inputs to the node j are linearly combined to give a weighted sum

$$z_j = b_j + \sum_{i=1}^{n} w_{j,i} x_i \tag{3.3}$$

where the weights $b_1, .., b_n$ and coefficients $w_1, .., w_n$ are estimated from the data. The sum is then modified by an activation function, like the $tahn$ function presented in Equation 3.4, to give the input for the next layer.

$$tahn(z) = 2\sigma 2z - 1 \tag{3.4}$$

There are many types of activation functions available for choosing and the performance of the functions can vary depending on the problem to be solved. At the moment, the most promising activation function is a half-wave rectifier called the rectified linear unit (ReLU) [41].

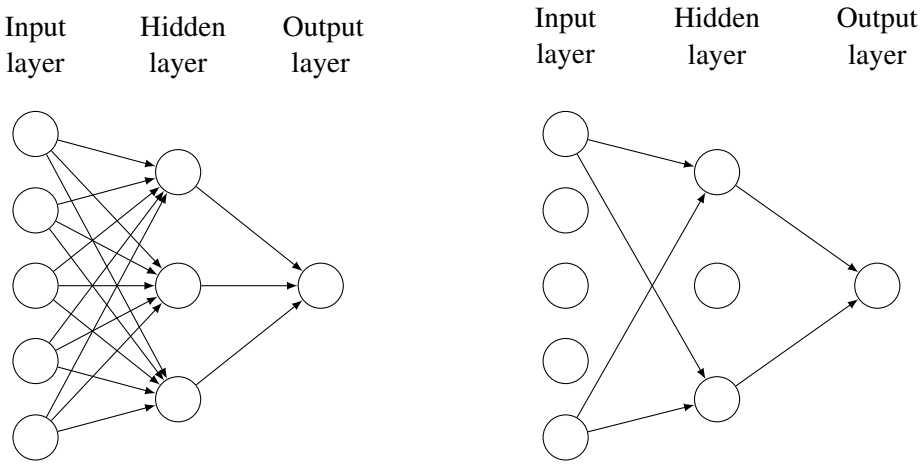


Figure 3.3 *An example of a dropout neural network with an input layer, a hidden layer and an output layer. Lines represent the existing connections between the nodes (or neurons).* ***Left****: A neural network with a hidden layer.* ***Right****: A thinned network with dropped units and connections. [64]*

Like any other regression method, also neural networks suffer from over-fitting. However, there are plenty of options to prevent this effect. For example, the training can be stopped as soon as the validation error starts to get worse. Also, different kinds of penalties, weight decays, can be added to the error term when the model complexity gets higher [52]. For learning problems with a small sample size and a limited amount of computing power, a technique called dropout has proven to perform well [64].

In dropout regularization nodes are randomly dropped out from the neural network during training [64]. The networks with ”thinned” layers are then combined to a single network by approximating the average for the networks. By increasing the dropout-rate p, generalization of the model results can be improved [64]. However, too high drop-out rate makes the network insensitive to changes in the data. Thus, choosing the right the dropout rate affects how well the model will perform.

A Python package called *Keras* is used to build the neural networks used in this thesis. *Keras* provides the features for building basic neural network models with the weight decay and the drop-out regularization.

3.1.4 Support vector machines

Like the neural networks, support vector machines (SVMs) have traditionally been used for classification problems. However, due to their great ability to produce generalized results, SVMs have also been successfully applied for regression problems, like time

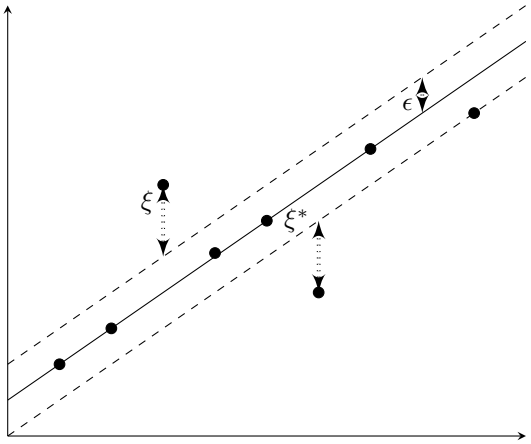


Figure 3.4 *An example of a support vector machine regression with a linear kernel. The support vector machine algorithm finds a hyperplane for which the data points are placed within the tolerated error margin ϵ. The data points that lie outside the dashed lines contribute to the cost, as the deviations are penalized, in the linear kernel case, in a linear fashion by the slack variables ξ and ξ^*. The degree of penalized loss can be modified by a positive constant C. [60]*

series forecasting. In a classification problem, support vector machines search for a hyperplane which produces a maximal margin between the vectors of the two classes [21]. The location of this hyper-plane is determined by the data points which lie closest to the hyperplane. Similarly, in a regression problem, SVMs try to minimize the given error function

$$J = \frac{1}{2}\|\mathbf{w}^2\| + C\sum_{m=1}^{M} Loss(y_m, f(x_m)) \tag{3.5}$$

where **w** is the weight vector, x_m is the m^{th} training input, y_m is the target output and $Loss(y_m, f(x_m))$ is the loss function [60].

Support vector machines use a structural minimization principle to approximate a function by minimizing the upper bound of the generalization error [68, pp.94-96]. This approach helps SVMs to avoid the over-fitting problem and to improve the generalization of the results [68, pp.94-96]. The solution of SVMs is always at the global optimum because training SVMs resembles optimizing a linearly constrained quadratic programming problem [17].

Support vector machines use different kinds of similarity functions, called kernels, to map the feature vectors implicitly into a higher-dimensional space, without explicitly building a higher-dimensional representation [60]. This new enriched representation can be then solved in a simple linear fashion [60]. Thus, SVMs performance is improved in problems where there is a high amount of input features but a low amount of samples [60].

A Python package called *sklearn.svm.SVR* is used for building the two different support vector regression models applied in this thesis [54]. The *sklearn.svm.SVR* package includes four different kernel types, named the linear, the polynomial, the radial basis function and the sigmoid. All these kernels use a penalty parameter ϵ as the error term. The degree of the penalty can by modified by multiplying it with a positive constant C. In addition, both the polynomial and the radial basis kernel functions also use an extra parameter γ which controls how wide-spread an influence support vectors have. Generally speaking, a small C value results in a high variance and a low bias, while a large C value leads to just the opposite result. In the same way, a large γ leads to a high bias and a low variance, and vice versa. The first model of the two models based on the support vector machine algorithm, uses a polynomial function as the kernel for the model. For the second model, the radial basis function (RBF) is chosen as the kernel type.

3.2 Ensemble methods

Section 3.1 described multiple different machine learning models, which can be used for regression problems. In practice, it is easy to implement all of these models with modern open source tools and train the models on the same data. After the training is done, the best performing model can be selected for generating the final predictions. However, instead of selecting one best model, better results can often be achieved by combining the predictions generated by different learning algorithms [23, 64]. This kind of model ensembles are now a standard [3].

There are three basic questions one has to answer when constructing an ensemble model. One should choose what kind of models are included in the ensemble estimator, how will the input features be fed to the models, and how will the results of the different models be combined.

The models chosen for the ensemble estimator can be either homogeneous or heterogeneous [44]. In a homogeneous ensemble estimator, all models use the same algorithm to find the best fit for the training data. Making multiple estimations with the same model, can help to reduce the test error [9, pp. 364-369]. Homogeneous ensemble estimators enable also dividing the prediction problem into smaller sub-samples which can be then learned individually [37]. In contrast, in heterogeneous models, different types of learning algorithms are used to find the best fit for the training data [44]. Heterogeneous ensemble models utilize the unique capabilities of different learning algorithms to help the model capture the different patterns in the data [44].

Also the input features can be fed to the models in two ways. One may choose to feed all input features in the same manner to all the models and rely that models can choose

the best features themselves. Another option is to customize the input features for each model. This kind of collaborative models should be able to take better advantage of the special characteristics of the individual learning algorithms. This way it is also possible to better avoid the curse of dimensionality because the number of input featured fed to a model can be reduced.

In regression problems, the models can be combined simply by averaging the results of individual models [9, pp. 653-654]. Another possibility is to view the model combination in an ordinary least squares framework. In this case, individual forecasts act as the explanatory variables and the actual values as the response values [20]. However, the second method requires enough data so that both the individual and the ensemble models can be trained. This means that the data is split into five different data sets.

In this thesis, an ensemble model, which combines the five models described in Section 3.1, is used for the sawnwood demand forecasting problem. These models are the Random forest, the K nearest neighbours, the Neural network, the Support vector machine with the polynomial kernel and the Support vector machine with the RBF kernel. The models are combined into the Ensemble estimator by taking the weighted average of these individual models' predictions. Thus, the final prediction y is

$$y(x; w) = \sum_{j=1}^{N} w y_j, \tag{3.6}$$

where w is a set of weights, y_j is the prediction output of the model j, N is number of the models.

The weights are assigned to the models based on the cross-validation scores so that better performing models get higher weights. While a smaller cross-validation means a smaller error, then the weight for the model j is calculated by taking the inverse of cross-validation scores. Before taking the inverse of the scores, the score can be raised to the power p

$$w_j = \frac{1}{\epsilon_j^p}, \tag{3.7}$$

where w_j is the weight and ϵ is the cross-validation error for the model j.

Figure 3.5 illustrates the Ensemble estimator applied to the sawnwood demand forecasting. The input data is fed to all five models in the same format. After this each model is trained and tested. The cross-validation picks the best settings for the individual models. Next, the Ensemble estimator is formed by taking the weighted sum of the models' pre-

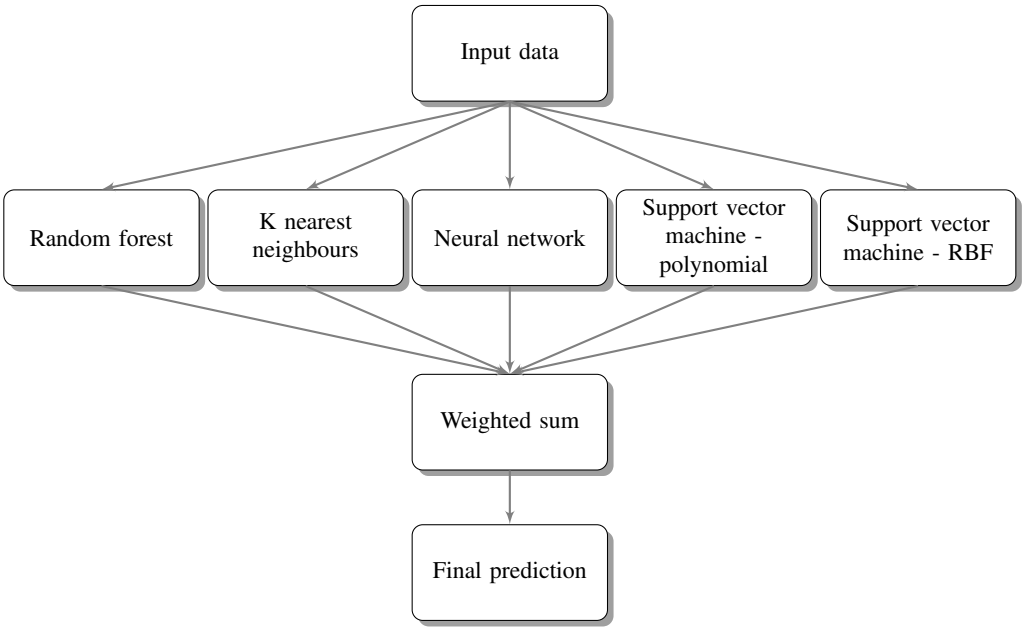


Figure 3.5 *The Ensemble estimator. The input data is fed to all five models in the same format. Next, each model is trained and tested. The cross-validation picks the best setting for the individual model, after which an ensemble of these models is formed by taking the weighted sum of the predictions produced by the models. The weights are assigned for the models based on the cross-validation score. As the result of this process, the estimator will produce the final prediction.*

dictions based on the cross-validation score. As the result of this process, the estimator will produce the final prediction.

4. SELECTING AND OPTIMIZING HYPER-PARAMETERS

All the machine learning models used in this thesis have some parameters which have to be set before the models can be trained to the data set. For the k nearest neighbours, these are the number of neighbours and the distribution of weights assigned to each neighbour. More advanced models, like the classifiers based on sophisticated feature extraction techniques, might have up to fifty parameters which affect the performance of the model [8]. These hyper-parameters affect the way the model behaves when new data is introduced to it. The target for choosing these hyper-parameters is usually to minimize the testing error [7]. This way, the results of the model can be generalized better. The problem of identifying the best values for the given hyper-parameters is called hyper-parameter optimization [7].

If the hyper-parameters are defined as vector λ, which contains all the values controlling the algorithm execution, then the hyper-parameter optimization can be formulated as a minimization problem, that is:

$$minf(\lambda) = \mathcal{L}(A_\lambda, \mathcal{D}_{train}, \mathcal{D}_{valid}), \tag{4.1}$$

where $\mathcal{D}_{train}$ is the training data set, $\mathcal{D}_{valid}$ is the validation data set, A is the machine learning algorithm and $\mathcal{L}$ is the loss function [7]. However in reality, solving this equation is in most cases impossible because it would require that the true generalization error of the model could be computed [7]. Thus, the range of possible values for each hyper-parameter must be limited so that a computer algorithm can search through the hyper-parameters set and select the best values for the parameters.

The hyper-parameter optimization algorithm can take two basic forms: the grid search or the random search. The grid search algorithm is a traditional way for tuning the hyper-parameters. The grid search algorithm searches exhaustively through a subset of the hyper-parameter space which is formed by assembling every possible combination of individual parameter values. However, the number of possible combinations grows exponentially to the number of hyper-parameters to be optimized [7]. This can be com-

Table 4.1 *The hyper-parameters used in the models. The hyper-parameters listed in the table will be optimized with the randomized search algorithm. The models applied in this thesis also include additional parameters which are not listed in the table. These parameters will be set to their default values.*

Model	**Hyper-parameters**
K nearest neighbours	Number of neighbours, Weight function
Random forest	Maximum depth of the tree, Number of the estimators
Support vector poly	C, γ
Support vector rbf	C, γ
Neural network	Weight decay, Dropout rate

putationally expensive if the number of hyper-parameters gets high.

Another commonly used method for the hyper-parameter selection is the random search algorithm [7, 8, 40]. The random search algorithm draws samples of parameters from a distribution over possible hyper-parameter values [7]. Compared to the grid search algorithm, the random search algorithm has proven to find as good or even better results with significantly less computing capacity [7, 8]. Usually machine learning algorithms are more sensitive to changes in some dimensions than others [16, 7]. The random search algorithm enables setting test cases more efficiently so that they can better cover the relevant dimensions [7]. With the random search algorithm, it is also easier to control the computing time if the dimensions change because the number of iterations is usually predefined in these algorithms. This is beneficial especially when building ensemble models for multiple data sets, because the computing time increases exponentially to the number of different learning algorithms used for training the different data sets.

In this thesis, the random search algorithm is used for selecting the hyper-parameters for each model. Table 4.1 lists the hyper-parameters for each machine learning model that is used for predicting the sawnwood demand in this thesis. The number of iterations used for searching the optimal parameter depends on the feature space of each model.

5. CASE STUDY

Chapter 5 presents the case study which was performed to evaluate the performance of the different machine learning models presented in Chapter 3. This chapter is divided into four sections. Section 5.1 describes the content and the preprocessing of data. After this, Section 5.2 presents the objectives, procedures and programs that where used in conducting the experiment. In Section 5.3 the error metrics that were applied to evaluate the performance of the different models are defined. Finally, Section 5.4 concludes the key findings of the study.

5.1 Data

5.1.1 Predicted variable

Each of the six machine learning models presented in Chapter 3 are tested on four different data sets. First two data sets represent the sawnwood demand in cubic meters (m^3) in the Finnish market so that the first data set is for whitewood (*Picea abies*) and the second for redwood (*Pinus sylvestris*). In these two cases, the sawnwood demand is measured as the total sawnwood consumption in Finland. Since there is no value readily available for sawnwood consumption, sawnwood consumption is calculated based on the production, warehousing and trade statistics using the following equation

$$Consumption = Production + Import - Export - Warehouse\,change \tag{5.1}$$

The third and the fourth data set represent the demand for imported sawnwood in France. Similarly to the Finnish market, data is collected separately for whitewood and redwood and measured in cubic meters (m^3). The granularity of the data corresponds to the eight digit level in the Harmonized Systems (HS) classification - a classification scheme used by the World Customs Organizations.

Sawnwood trade production and warehousing data are collected from the database of the

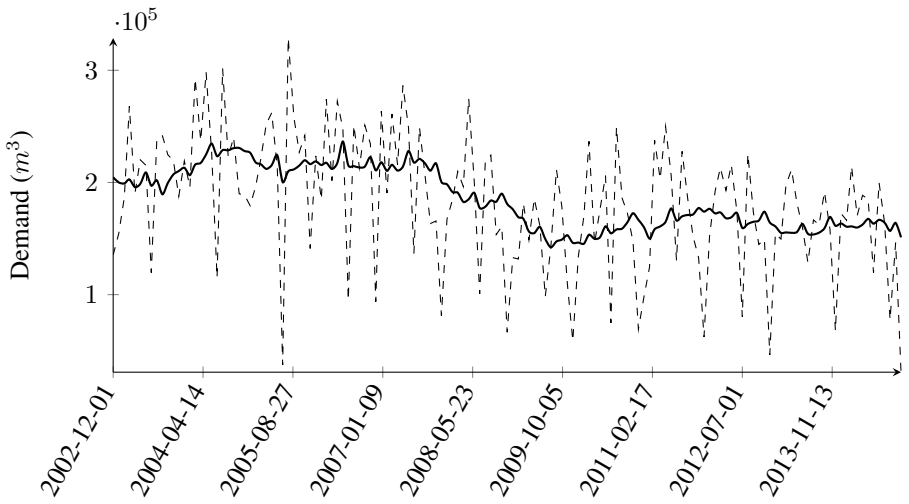


Figure 5.1 *The whitewood consumption in Finland. The dashed line represents the monthly consumption of whitewood in Finland calculated based on Equation 5.1. The solid black line is the trend of the whitewood consumption calculated as the twelve months moving average of the consumption values. The data is collected from the Eurostat's (available at http://ec.europa.eu/eurostat) and the Finnish Forest Industries Association's databases (available for request from https://www.forestindustries.fi/statistics/ for the members of the association).*

Finnish Forest Industries Association which is available for the members of the association. Trade statistics for Finland and France are collected from Eurostat's (the statistical office of the European communities) database. Table 5.1 summarizes the content and the sources for the data of the predicted variables.

Data is collected for a time period from 1st of January 2002 to 30th of June 2016 at a monthly frequency. This corresponds to a total amount of 174 labelled data points for the predicted variable and for the input features described in the next section (Section 5.1.2). Figure 5.1 illustrates what the data looks like for the whitewood data set of Finland. The figure shows that the data is characterized by strong seasonality and the demand might also undergo structural changes.

5.1.2 Features for sawnwood demand forecasting

In addition to the historical values for the trade data, a number of features are used as leading indicators for predicting the sawnwood demand in the given market. Leading indicators are economic variables which anticipate or contain useful information for predicting future developments in other variables. As it was stated earlier in this thesis, sawnwood demand can be seen as a derived demand of the different end uses of sawnwood. Based on the expert knowledge of the case company's representatives, several features were identified as the leading indicators for the sawnwood demand. After multiple iteration rounds, the list was reduced to six indicators: historical demand values, building permits, housing

loans, production in construction industry, month and year. These indicators are used as the leading indicators in all four models. Table 5.2 gives a detailed description of each indicator. In addition, multiple different leading indicators, like foreign exchange rates, gross domestic production, purchase manager index were fed as an input variable for the models. However, these indicators were excluded from the final data set of input features because they did not improve the performance of the model.

Historical demand values: The previous twelve months demand values act as input values for each model. The historical demand values set the baseline for the forecast and help the forecast models to pick the potential lags from the demand data.

Building permits: A building permit is the final authorization to start work on a building project and usually they are granted by public authorities in response to an application based on a specific building plan [26]. Building permits are measured in this thesis as the square metres of useful floor area granted in a given month. The number of building permits provides information about the workload of the construction industry in the near future [26]. Eurostat states that there are differences in the rules and procedures according to which such permits are granted in the European Union Member States, but in none of the countries does the permit imply an obligation to start the construction [26]. This is why the number of building permits overestimates the actual building projects realized later in the future [26].

Housing loans: Housing loans are defined in this thesis as loans to households for the purpose of purchasing a house or improving a house purchased earlier. For Finland and France the amount of Euro-denominated housing loans is available in European Central Bank's database as a floating rate or as an initial rate fixation to euro area households [25]. Housing loans are usually secured by residential property (i.e. mortgage loans) that is used for house purchase or by other types of assets [25].

Production in construction: Production in construction indicates the output and activity of the construction sector measured as monthly changes in the volume of the output [28]. Production in construction is compiled as a fixed base year volume-index so that the current base year is 2010 (Index 2010 = 100) [28]. Both building construction and civil engineering works are included in this index [28].

Month: Month is included into the input data set as an integer value. This helps the machine learning models to pick seasonality from the sawnwood demand data which is important due to the strong seasonality effect in the sawnwood demand.

Year: Year is also given as an integer value. This should help the different learning algorithms to better capture the trend from the demand data.

Table 5.1 *The description of the predicted variable data sets used in the testing of model. The models are tested on four different data sets: whitewood (Data set 1) and redwood (Data set 2) consumption in Finland and whitewood (Data set 3) and redwood (Data set 4) import to France. All values are collected as floating point values. Eurostat's database is available at http://ec.europa.eu/eurostat and Finnish Forest Industries' data sets are available for request from https://www.forestindustries.fi/statistics/ for the members of the association.*

Data set	**Country**	**Species**	**Data**	**Source**	**Unit**
Data set 1	Finland	Whitewood	Import	Eurostat	m^3
			Export	Eurostat	m^3
			Production	Finnish Forest Industries	m^3
			Warehouse	Finnish Forest Industries	m^3
Data set 2	Finland	Redwood	Import	Eurostat	m^3
			Export	Eurostat	m^3
			Production	Finnish Forest Industries	m^3
			Warehouse	Finnish Forest Industries	m^3
Data set 3	France	Whitewood	Import	Eurostat	m^3
Data set 4	France	Redwood	Import	Eurostat	m^3

Table 5.2 *The description of the input features. Six different variables are used as input features for the models. Values are collected as floating point or integer values. Eurostat's database is available at http://ec.europa.eu/eurostat, Natural Resource Institutes database at http://statdb.luke.fi/PXWeb/pxweb/en/, Statistics Finland at http://pxnet2.stat.fi/PXWeb/pxweb/en/StatFin/ and European Central Bank's database at http://sdw.ecb.europa.eu.*

Country	Feature	Source	Description	Data type
Finland	Historical consumption	Finnish Forest Industries, Eurostat	Historical consumption values of previous 12 months calculated based on Equation 5.1 (m^3)	Float
	Building permits	Statistics Finland	Floor area (m^2) for detached houses	Float
	Housing loans	European Central Bank	Lending for house purchase - Financial transactions (m€)	Float
	Production in construction	Eurostat	An indicator which measures monthly changes in the price adjusted output of construction	Float
	Month		1:Jan, 2:Feb, ..., 12:Dec	Integer
	Year		2002, 2003, ..., 2016	Integer
France	Historical import	Eurostat	Historical import values of previous 12 months (m^3)	Float
	Building permits	Eurostat	An index for floor area of new residential buildings excluding residencies for communities measured in square meters	Float
	Housing loans	European Central Bank	Lending for house purchase - Financial transactions (m€)	Float
	Production in construction	Eurostat	An indicator which measures monthly changes in the price adjusted output of construction	Float
	Month		1:Jan, 2:Feb, ..., 12:Dec	Integer
	Year		2002, 2003, ..., 2016	Integer

5.1.3 Processing the data

The data is collected into data sheets where each row represents a certain time period and each column consists of the input data for a given feature. Before the learning algorithms can be fitted into the data, the data is standardized by removing the mean and scaling to unit variance. This helps machine learning algorithms to perform better because many of them (e.g. support vector machine and neural networks) assume that all features are centered around zero and have a variance in the same order [35, 58, 42]. A feature that has an orders of magnitude larger variance than other features might dominate the objective function and make the estimator unable to learn from the other features correctly or learning might become slower [35, 42].

Standardization is done so that

$$z = \frac{x - \mu}{\sigma} \tag{5.2}$$

with mean

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i) \tag{5.3}$$

and standard deviation

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \tag{5.4}$$

The final results are calculated by reversing this process after the learning algorithms have been fitted to the data and the validation results are calculated based on the fitted models.

After standardization the data is split into training, validation and testing sets. The data from January 2002 to May 2014 is used for training and validation, and the rest of the data is used for the final testing of the developed model. The training and the validation values are treated as a single data set because the validation is done using K-fold cross validation. In this process, a training sample is divided into k sub-samples of which $k - 1$ samples are used for training and one sample is used for validation at once. The procedure is repeated k times so that each sub-sample is used once as a validation set. In this thesis, k is set to five which means that the sample is divided so that 120 data points are used for the training and 30 data points for the validation. The K-fold cross-validation is chosen

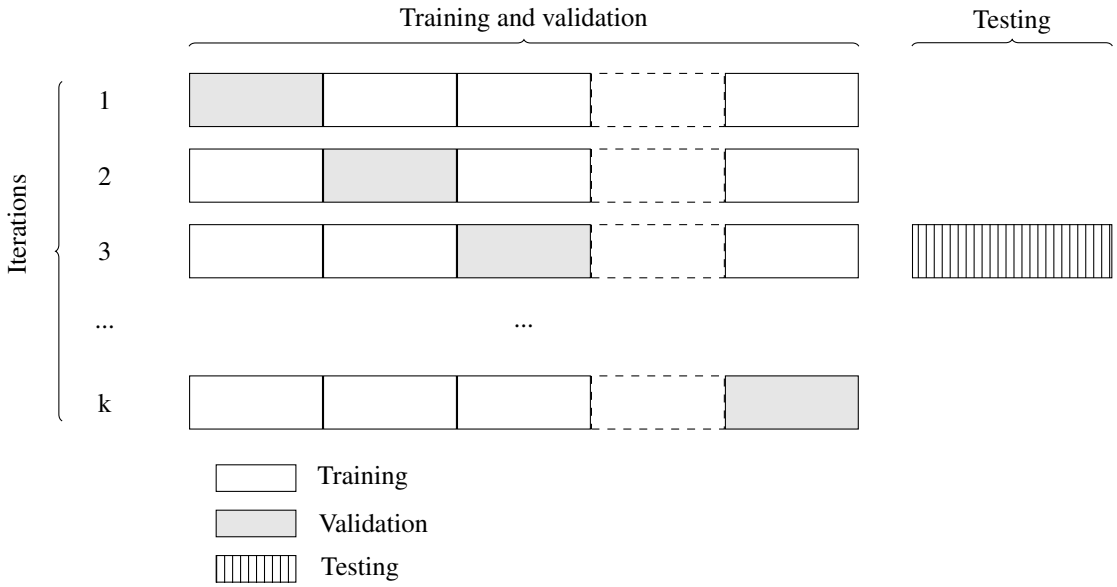


Figure 5.2 *An example of splitting data into the training, the validation and the testing data sets. Each of the four data sets is divided into three parts: the training, the validation and the testing data sets. The training and validation of the models is done by using the K-fold cross validation in which the training sample is divided into k sub-samples of which $k-1$ sample are used for training and one sample for validation at once. The procedure is repeated k times so that each sub-sample is used once as validation set. After the training and validation are done the performance of the models is tested with the testing data set.*

as validation method because it should provide a robust model selection [6]. This is an useful feature for the validation model selection when the chosen model should perform well on multiple different data sets.

After the data is split to training, validation and testing data sets, the data is transformed into the input and output data sets. The input data set is formulated so that for each time period t, twelve most recent values of each input variable are combined as a single input array. For example, for the forecast of whitewood import to France at the time period t, for each input variable x values at time periods $t-1...t-12$ are picked into the input array. Thus, the input values at time t are:

$$Inputdata(t) = \quad x_1(t), ..., x_1(x_{-11}), ..., x_j(t), ..., x_j(t_{-11}) \tag{5.5}$$

This way each input data set for whitewood import to France contains in total 72 variables (6 input features, 12 data points for each feature) which are used for predicting the output variables y.

In the same way, output variables are transformed into data sets which contain demand values for next twelve months.

$$Outputdata(t) = \quad y(t_1), ..., y(t_{12}) \tag{5.6}$$

Using the Direct strategy introduced in Section 2.4, t functions $f_t(x)$ are estimated from the data. The final prediction is a collection of the predictions calculated based the individual models which can be formulated as follows:

$$\begin{gathered} y_{t+1}, y_{t+13}, ..., y_{n-12} = f_1(x_1(t), ..., x_1(t_{-11}), ..., x_j(t), ..., x_j(t_{-11})) \\ y_{t+2}, y_{t+14}, ..., y_{n-11} = f_2(x_1(t_1), ..., x_1(t_{-10}), ..., x_j(t_1), ..., x_j(t_{-10})) \\ ... \\ y_{t-n-12}, ..., y_n = f_t(x_1(t_{n-1}), ..., x_1(t_{n-11}), ..., x_j(t_{n-11}), ..., x_j(t_{n-11})), \end{gathered} \tag{5.7}$$

where n is the number of the input-output data set combinations, t is the horizon for the forecast (i.e the number of the time-steps for which the forecast is produced) and j is the number of input features.

5.1.4 Potential problems with the data

Discrepancies in the data are known problems with trade statistics in general and especially for the trade statistics of forest products. These discrepancies create noise to the trade statistics which will affect the accuracy of the predictions. For example, researchers have analyzed the differences between SITC (Standard international trade classification) Revision 2 total trade values and values for four-digit SITC components published in United Nations' Comtrade database [57]. The results point out that the values for these two reporting levels differentiated 1 percentage for OECD (Organization for Economic Cooperation and Development) countries and 2.7 percentage for high income non-OECD countries. This can be seen as a lower limit for the forecasting error.

In addition, the trade data can be quite erratic [67, 27]. Solely the time lag, which is the result of transport times and delays in processing the same operation, can be recorded under a different reference period [27]. For this reason, the forecasting errors are calculated also from the moving average with different window lengths.

5.2 Experiment design

5.2.1 Objectives

The objective of this experiment is to evaluate the performance of each prediction model in predicting the sawnwood demand in the selected markets. The performance is evaluated on overall level and for each market separately. The models are ranked based on their performance. In addition, the performance of the models is examined by comparing the moving average of prediction values to the moving average of actual values. The models are validated by forecasting the next twelve months' sawnwood demand for all four data sets. The demand data is aggregated by one month.

5.2.2 Procedures

The experiment is done by performing training and prediction for each market model individually. The models are run using an Intel core i5 processor which has 4 cores and 4 GB memory. On average, the models run approximately 6 minutes, and as a result of the experiment 30 sets of twelve month forecast for each of the four markets is formulated. This makes a total of 1 440 data points which can be used for evaluating the performance of the models.

The forecasting process includes five steps. First, the data is collected from the sources mentioned in Tables 5.1 and 5.2. After this, the data is preprocessed as described in Section 5.1.3. The data is not modified or cleaned before the preprocessing. Next, the individual models are trained and cross-validated based on the training and validation data sets. Once the models are trained and the optimal values for the hyper-parameters are selected based on the cross-validation, the weights for the Ensemble estimator are assigned based on the cross-validation score. Finally, the newly constructed Ensemble estimator and the individual models are used for calculating the forecast for the validation data set. The performance of the models is evaluated based on these predictions and the performance metrics presented in the Section 5.3.

5.2.3 Programs

The models and the data collection for the models are coded in Python. The forecasting program uses multiple different Python packages. The standardization of data is done by *sklearn.preprocessing* package. The cross-validation for the models is done by the *RandomizedSearchCV* method of the *sklearn.model_selection* package. The packages which are used for building each model are mentioned in Chapter 3 - Models.

5.3 Performance metrics

Machine learning algorithms need an objective function to separate the well performing models from the ones that performs weekly [23]. The objective function is used both internally by the optimization function and externally by the model developer to evaluate the model performance. An external objective function, or functions, may differ from the internal objective functions. For external model evaluation, a good strategy is to use multiple objective functions. In this thesis, the model performance is evaluated by calculating the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the root mean squared error (RMSE) for the developed models and for a naive forecast. After this, the results are compared with one another.

The mean absolute error is the average of the absolute prediction errors. The MAE is a good metric for comparing the performance of different models in the same data set. Especially if the model developer is well aware of the output data characteristics, this measurement gives a good sense of how far away the prediction is from the actual values. The MAE is calculated by the following equation

$$MAE = \frac{1}{n}\sum_{t=1}^{n} ||F_t - A_i|| \tag{5.8}$$

where n denotes the number of data points, F_t the predicted values at time t and A_t the actual values at time t.

Another good option for comparing the performance of different prediction models in the same data set is the root mean squared error which is the square root of the average of squared prediction errors. The RMSE incorporates both the variance and the bias which usually leads to more stable models. Compared to the regular mean squared error (MSE), RMSE might be easier to interpret because the error measure has the same unit of measurement as the predicted value. The RMSE can be calculated by taking the square root of the mean squared error so that

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(F_t - A_t)^2}, \tag{5.9}$$

in which n is the number of data points, F_t the predicted values at time t and the A_t actual values at time t.

However, the MAE and the RMSE are not suitable for comparing prediction errors be-

tween different data sets. For this purpose, the mean absolute percentage error (MAPE) is a better option. The MAPE is also intuitively easy to understand. The MAPE can be calculated in the following way

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} ||\frac{F_t - A_t}{A_t}|| \tag{5.10}$$

where n denotes the number of data points, F_t the predicted values at time t and A_t the actual values at time t. Sometimes the MAPE might get infinite values if the actual values get close to zero. In this thesis all data values for each data set are fairly large positive numbers. For this reason, the MAPE will provide sensible results.

In addition to these three standard error metrics, the models are benchmarked against the naive forecast which is formed by taking an average of n previous values of the actual demand for the same month. This way, the naive forecast can be defined as

$$D_t = \frac{1}{n}\sum_{t=1}^{n} D_{t-n} \tag{5.11}$$

where D_t is the forecast for the demand at given month t, D_{t-n} is the actual demand at the same month in previous years and n is the number of years which are included to the average calculation. In this thesis, n is set to 3 because this produces the best estimate for the different data sets. For example, the naive forecast for the demand for sawnwood in May 2018 can be calculated as the average of the the demand for sawnwood in May 2015, 2016 and 2017.

5.4 Findings

The models are first compared by calculating the average error rates of all four data sets. Table 5.3a presents the MAE, the MAPE and the RMSE for each of the seven models. The Ensemble estimator has lower error rates than the other six forecasting models measured both in absolute (MAE and RMSE) and relative (MAPE) measures. The Support vector machine with polynomial kernel produces the second best forecast measured in all three error metrics and after it the third best model is the Neural network. The highest error rates are obtained by the K nearest neighbours algorithm which performs worst measured in MAPE, MAE and RMSE. In addition, the K nearest neighbours model is the only model which performs worse than the Naive forecast measured in all three error metrics. Also, the Random forest model has higher MAPE than the Naive forecast but

Table 5.3 *The MAPE, the MAE and the RMSE of each model. The errors are calculated as the average of the four data sets (a) from monthly values and (b) from twelve month moving averages. The embolden values represent the lowest error rate for each performance metric.*

(a) *Monthly data*

Model	**MAPE**	**MAE**	**RMSE**
Naive forecast	24.27	22 279	30 631
K nearest neighbours	28.62	24 584	32 475
Random forest	27.41	21 796	29 863
Support vector poly	17.53	14 952	21 950
Support vector rbf	18.70	16 925	25 026
Neural network	18.51	16 261	22 669
Ensemble estimator	**17.24**	**14 736**	**21 292**

(b) *Twelve months moving average*

Model	**MAPE**	**MAE**	**RMSE**
Naive forecast	16.22	17 004	20 068
K nearest neighbours	20.26	20 152	23 480
Random forest	18.08	18 427	21 820
Support vector poly	5.32	5 922	7 399
Support vector rbf	**4.03**	**3 891**	5 789
Neural network	5.39	5 114	6 826
Ensemble estimator	4.07	4 118	**5 362**

it scores lower MAE and RMSE values than the Naive forecast. Therefore, it seems that the models can be divided into two groups based on their performance. The first group includes the models which perform better than the Naive forecast measured in all three error metrics. The second group includes the models that perform worse than the Naive forecast measured in one or more error metrics. The first group consist of four models, which are the Ensemble estimator, the Neural network and the Support vector model with both polynomial and radial basis function kernel. The second group includes two models which are the K nearest neighbours and the Random forest.

Table 5.3b shows that error rates drop significantly when the rates are calculated from twelve months moving average of the predicted and actual values. The drop can be explained partly by the fact that the twelve months moving average smooths the strong seasonality in the data. For the Ensemble estimator, the MAPE drops 13.17 percentage points from 17.24 percentage to 4.07 percentage. Also for the Naive forecast, the MAPE drops 8.05 percentage points from 24.27 percentage to 16.22 percentage. The lowest MAPE is obtained by a model which uses the Support vector machine algorithm with radial basis function kernel. This model also scores the lowest value for MAE. However, the Ensemble estimator has the lowest RMSE, which means that on average the Ensemble estimator has few very large errors.

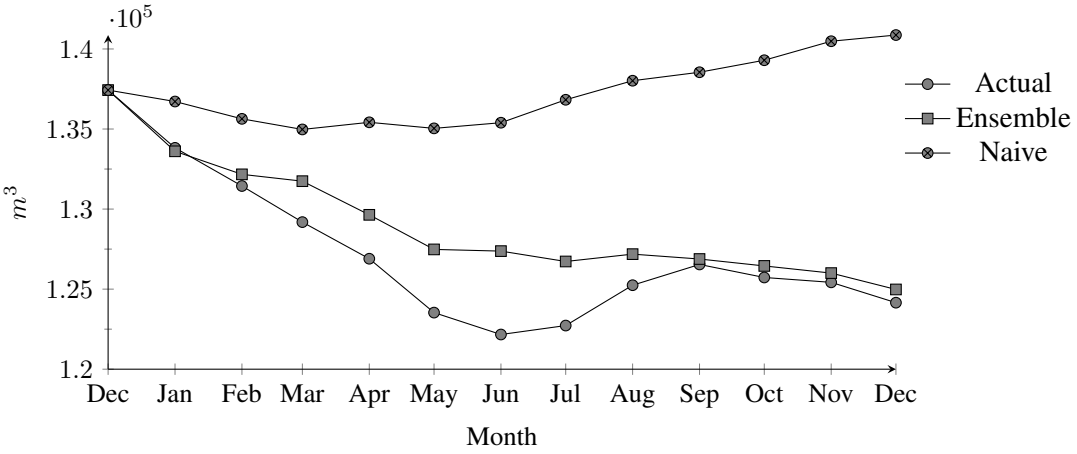


Figure 5.3 *An example of the results for twelve months rolling forecast. The figure illustrates the difference of the predictions calculated based on the Ensemble estimator and the Naive forecast for a one year time period. The lines represent the twelve months moving averages.*

The drop in the error rate is more significant for the more advanced learning algorithms (i.e Support vector machines, Neural network and Ensemble estimator) than for the more conventional models (i.e K nearest neighbours and Random forest). This indicates that these models are able to capture better the changes in the long-term trends of the sawn-wood demand. Figure 5.3 illustrates well how this effect can be seen when the forecast produced by the Ensemble estimator and the Naive forecast are compared. The Naive forecast produces results that seem to be going on just the opposite direction than where the market is going. The Ensemble estimator is able to predict the direction of the demand even if it is not able to capture fully the dynamics of the market.

Also, when taking the sum of ranks for each model per data set, the same models seem to be performing the best. The sum of ranks is calculated so that every model gets a rank number based on how well it performs on a given data set. The best-performing model gets rank number one and the worst-performing model rank number seven. The same ranking process is done for all four data sets. After each data set and model combination has a rank number, the rank numbers are summed.

Table 5.4a shows the MAPEs and the rankings based on the MAPE for each data set and model. The Support vector machine with the radial basis function kernel has the lowest rank sum for MAPE (9). The model has the lowest MAPE for both the whitewood data set of Finland and the whitewood data set of France. The Neural network performs best on the redwood data set of Finland while the Support vector machine with the polynomial kernel scores the lowest MAPE for the redwood data set of France. However, the difference in the error rates is not very large for the three best-performing models. The standard deviation of the MAPE varies from 1.1 to 3.3 percentage points while the average MAPE varies from 13.5 to 28.5 percentage.

Table 5.4 *(a) MAPE, ranking based on the MAPE and sum of the ranks for each model and data set, (b) MAE, ranking based on the MAE and sum of the ranks for each model and data set, (c) RMSE, ranking based on the RMSE and sum of the ranks for each model and data set Nf=Naive forecast, Knn=K nearest neighbours, Rf=Random forest, Svmp=Support vector machine with polynomial kernel, Svmr=Support vector machine with RBF kernel, Nn=Neural network, Ee=Ensemble estimator*

(a) *MAPE*

Dataset	**Nf**	**Knn**	**Rf**	**Svmp**	**Svmr**	**Nn**	**Ee**
Finland whitewood	16(2)	25(7)	22(6)	20(5)	15(1)	17(4)	16(3)
Finland redwood	39(6)	36(5)	39(7)	26(2)	34(4)	26(1)	28(3)
France whitewood	16(2)	25(7)	22(6)	20(5)	15(1)	17(4)	16(3)
France redwood	16(5)	24(7)	21(6)	12(1)	14(3)	15(4)	13(2)
Sum of ranks	**(15)**	**(26)**	**(25)**	**(13)**	**(9)**	**(13)**	**(11)**

(b) *MAE*

Dataset	**Nf**	**Knn**	**Rf**	**Svmp**	**Svmr**	**Nn**	**Ee**
Finland whitewood	16 793(2)	23 202(7)	18 488(5)	18 316(4)	18 186(3)	18 856(6)	15 545(1)
Finland redwood	38 416(7)	35 190(5)	36 390(6)	26 408(1)	35 003(4)	26 689(2)	28 024(3)
France whitewood	16 793(2)	23 202(7)	18 488(5)	18 316(4)	18 186(3)	18 856(6)	15 545(1)
France redwood	3 544(5)	4 648(7)	4 111(6)	2 664(1)	3 002(3)	3 167(4)	2 755(2)
Sum of ranks	**(16)**	**(26)**	**(22)**	**(10)**	**13)**	**(18)**	**(7)**

(c) *RMSE*

Dataset	**Nf**	**Knn**	**Rf**	**Svmp**	**Svmr**	**Nn**	**Ee**
Finland whitewood	22 602(2)	30 468(7)	25 780(6)	24 333(5)	22 860(3)	23 834(4)	20 408(1)
Finland redwood	44 796(7)	40 878(4)	43 239(6)	32 819(2)	41 595(5)	32 135(1)	33 660(3)
France whitewood	22 602(2)	30 468 (7)	25 780(6)	24 333(5)	22 860(3)	23 834(4)	20 408(1)
France redwood	4 181(5)	5 279(7)	4 583(6)	3 235(1)	3 568(3)	3 827(4)	3 299(2)
Sum of ranks	**(16)**	**(25)**	**(24)**	**(13)**	**(14)**	**(13)**	**(7)**

The Neural network, the Support vector machine with the polynomial kernel, the Support vector machine with the RBF kernel and the Ensemble estimator get the lowest sum of ranks when the error metric is changed to MAE as can be seen from the Table 5.4b. This time the Ensemble estimator has the lowest rank sum of 7 while the Support vector machine with the polynomial kernel has the second lowest rank sum of 10. The Ensemble estimator has the lowest MAE value for the both whitewood data sets while the Support vector machine with the polynomial kernel has the lowest MAE for the both redwood data sets.

Finally, Table 5.4c shows the same comparison for the models with RMSE as the error metric. This time, the Ensemble estimator has the lowest sum of ranks because it gets the lowest RMSE for the both whitewood data sets and the second lowest RMSE for the redwood data set of France. The Support vector machine with the polynomial kernel and

the Neural network model share the second place with the rank sum of 13. The Neural network has the lowest RMSE value for the redwood data set of Finland while the Support vector machine with the polynomial kernel has the lowest RMSE value for the redwood data set of France.

Figure 5.4 summarizes the results of the rank sum comparisons. The figure shows that the rank sums vary especially for the Support vector machine with the RBF kernel and for the Neural network. The rank sum for the Support vector machine with the RBF kernel is 9 when the MAPE is used as the error metric but it grows to 14 when the error metric is changed to the RMSE. Similar difference can be seen also when comparing the rank sums of different error metrics for the Neural network. The rank sum is 13 when the error is measured as MAPE or RMSE but it grows to 18 when the MAE is used as the error metric. Also, the Ensemble estimator has higher sum of ranks when the error metric is the MAPE than when the error metric is either RMSE or MAE. In contrast, the Naive forecast and the K nearest neighbours have only minor changes in the rank sums. The Naive forecast gets the rank sum 15 when the error metric is the MAPE and 16 when the error metric is the RMSE or the MAE. In the same way, the K nearest neighbours has the rank sum 26 when the error metric is the MAPE or the MAE, and 25 when the error metric is the RMSE.

The changes in the rank sums get smaller when the error rates are calculated from two month rolling means and they diminish when the error rates are calculated from the twelve months rolling means. Thus, it seems that the different models give more ambiguous results when the error rates are calculated based on the rolling mean with the shorter window length.

5.4.1 Comparing the performance of the models on different markets

European sawnwood markets follow demand patterns that are similar kind with one another. The demand is usually higher during spring and autumn seasons and lower during summer and winter seasons. The demand also varies a lot from month to month. However, there are also some differences between the demand in different markets which makes it interesting to compare the prediction accuracy of the different models also on a market level. This comparison is done by calculating the MAPE from the rolling means calculated with different window lengths. The MAPE is used because it allows the comparison of time series with different scales. The results are illustrated in Figures 5.5a and 5.5b for the whitewood and the redwood data sets of Finland and in Figures 5.5c and 5.5d for the whitewood and the redwood data sets of France.

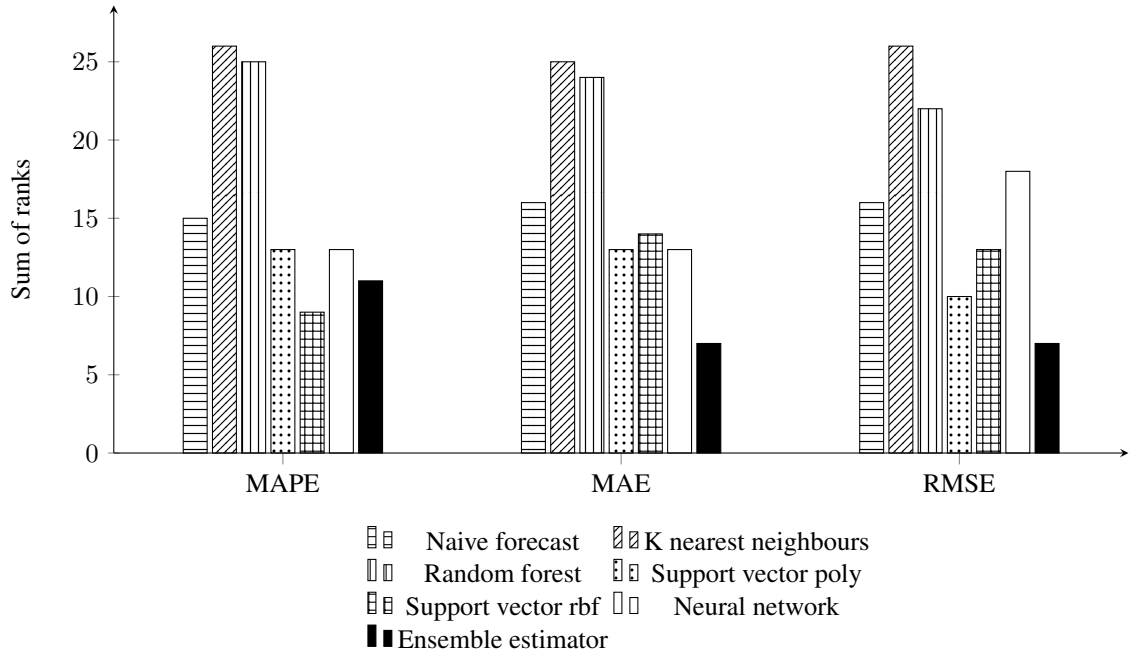


Figure 5.4 *Sum of ranks for each model calculated based on MAPE, MAE and RMSE (More detailed information is provided in Appendices A, B and C.). Every model gets a rank number based on how well it performs on a given data set. The best model gets rank number one and the worst model rank number seven. The same is done for all four data sets. After the each data set is ranked, the rank numbers are summed.*

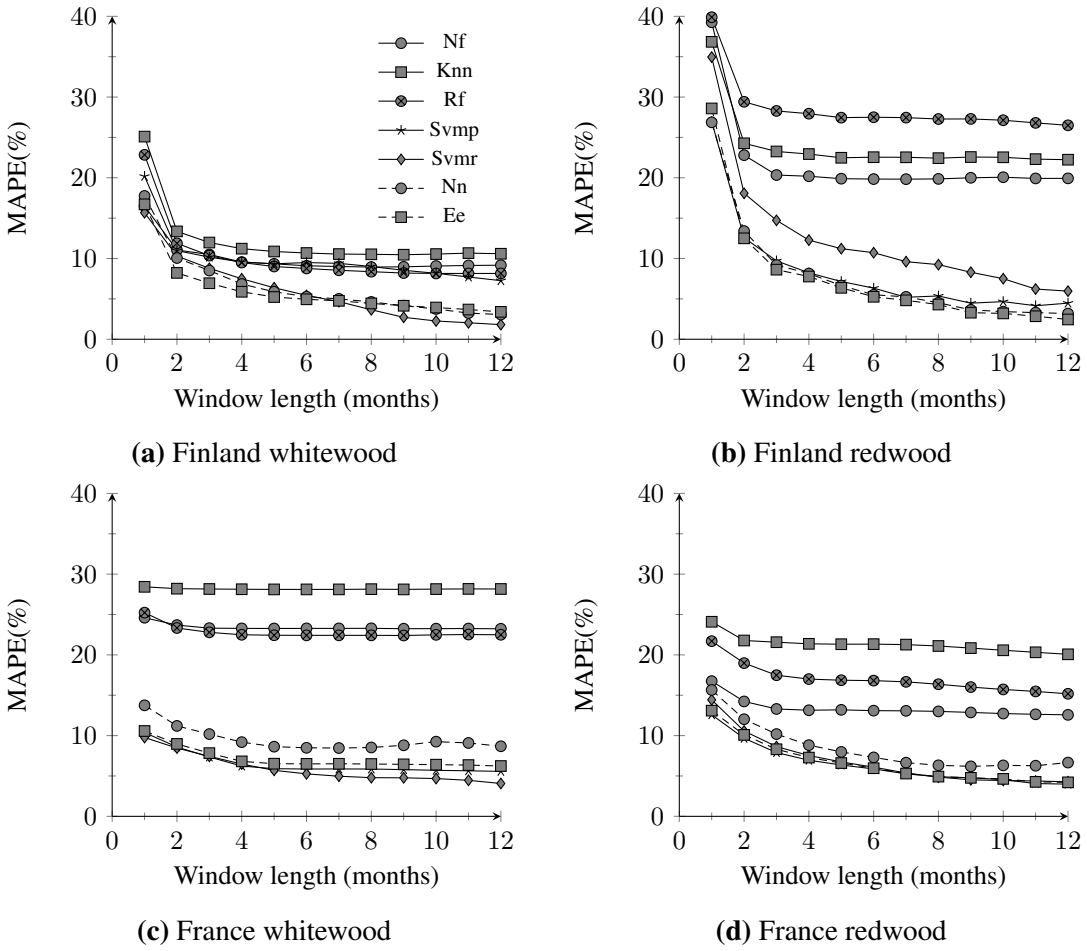
The results show that all seven models get the highest MAPE for the redwood data set of Finland. The difference to the second highest MAPE varies from 0.26 to 4.67 percentage points depending on the model. Both Support vector machine models, the K nearest neighbours and the Ensemble estimator score the lowest MAPEs for the whitewood data set of France while the K nearest neighbours and the Random forest get the lowest MAPE for the redwood data set of France. The Naive forecast is the only model which has the lowest MAPE value for the whitewood data set of Finland. The difference between the lowest and the highest MAPE values is biggest for the Support vector machine with the radial basis function kernel which has a spread of 25.19 percentage points between the French whitewood and Finnish redwood data sets.

The MAPEs drop significantly when the error rates are calculated from rolling means with the longer window lengths. At the same time, also the rankings between different data sets change. Only the Naive forecast has the lowest and the highest MAPE for the same data sets with different window lengths. Figures 5.5a and 5.5b give more detailed information of the performance of each model on the redwood and the whitewood data sets of Finland when the MAPE is calculated from the rolling mean with different window lengths. The biggest drop in the error rate can be seen when the window length is changed from one to two. For the Naive forecast, the Random forest and the K nearest neighbours models, the error rate levels out after the window length gets greater than four months. For the other

four models, the error rate goes down until the twelve months limit is reached.

A comparison of Figures 5.5a and 5.5b shows that the MAPEs are smaller for the whitewood data set of Finland than for the redwood data set of Finland when the window length is under eight months for most of the models. The only exception is the Support vector machine with the polynomial kernel for which the MAPE gets smaller for the redwood data set already after the window length grows greater that two months. After this, the MAPEs get smaller for the redwood data set. However, with the longer window lengths the difference between the MAPEs decreases. With the window length of 12 months, the smallest MAPE for the whitewood and the redwood data sets are 1.81 (obtained by the Support vector machine with the RBF kernel) and 2.45 (obtained by the Ensemble estimator) percentage respectively.

The same effect also applies to the French market. For the two Support vector machine models, the Neural network and the Ensemble estimator, the forecasts for the whitewood data set are more accurate than for the redwood data set when the window length is smaller than three months for the Neural network, seven months for the Support vector machine with the polynomial kernel, nine months for the Support vector machine with the RBF kernel and six months for the Ensemble estimator (Figures 5.5c and 5.5d). After this, the MAPE is smaller for the redwood forecasts. For the Naive forecast, the K nearest neighbours and the Random forest, the effect for the both data sets of France is just the opposite as with the data sets of Finland: the MAPEs are smaller for the redwood data set with every window length. (Figures 5.5c and 5.5d). With a window length of twelve months, the smallest MAPE is obtained by the Support vector machine with the RBF kernel for the whitewood data set of France (4.07 percentage) and by the Support vector machine with the polynomial kernel for the redwood data set of France (3.99 percentage).



(a) Finland whitewood

(b) Finland redwood

(c) France whitewood

(d) France redwood

Figure 5.5 *MAPE for the forecasts of whitewood (a) and redwood (b) demand in Finland and for whitewood (c) and redwood (d) import to France using different window lengths for calculating the moving average. The MAPE is calculated for window lengths from 1 to 12 for seven different models: Nf=Naive forecast, Knn=K nearest neighbours, Rf=Random forest, Svmp=Support vector machine with polynomial kernel, Svmr=Support vector machine with RBF kernel, Nn=Neural network and Ee=Ensmeble estimator.*

6. CONCLUSIONS

This thesis proposes a mathematical model for sawnwood demand forecasting. The final model is an ensemble of machine learning models which gives a prediction based on the weighted sum of the forecasts produced by five different machine learning models. These five machine learning models are: the K nearest neighbours, the Random forest, the Support vector machine with the polynomial kernel, the Support vector machine with the RBF kernel and the Neural network. Six different variables were given as input features for the model. These variables include four market-specific factor such as the historical sawnwood demand, the number of housing loans, the number of building permits and the volume of production in the construction industry, and two dummy variables: month and year. The performance of the model was evaluated based on a case study in which four different data sets were used for testing the prediction accuracy of the model. The data sets represent the whitewood and redwood demand in Finland and France at a monthly interval. For each data set, the prediction accuracy of the models was measured with three error metrics: the MAPE, the MAE and the RMSE. In addition, the performance of the model was compared against the individual learning algorithms and a naive forecast. Thus, a total of seven different models were applied to produce a forecast for each data set.

The results of the case study show that the Ensemble estimator outperforms the other six models measured in all three error metrics when the error rates are calculated as the average of the four data sets. However, when the results are compared at individual data set level, the Ensemble estimator performs best only in four out of twelve cases. This result indicates that a single method cannot provide the best answer in all of the prediction tasks. All in all, four out of seven models provide the lowest error rate at least for one data set and one error metric. When the error rates are calculated from the twelve months moving averages of the predicted and the actual values, the error rates drop. The error rates decrease more for the more advanced learning models, like the Support vector machine with the polynomial kernel, the Support vector machine with the RBF kernel, the Neural network and the Ensemble estimator, than for the more conventional models such as the K nearest neighbours and the Random forest. This result indicates that the more advanced models are able to capture the trend component better from the data sets.

The results also demonstrates that there are differences in how well the models can predict the sawnwood demand in different markets. The accuracy of the predictions can vary significantly depending on the characteristics of the data set. The difference between the highest and the lowest MAPE for a model can change over 20 percentage points when the error rates are calculated at a monthly interval. When the error rates are recalculated from the 12 months moving averages, the difference between the highest and the lowest MAPE value remains as high as 20 percentage for the more conventional models like the K nearest neighbours and the Random forest. When the same calculation principles are applied for the more advanced models, like the Support vector machine with a polynomial kernel, the Support vector machine with a RBF kernel, the Neural network and the Ensemble estimator, the difference in the error rates decreases to four percentage points. The results indicates that the more advanced models are able to produce better predictions when the data sets are aggregated on a higher level.

This thesis proves that machine learning methods can be applied successfully for sawnwood demand forecasting. Even a well-established business, like the forest industry, can benefit from applying modern mathematical models for their demand forecasting efforts similarly like the high-tech companies that are usually associated with this kind of innovations. At the same time, this thesis introduces machine learning as a new method for the research in the field of forest science. For practitioners this thesis provides guidelines for applying the machine learning methods for the demand forecasting. The case study describes the data, the processes and the programs that are required for building the Ensemble estimator presented in this thesis. The same methodology can be applied also for forecasting other commodities, for which the data is available.

Naturally, the Ensemble estimator developed in this thesis also has some limitations. In addition to the four data sets, presented in Chapter 5, the model was initially tested also on a fifth data set. This data set contained the sawnwood import to China between the years 2010 and 2016. However, the number of the data points was too low for training and testing the models. The data set contained only 78 labelled data points which is not enough for training and testing the Ensemble estimator properly. The model needs at least 150 data points to function well. The more data the models can use, the better they perform. In cases where the amount of data is limited, more traditional models, like the autoregressive and moving average based models, could potentially perform better. Furthermore, the random search algorithm used for the hyper-parameter optimization in this thesis needs some sort of limitations for the space from which the hyper-parameters are searched. Finding the optimal ranges for the hyper-parameters is still a somewhat heuristic process. Thus, some level of caution is needed when the predictions of the Ensemble estimator and individual machine learning models are interpreted.

Future research could consider applying a collaborative model for the demand forecasting or applying the machine learning approaches to predict the demand of other commodities in the forest product value chain. In the collaborative model, the forecasting task is separated into multiple sub-tasks taken care of by different models. The whole feature space is split into several feature subsets which will serve as input values for a specific model. This could improve the performance of the model, because the number of input features fed to a model could be reduced. This would also enable the model developer to better capture the full potential of different machine learning algorithms. The Ensemble estimator could also be applied to other markets where data is available. Potential countries for the future research are all EU member countries because they report their trade statistics to the Eurostat, like Finland and France that were used as target markets in this thesis. However, this also requires better understanding of the input features which explain the sawnwood demand in the given market. It would also be interesting, to compare the performance of the machine learning models to the performance of statistical methods, like the vector-auto-regression or different forms of the auto-regressive models with external input features. By applying the statistical and the machine learning methods for predicting different commodities in the forest sector, a better understanding could be gained about suitable methods for solving different kinds of forecasting problems.

BIBLIOGRAPHY

[1] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An Empirical Comparison of Machine Learning Models for Time Series Forecasting," *Econometric Reviews*, vol. 29, no. 5-6, pp. 594–621, Aug. 2010. [Online]. Available: http://dx.doi.org/10.1080/07474938.2010.481556

[2] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992. [Online]. Available: http://amstat.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879

[3] E. Bauer and R. Kohavi, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," *Machine Learning*, vol. 36, no. 1-2, pp. 105–139, Jul. 1999. [Online]. Available: http://link.springer.com/article/10.1023/A:1007515423169

[4] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7067–7083, Jun. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417412000528

[5] S. Ben Taieb, A. Sorjamaa, and G. Bontempi, "Multiple-output modeling for multi-step-ahead time series forecasting," *Neurocomputing*, vol. 73, no. 10–12, pp. 1950–1957, Jun. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231210001013

[6] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, May 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025511006773

[7] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012. [Online]. Available: http://www.jmlr.org/papers/v13/bergstra12a.html

[8] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger,

Eds. Curran Associates, Inc., 2011, pp. 2546–2554. [Online]. Available: http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf

[9] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, Oct. 2007.

[10] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, Apr. 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304407686900631

[11] G. Bontempi, S. B. Taieb, and Y.-A. L. Borgne, "Machine Learning Strategies for Time Series Forecasting," in *Business Intelligence*, ser. Lecture Notes in Business Information Processing, M.-A. Aufaure and E. Zimányi, Eds. Springer Berlin Heidelberg, 2013, no. 138, pp. 62–77, dOI: 10.1007/978-3-642-36318-4_3. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-36318-4_3

[12] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, New Jersey: Wiley, Jun. 2015.

[13] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://link.springer.com/article/10.1023/A:1010933404324

[14] ——, "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)," *Statistical Science*, vol. 16, no. 3, pp. 199–231, Aug. 2001. [Online]. Available: http://projecteuclid.org/euclid.ss/1009213726

[15] J. Buongiorno, "Forest sector modeling: a synthesis of econometrics, mathematical programming, and system dynamics methods," *International Journal of Forecasting*, vol. 12, no. 3, pp. 329–343, Sep. 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0169207096006681

[16] R. E. Caflisch, W. Morokoff, and A. Owen, *Valuation of Mortgage Backed Securities Using Brownian Bridges to Reduce Effective Dimension*. UCLA CAM Report, 1997.

[17] L. Cao, "Support vector machines experts for time series forecasting," *Neurocomputing*, vol. 51, pp. 321–339, Apr. 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231202005775

[18] D. Castelvecchi, "Can we open the black box of AI?" *Nature News*, vol. 538, no. 7623, p. 20, Oct. 2016. [Online]. Available: http://www.nature.com/news/can-we-open-the-black-box-of-ai-1.20731

[19] C. Chatfield, *Time-Series Forecasting*, 1st ed. Boca Raton, Florida, USA: CRC Press, Oct. 2000.

[20] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," *International Journal of Forecasting*, vol. 5, no. 4, pp. 559–583, Jan. 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0169207089900125

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Feb. 1995. [Online]. Available: http://link.springer.com/article/10.1007/BF00994018

[22] P. Domingos, "A Unified Bias-Variance Decomposition and its Applications," *In Proc. 17th International Conf. on Machine Learning*, pp. 231–238, 2000.

[23] ——, "A Few Useful Things to Know About Machine Learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012. [Online]. Available: http://doi.acm.org/10.1145/2347736.2347755

[24] R. F. Engle, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982. [Online]. Available: http://www.jstor.org/stable/1912773

[25] European Central Bank, "All glossary entries glossary." [Online]. Available: https://www.ecb.europa.eu/home/glossary/html/glossp.en.html

[26] Eurostat, "Glossary:Building permit - Statistics Explained," 2016. [Online]. Available: http://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Building_permit

[27] Eurostat, "International trade statistics - background - Statistics Explained," 2016. [Online]. Available: http://ec.europa.eu/eurostat/statistics-explained/index.php/International_trade_statistics_-_background

[28] Eurostat, "Production in construction - Eurostat," 2017. [Online]. Available: http://ec.europa.eu/eurostat/web/products-datasets/-/teiis500

[29] S. E. Fahlman and C. Lebiere, "The Cascade-Correlation Learning Architecture," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 524–532. [Online]. Available: http://papers.nips.cc/paper/207-the-cascade-correlation-learning-architecture.pdf

[30] Finnish Forest Industries (Metsäteollisuus ry), "Sawmill industry - Industry - Statistics - Forestindustries - Sawn redwood and whitewood production in Finland," Jan. 2017. [Online]. Available: https://www.forestindustries.fi/statistics/industry/20-Sawmill%20Industry/

[31] E. Fix and J. Hodges, "Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties," USAF School of Aviation Medicine, Tech. Rep., Feb. 1951.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. Cambrigde, Massachusetts, United State of America: MIT Press, Nov. 2016.

[33] C. W. J. Granger, *Essays in Econometrics: Collected Papers of Clive W. J. Granger*, 1st ed. Cambridge, United Kingdom: Cambridge University Press, Jul. 2001.

[34] L. Hetemäki, R. Hänninen, and A. Toppinen, "Short-Term Forecasting Models for the Finnish Forest Sector: Lumber Exports and Sawlog Demand," *Forest Science*, vol. 50, no. 4, pp. 461–472, Aug. 2004.

[35] C.-w. Hsu, C.-c. Chang, and C.-j. Lin, *A practical guide to support vector classification*. Department of Computer Science, National Taiwan University, 2010.

[36] E. Hurmekoski, L. Hetemäki, and M. Linden, "Factors affecting sawnwood consumption in Europe," *Forest Policy and Economics*, vol. 50, pp. 236–248, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389934114001397

[37] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Mar. 1991. [Online]. Available: http://dx.doi.org/10.1162/neco.1991.3.1.79

[38] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge, United Kingdom: Cambridge University Press, 2004.

[39] A. Krogh and J. A. Hertz, "A Simple Weight Decay Can Improve Generalization," in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. Morgan-Kaufmann, 1992, pp. 950–957. [Online]. Available: http://papers.nips.cc/paper/563-a-simple-weight-decay-can-improve-generalization.pdf

[40] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, "An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 473–480. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273556

[41] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html

[42] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. B. Orr and K.-R. Müller, Eds. Springer Berlin Heidelberg, 1998, no. 1524, pp. 9–50, dOI: 10.1007/3-540-49430-8_2. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-49430-8_2

[43] P. Leskinen and J. Kangas, "Modelling and simulation of timber prices for forest planning calculations," *Scandinavian Journal of Forest Research*, vol. 13, no. 1-4, pp. 469–476, Jan. 1998. [Online]. Available: http://dx.doi.org/10.1080/02827589809383008

[44] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, Nov. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221715004208

[45] A. U. Levin, T. K. Leen, and J. E. Moody, "Fast Pruning Using Principal Components," in *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. Morgan-Kaufmann, 1994, pp. 35–42. [Online]. Available: http://papers.nips.cc/paper/754-fast-pruning-using-principal-components.pdf

[46] G. Louppe, "Understanding Random Forests: From Theory to Practice," *arXiv:1407.7502 [stat]*, Jul. 2014, arXiv: 1407.7502. [Online]. Available: http://arxiv.org/abs/1407.7502

[47] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, Jun. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865514000221

[48] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*, 1st ed. New York, United States of America: Springer Science & Business Media, Dec. 2005.

[49] B. Mei, M. Clutter, and T. Harris, "Modeling and forecasting pine sawtimber stumpage prices in the US South by various time series models," *Canadian Journal of Forest Research*, vol. 40, no. 8, pp. 1506–1516, Jul. 2010. [Online]. Available: http://www.nrcresearchpress.com/doi/abs/10.1139/x10-087

[50] N. Morgan and H. Bourlard, "Generalization and Parameter Estimation in Feedforward Nets: Some Experiments," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 630–637. [Online]. Available: http://papers.nips.cc/paper/

275-generalization-and-parameter-estimation-in-feedforward-nets-some-experiments.pdf

[51] K. Niquidet and L. Sun, "Do forest products prices display long memory?" *Canadian Journal of Agricultural Economics/Revue canadienne d'agroeconomie*, vol. 60, no. 2, pp. 239–261, 2012. [Online]. Available: http://cfs.nrcan.gc.ca/publications?id=33735

[52] S. J. Nowlan and G. E. Hinton, "Simplifying Neural Networks by Soft Weight-sharing," *Neural Comput.*, vol. 4, no. 4, pp. 473–493, Jul. 1992. [Online]. Available: http://dx.doi.org/10.1162/neco.1992.4.4.473

[53] A. Pankratz, *Forecasting with Univariate Box - Jenkins Models: Concepts and Cases*. John Wiley & Sons, Sep. 2009.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2078195

[55] L. Prechelt, "Early Stopping-But When?" in *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*. London, UK, UK: Springer-Verlag, 1998, pp. 55–69. [Online]. Available: http://dl.acm.org/citation.cfm?id=645754.668392

[56] J. P. Prestemon and T. P. Holmes, "Timber Price Dynamics Following a Natural Catastrophe," *American Journal of Agricultural Economics*, vol. 82, no. 1, pp. 145–160, Feb. 2000. [Online]. Available: https://academic.oup.com/ajae/article-abstract/82/1/145/64298/Timber-Price-Dynamics-Following-a-Natural

[57] J. Rozanski and A. Yeats, "On the (in)accuracy of economic observations: An assessment of trends in the reliability of international trade statistics," *Journal of Development Economics*, vol. 44, no. 1, pp. 103–130, Jun. 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304387894000085

[58] Scikit-learn developers, "sklearn.preprocessing.StandardScaler — scikit-learn 0.18.1 documentation," 2017. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

[59] M. R. Segal, "Machine Learning Benchmarks and Random Forest Regression," *Center for Bioinformatics & Molecular Biostatistics*, Apr. 2004. [Online]. Available: http://escholarship.org/uc/item/35x3v9t4

[60] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004. [Online]. Available: http://link.springer.com/article/10.1023/B:STCO.0000035301.49549.88

[61] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, no. 4, pp. 637–649, Jun. 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S089360809800032X

[62] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, and A. Lendasse, "Methodology for long-term prediction of time series," *Neurocomputing*, vol. 70, no. 16–18, pp. 2861–2869, Oct. 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231207001610

[63] A. Sorjamaa and A. Lendasse, "Time Series Prediction using DirRec Strategy," 2006.

[64] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://www.jmlr.org/papers/v15/srivastava14a.html

[65] S. B. Taieb, G. Bontempi, A. Sorjamaa, and A. Lendasse, "Long-term prediction of time series by combining direct and MIMO strategies," in *2009 International Joint Conference on Neural Networks*, Jun. 2009, pp. 3054–3061.

[66] B. J. Thorsen, "Spatial integration in the Nordic timber market: Longrun equilibria and shortrun dynamics," *Scandinavian Journal of Forest Research*, vol. 13, no. 1-4, pp. 488–498, Jan. 1998. [Online]. Available: http://dx.doi.org/10.1080/02827589809383010

[67] United States Bureau of the Census, *1969 technical notes on new census method for seasonal adjustment of foreign trade data*. United States of America: U.S. Dept. of Commerce, Bureau of the Census [for sale by the Supt. of Docs., U.S. Gov't. Print. Off., 1969.

[68] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer New York, 2000. [Online]. Available: http://link.springer.com/10.1007/978-1-4757-3264-1

[69] R. Xu, "Machine learning for real-time demand forecasting," Thesis, Massachusetts Institute of Technology, 2015. [Online]. Available: http://dspace.mit.edu/handle/1721.1/99565

[70] E. Zivot and J. Wang, *Modeling Financial Time Series with S-PLUS*. Springer Science & Business Media, Nov. 2013.

APPENDIX A. MAE FOR THE DIFFERENT DATA SETS

Table 1 *MAE (m^3) for each model when the window length for rolling mean is changed. Nf=Naive forecast, Knn=K nearest neighbours, Rf=Random forest, Svmp=Support vector machine with polynomial kernel, Svmr=Support vector machine with RBF kernel, Nn=Neural network, Ee=Ensemble estimator*

Window length	Data set	Nf	Knn	Rf	Svmp	Svmr	Nn	Ee
1 month	Finland - Whitewood	16 793(2)	23 202(7)	18 488(5)	18 316(4)	18 186(3)	18 856(6)	15 545(1)
	Finland - Redwood	38 416(7)	35 190(5)	36 390(6)	26 408(1)	35 003(4)	26 689(2)	28 024(3)
	France - Whitewood	16 793(2)	23 202(7)	18 488(5)	18 316(4)	18 186(3)	18 856(6)	15 545(1)
	France - Redwood	3 544(5)	4 648(7)	4 111(6)	2 664(1)	3 002(3)	3 167(4)	2 755(2)
	Sum of ranks	**(16)**	**(26)**	**(22)**	**(10)**	**13)**	**(18)**	**(7)**
2 months	Finland - Whitewood	14 736(6)	17 473(7)	14 162(2)	14 230(3)	14 290(5)	14 274(4)	10 918(1)
	Finland - Redwood	25 082(5)	26 343(6)	30 883(7)	14 035(2)	19 850(4)	14 612(3)	13 763(1)
	France - Whitewood	29 332(6)	35 118(7)	28 069(5)	10 633(2)	10 374(1)	13 756(4)	11 080(3)
	France - Redwood	2 952(5)	4 391(7)	3 704(6)	2 004(1)	2 198(3)	2 459(4)	2 087(2)
	Sum of ranks	**(22)**	**(27)**	**(20)**	**(8)**	**(13)**	**(15)**	**(7)**
12 months	Finland - Whitewood	13 540(6)	15 673(7)	12 054(5)	10 777(4)	2 692(1)	4 561(2)	5 021(3)
	Finland - Redwood	22 824(5)	25 444(6)	30 300(7)	5 137(3)	6 877(4)	3 672(2)	2 810(1)
	France - Whitewood	28 925(6)	35 128(7)	28 056(5)	6 906(2)	5 063(1)	10 780(4)	7 728(3)
	France - Redwood	2 727(5)	4 360(7)	3 296(6)	864(1)	931(3)	1 442(4)	912(2)
	Sum of ranks	**(22)**	**(27)**	**(23)**	**(10)**	**(9)**	**(12)**	**(9)**

APPENDIX B. RMSE FOR THE DIFFERENT DATA SETS

Table 2 *RMSE (m^3) for each model when the window length for rolling mean is changed. Nf=Naive forecast, Knn=K nearest neighbours, Rf=Random forest, Svmp=Support vector machine with polynomial kernel, Svmr=Support vector machine with RBF kernel, Nn=Neural network, Ee=Ensemble estimator*

Window length	Data set	Nf	Knn	Rf	Svmp	Svmr	Nn	Ee
1 month	Finland - Whitewood	22 602(2)	30 468(7)	25 780(6)	24 333(5)	22 860(3)	23 834(4)	20 408(1)
	Finland - Redwood	44 796(7)	40 878(4)	43 239(6)	32 819(2)	41 595(5)	32 135(1)	33660(3)
	France - Whitewood	22 602(2)	30 468(7)	25 780(6)	24 333(5)	22 860(3)	23 834(4)	20 408(1)
	France - Redwood	4 181(5)	5 279(7)	4 583(6)	3 235(1)	3 568(3)	3 827(4)	3 299(2)
	Sum of ranks	**(16)**	**(25)**	**(24)**	**(13)**	**(14)**	**(13)**	**(7)**
2 months	Finland - Whitewood	17 851(4)	22 010(7)	18 013(6)	17 871(5)	16 720(2)	16 893(3)	13 219(1)
	Finland - Redwood	29 612(5)	30 562(6)	34 139(7)	17 310(2)	23 900(4)	17 511(3)	17 219(1)
	France - Whitewood	32 026(6)	37 501(7)	29 248(5)	12 399(2)	12 057(1)	16 299(4)	12 796(3)
	France - Redwood	3 391(5)	4 758(7)	4 026(6)	2 417(1)	2 632(3)	2 946(4)	2 455(2)
	Sum of ranks	**(20)**	**(27)**	**(24)**	**(10)**	**(10)**	**(14)**	**(7)**
12 months	Finland - Whitewood	13 540(6)	15 673(7)	12 054(5)	10 777(4)	2 692(1)	4 561(2)	5 021(3)
	Finland - Redwood	22 824(5)	25 444(6)	30 300(7)	5 137(3)	6 877(4)	3 672(2)	2 810(1)
	France - Whitewood	28 925(6)	35 128(7)	28 056(5)	6 906(2)	5 063(1)	10 780(4)	7 728(3)
	France - Redwood	2 727(5)	4 360(7)	3 296(6)	864(1)	931(3)	1 442(4)	912(2)
	Sum of ranks	**(22)**	**(27)**	**(23)**	**(10)**	**(9)**	**(12)**	**(9)**

APPENDIX C. MAPE FOR THE DIFFERENT DATA SETS

Table 3 *MAPE (%) for each model when the window length for rolling mean is changed. Nf=Naive forecast, Knn=K nearest neighbours, Rf=Random forest, Svmp=Support vector machine with polynomial kernel, Svmr=Support vector machine with RBF kernel, Nn=Neural network, Ee=Ensemble estimator*

Window length	Data set	Nf	Knn	Rf	Svmp	Svmr	Nn	Ee
1 month	Finland - Whitewood	16(2)	25(7)	22(6)	20(5)	15(1)	17(4)	16(3)
	Finland - Redwood	39(6)	36(5)	39(7)	26(2)	34(4)	26(1)	28(3)
	France - Whitewood	16(2)	25(7)	22(6)	20(5)	15(1)	17(4)	16(3)
	France - Redwood	16(5)	24(7)	21(6)	12(1)	14(3)	15(4)	13(2)
	Sum of ranks	**(15)**	**(26)**	**(25)**	**(13)**	**(9)**	**(13)**	**(11)**
2 months	Finland - Whitewood	11(5)	13(7)	11(6)	10(4)	10(3)	10(2)	8(1)
	Finland - Redwood	22(5)	24(6)	29(7)	12(2)	18(4)	13(3)	12(1)
	France - Whitewood	23(6)	28(7)	23(5)	8(2)	8(1)	11(4)	8(3)
	France - Redwood	14(5)	21(7)	18(6)	9(1)	10(3)	12(4)	10(2)
	Sum of ranks	**(21)**	**(27)**	**(24)**	**(9)**	**(11)**	**(13)**	**(7)**
12 months	Finland - Whitewood	9(6)	10(7)	8(5)	7(4)	1(1)	3(2)	3(3)
	Finland - Redwood	19(5)	22(6)	26(7)	4(3)	5(4)	3(2)	2(1)
	France - Whitewood	23(6)	28(7)	22(5)	5(2)	4(1)	8(4)	6(3)
	France - Redwood	12(5)	20(7)	15(6)	3(1)	4(3)	6(4)	4(2)
	Sum of ranks	**(22)**	**(27)**	**(23)**	**(10)**	**(9)**	**(12)**	**(9)**